



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**PEMERINGKATAN PERGURUAN TINGGI
MENGUNAKAN METODE *LABELED LATENT
DIRICHLET ALLOCATION* (L-LDA) UNTUK MENGUKUR
TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN
TINGGI DI INDONESIA**

Eko Putro Fitrianto
NRP 5113 100 194

Dosen Pembimbing I
Prof. Drs. Ec. Ir. Rianarto Sarno, M.Sc.,Ph.D.

Dosen Pembimbing II
Bagus Setya Rintyarna, S.T., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**PEMERINGKATAN PERGURUAN TINGGI
MENGUNAKAN METODE *LABELED LATENT
DIRICHLET ALLOCATION* (L-LDA) UNTUK MENGUKUR
TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN
TINGGI DI INDONESIA**

**Eko Putro Fitrianto
NRP 5113 100 194**

**Dosen Pembimbing I
Prof. Drs. Ec. Ir. Rianarto Sarno, M.Sc., Ph.D.**

**Dosen Pembimbing II
Bagus Setya Rintyarna, S.T., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

UNIVERSITY RANKING USING *LABELED LATENT DIRICHLET ALLOCATION* (L-LDA) METHOD TO MEASURE TECHNOLOGY READINESS LEVEL OF UNIVERSITY IN INDONESIA

**Eko Putro Fitrianto
NRP 5113 100 194**

**Supervisor I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Supervisor II
Bagus Setya Rintyarna, S.T., M.Kom.**

**INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PEMERINGKATAN PERGURUAN TINGGI MENGGUNAKAN METODE *LABELED LATENT DIRICHLET ALLOCATION (L-LDA)* UNTUK MENGUKUR TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN TINGGI DI INDONESIA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

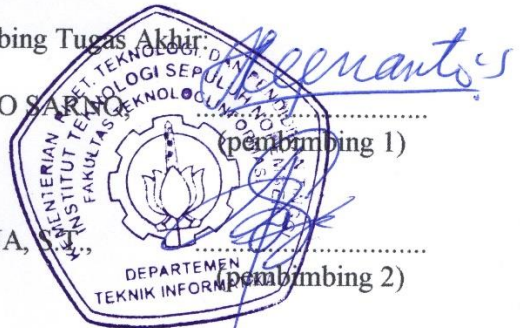
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

EKO PUTRO FITRIANTO
NRP. 5113 100 194

Disetujui oleh Dosen Pembimbing Tugas Akhir

Prof. Drs. Ec. Ir. RIYANARTO SARINDO
M.Sc., Ph.D.
NIP: 19590803 198601 1 001



(pembimbing 1)

BAGUS SETYA RINTYARNA,
M.Kom.
NIK: 0509502

(pembimbing 2)

SURABAYA
JULI, 2017

[Halaman ini sengaja dikosongkan]

**PEMERINGKATAN PERGURUAN TINGGI
MENGUNAKAN METODE *LABELED LATENT
DIRICHLET ALLOCATION (L-LDA)* UNTUK
MENGUKUR TINGKAT KESIAPTERAPAN
TEKNOLOGI PERGURUAN TINGGI DI
INDONESIA**

Nama : Eko Putro Fitrianto
NRP : 5113100194
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc.,Ph.D.
Dosen Pembimbing II : Bagus Setya Rintyarna, S.T., M.Kom.

Abstrak

Pemeringkatan perguruan tinggi merupakan sebuah cara untuk mengukur kualitas dan kesiapan perguruan tinggi dalam menjalankan proses pembelajaran kepada mahasiswa yang dilihat dari berbagai macam kriteria penilaian. Pemeringkatan perguruan tinggi menjadi salah satu hal yang penting untuk mengukur reputasi perguruan tinggi di dunia internasional maupun di dalam negeri. Semakin baik reputasi, maka akan semakin baik kualitas pendidikan perguruan tinggi tersebut. Aspek kriteria penilaian bergantung pada berbagai macam penilaian yang dapat disimpulkan menjadi dua jenis yaitu kualitatif dan kuantitatif. Kriteria penilaian yang akan ditentukan harus mampu dijangkau oleh perguruan tinggi agar menghasilkan pemeringkatan yang seimbang dan ideal.

Dalam Tugas Akhir ini diusulkan sebuah gagasan baru sistem pemeringkatan perguruan tinggi di Indonesia menggunakan metode Labeled Latent Dirichlet Allocation (L-LDA) untuk mengoptimalkan kriteria penilaian yang bersifat kualitatif, yaitu kriteria reputasi akademik dengan menggunakan metode topik modeling untuk menggali topik utama dari dokumen

yang berisi abstrak penelitian akademisi di setiap perguruan tinggi, yang sebelumnya dikerjakan secara manual dengan melakukan survey oleh manusia. Menggunakan 800 dokumen abstrak yang didapatkan dari Google Scholar, 9 corpus kata tingkat kesiapterapan teknologi untuk dataset kriteria yang bersifat kualitatif, serta basis data online dari Science and Techonogy Index (SINTA) dan pangkalan data PDDIKTI untuk dataset kriteria yang bersifat kuantitatif. Pengoptimalan kriteria kualitatif bertujuan untuk menghasilkan hasil pemeringkatan yang efisien dan sedernaha dari yang sebelumnya.

Hasil implementasi sistem pemeringkatan Perguruan Tinggi menunjukkan bahwa kinerja sistem secara keseluruhan telah mendekati Ground Truth yaitu pemeringkatan QS University Rankings dengan tingkat similarity sistem sebesar 78,3%, serta tingkat distance metode L-LDA dengan Ground Truth sebesar 20 dan tingkat keberhasilan peringkat dalam toleransi sebesar 66, 67%. Dengan hasil ini menunjukkan bahwa sistem telah berhasil melakukan optimasi kriteria kualitatif reputasi akademik. Diharapkan pengembangan proses pemeringkatan selanjutnya dapat mengacu pada prinsip optimasi kriteria yang bersifat kualitatif serta mengoptimalkan kembali proses topik modeling agar menjadi lebih efisien dalam menghemat waktu kerja dan biaya serta mendapatkan hasil yang akurat.

Kata kunci: Kualitas Perguruan Tinggi, Kualitatif, Kuantitatif, Labeled Latent Dirichlet Allocation, L-LDA, Optimasi, Pemeringkatan Perguruan Tinggi, QS University Rangking, Reputasi

UNIVERSITY RANGKING USING *LABELED LATENT DIRICHLET ALLOCATION (L-LDA)* METHOD TO MEASURE TECHNOLOGY READINESS LEVEL OF UNIVERSITY IN INDONESIA

Student Name : Eko Putri Fitrianto
NRP : 5113100194
Major : Informatics Department – FTIf ITS
Supervisor I : Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Supervisor II : Bagus Setya Rintyarna, S.T., M.Kom.

Abstract

University Ranking is a way to measure quality and readiness of college to apply their learning process to college student that seen from some criteria assessment. Universities ranking be important to measure college reputation in the outside world and within the country. With a good reputation, then can show if quality college is well. Not only that, but can become reflection of quality national education in the international outlook. Aspect of assessment relies variety of assessment that can be summarized into two types that is qualitative and quantitative. Assessment aspect specified should be able to be reached by the college in order to produce a balanced rankings.

In this final project proposed a new idea for rating system universities in Indonesia using Labeled Latent Dirichlet Allocation (L-LDA) method to optimize the assessment criteria are qualitative, in order to generate the rating more efficient and simple. In addition, to measure the application level technology in university. Using 800 abstract documents obtained from Google Scholar, 9 corpus words technology readiness level (TKT) for qualitative criteria datasets, as well as an online database of Science and Techonogy Index (SINTA) and PDDIKTI

database for quantitative criteria datasets. Optimization of qualitative criteria aims to produce more efficient and more efficient results than ever before. In addition, measuring the level of technological readiness in Higher Education to run the learning process.

The implementation of university ranking system shows that the result is close to the Ground Truth, which is QS University Rankings ranking with similarity value is 78,33%, gap distance method with Ground Truth is 20, and tolerance of difference rank is 66,67%. With these results indicate that the system has successfully optimized the qualitative criteria of academic reputation with some notes in the topic modeling method as an evaluation material in future development to approximate existing results. It is expected that the development of the rating process can then refer to the principle of optimization of qualitative criteria to be more efficient and simpler in saving work time.

Keywords: Labeled Latent Dirichlet Allocation, L-LDA, Optimation, University Rankings, University Quality, QS University Rangings, Qualitative, Quantitative, Reputation, Technology Readiness Level

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena atas berkat rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Pemerinkatan Perguruan Tinggi Menggunakan Metode Labeled Latent Dirichlet Allocation (L-LDA) Untuk Mengukur Tingkat Kesiapterapan Teknologi Perguruan Tinggi Di Indonesia”

Pengerjaan Tugas Akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku Tugas Akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak, ibu, adik dan keluarga yang selalu memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis untuk menyelesaikan Tugas Akhir ini.
2. Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D. dan bapak Bagus Setya Rintyarna selaku dosen pembimbing yang telah memberi ide dan bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTif ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.

5. Segenap dosen rumpun mata kuliah Manajemen Informasi yang telah banyak memberikan ilmu dalam bidang RMK penulis.
6. Rekan-rekan mahasiswa S2 Teknik Informatika yang telah banyak meluangkan waktu dan bersedia membantu penulis dalam memahami pengerjaan Tugas Akhir ini.
7. Rekan-rekan pengerjaan satu tim, yaitu Donny, Gerry, dan Arga dalam penyelesaian Tugas Akhir dengan nama “IDN University Ranking”.
8. Rekan-rekan seperjuangan anak didik Tugas Akhir Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D. yaitu Zaenal, Renanda, Ichal, Faisal Bhaswara, Andi, Hariyanto, Dhiaz, Fidi, dan Ponigh.
9. Teman-teman Teknik Informatika Angkatan 2013 yang selalu mendukung, menyemangati, membantu dan mendengarkan suka duka selama proses pengerjaan Tugas Akhir.
10. Serta semua pihak yang turut membantu penulis dalam menyelesaikan Tugas Akhir ini yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan Tugas Akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan Tugas Akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan

Surabaya, Juli 2017

Eko Putro Fitrianto

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
Abstrak.....	ix
<i>Abstract</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL.....	xxiii
DAFTAR KODE SUMBER.....	xxvii
1 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.....	5
1.6 Metodologi.....	5
1.7 Sistematika Penulisan.....	7
2 BAB II DASAR TEORI.....	11
2.1 <i>University Rankings</i>	11
2.2 <i>QS World University Rankings</i>	11
2.3 Teks Mining.....	18
2.4 Teknik Grabbing.....	19
2.4.1 Fungsi cURL.....	20
2.4.2 Fungsi File_get_contents.....	20
2.4.3 Simple HTML Dom Parser.....	21

2.4.4	Regular Expression (ReGex)	22
2.5	Pusat Data <i>Scopus</i>	22
2.6	<i>Google Scholar</i>	23
2.7	<i>Science and Technology Index (SINTA)</i>	24
2.8	Pangkalan Data Pendidikan Tinggi (PDDIKTI)	25
2.9	<i>Alexa Traffict Rank</i>	26
2.10	<i>MOZ Analytics</i>	26
2.11	Bahasa Pemrograman Python	27
2.12	Library Python	28
2.13	PHP dan HTML	32
2.14	<i>Database Management System (DBMS) MySQL</i>	33
2.15	<i>Part-Of-Speech Tagger (POS Tagger)</i>	33
2.16	<i>Bloom Taxonomy</i>	35
2.17	Metode <i>Labeled Latent Dirichlet Allocation (LLDA)</i>	36
2.18	Macam-macam Pengembangan Metode LDA	42
2.19	<i>Term Frequency – Invers Document Frequency (TF-IDF)</i>	43
2.19.1	Term Frequency (TF)	43
2.19.2	Invers Documen Frequency (IDF)	43
2.20	Fase Teks Mining (Metode L-LDA) Studi Kasus Pemeringkatan Perguruan Tinggi	44
2.21	Akreditasi Perguruan Tinggi	48
2.22	<i>Tingkat Kesiapterapan Teknologi (TKT)</i>	49
3	BAB III ANALISIS DAN PERANCANGAN	53
3.1.	Tahap Analisis	53
3.1.1	Deskripsi Umum	53
3.1.2	Aktor Dalam Sitem	54

3.1.3	Spesifikasi Kebutuhan Sistem.....	54
3.1.4	Kasus Penggunaan.....	55
3.1.5	Conceptual Data Model (CDM).....	66
3.1.6	Physical Data Model (PDM).....	67
3.1.7	Label Kelas dalam Teks Mining Metode L-LDA	68
3.1.8	Ground Truth Pengujian Sistem	69
3.1.9	Dataset Pengujian Sistem.....	73
3.2	Tahap Perancangan	77
3.2.1	Perancangan Basis Data.....	77
3.2.2	Perancangan Antarmuka	86
3.2.3	Perancangan Kriteria Penilaian Perguruan Tinggi	89
3.2.4	Proses Sistem Aplikasi Pengguna (Diagram Aplikasi)	95
3.2.5	Proses Kerja Sistem Secara Keseluruhan (Flowchart Sistem)	96
3.2.6	Proses Teks Mining Metode L-LDA (Flowchart Metode L-LDA).....	97
4	BAB IV IMPLEMENTASI SISTEM.....	99
4.1	Lingkungan Impelementasi	99
4.2	Implementasi Syntax Basis Data	100
4.2.1	Create Database “databse_qs_rank”	100
4.2.2	Create Table “universitas”	100
4.2.3	Create Table “reputasi_akademik”	101
4.2.4	Create Tabel “akreditasi”	101
4.2.5	Create Tabel “jumlah_mahasiswa”	101
4.2.6	Create Tabel “jumlah_tenaga_pengajar_jabatan_ akademik”	102

4.2.7	Create Tabel “jumlah_tenaga_pengajar_tetap”	103
4.2.8	Create Tabel “jumlah_tenaga_pengajar_tidak_tetap”	103
4.2.9	Create Tabel “jumlah_sitasi_sinta”	104
4.2.10	Create Tabel “jumlah_author_sinta”	105
4.2.11	Create Tabel “precense”	105
4.2.12	Create Tabel “impact”	106
4.2.13	Create Tabel “alexa_rank”	106
4.2.14	Create Tabel “kumpulan_skor_final”	107
4.2.15	Create Tabel “4icu”	108
4.2.16	Create Tabel “dikti”	108
4.2.17	Create Tabel “qs_general”	108
4.2.18	Create Tabel “qs_asia”	109
4.2.19	Create Tabel “qs_llda”	109
4.2.20	Create Tabel “webometrics”	109
4.3	Implementasi Pseudocode Proses Teks Mining (Metode L-LDA).....	110
4.3.1	Teks Preprocessing	110
4.3.2	Memperkaya Corpus Keyword TKT dengan Teknik Sinonym Word (WordNet) dalam Level 1, 2, 3, dan all	111
4.3.3	Memperkaya Corpus Keyword TKT dengan Teknik POS Tagging.....	112
4.3.4	Update Konten Kata Corpus Keyword TKT	112
4.3.5	Menentukan Asumsi Label Topik Awal	115
4.3.6	Pembobotan Skor Penilaian.....	116
4.4	Implementasi Antarmuka Pengguna.....	118

4.4.1	Antarmuka Home (index)	118
4.4.2	Antarmuka Mencari Perguruan Tinggi Berdasarkan Potongan Nama	118
4.4.3	Antarmuka Hasil Pencarian Perguruan Tinggi Berdasarkan Keyword Potongan Nama	119
4.4.4	Antarmuka Melihat Detail Informasi Perguruan Tinggi	119
4.4.5	Antarmuka Melihat Latar Belakang Sistem Pemingkatan	120
4.4.6	Antarmuka Melihat Informasi Metode Penilaian	120
4.4.7	Antarmuka Pencarian Peringkat Perguruan Tinggi Berdasarkan Region	122
4.4.8	Antarmuka Hasil Pencarian Perguruan Tinggi Berdasar Region Wilayah Perguruan Tinggi	122
4.4.9	Antarmuka Pembobotan Pemingkatan Perguruan Tinggi	123
4.4.10	Antarmuka Hasil Pemingkatan Perguruan Tinggi dengan Berbagai Macam Tipe Pemingkatan	123
4.4.11	Antarmuka Error Handling Pembobotan Kriteria Pemingkatan	126
5	BAB V PENGUJIAN DAN EVALUASI	127
5.1	Lingkungan Uji Coba	127
5.2	Data Uji Coba	127
5.2.1	Data Dokumen Abstrak Paper Akademisi	127
5.2.2	Data Corpus Taxonomy Bloom	130
5.2.3	Data Corpus Tingkat Kesiapterapan Teknologi (TKT)	131
5.2.4	Rekap Jumlah Mahasiswa dan Tenaga Pengajar	135

5.2.5	Data Online Science and Technologu Index (SINTA)	137
5.2.6	Data Online Analisa Webhost	139
5.3	Skenario Pengujian	141
5.3.1	Skenario Pengambilan Data Online Menggunakan Teknik <i>Grabbing</i>	141
5.3.2	Skenario Pengambilan Data pada Tabel Database ..	146
5.3.3	Skenario Pengujian Dataset Menggunakan Teks Minning Metode L-LDA	150
5.3.4	Skenario Pengujian Sitem Berdasarkan Hasil Pembobotan Akhir Sesuai <i>Ground Truth QS Worlds University Rankings</i>	181
5.4	Evaluasi.....	182
6	BAB VI KESIMPULAN DAN SARAN.....	185
6.1	Kesimpulan	185
6.2	Saran	186
	DAFTAR PUSTAKA	187
	LAMPIRAN.....	191
	BIODATA PENULIS	229

DAFTAR GAMBAR

Gambar 2.3-1 Proses Utama Teks Mining.....	19
Gambar 2.17-1 Diagram Representasi Model Generatif LDA....	37
Gambar 2.17-2 Diagram model generativ L-LDA.....	41
Gambar 2.22-1 Hirarki Tingkat Kesiapterapan Teknologi (TKT)	50
Gambar 1-1 Diagram Kasus Penggunaan	56
Gambar 1-2 Diagram Aktifitas Use Case UC-001.....	59
Gambar 1-3 Diagram Aktifitas Use Case UC-002.....	61
Gambar 1-4 Diagram Aktifitas Use Case UC-003.....	62
Gambar 1-5 Diagram Aktifitas Use Case UC-004.....	63
Gambar 1-6 Diagram Aktifitas Use Case UC-005.....	65
Gambar 1-7 Conceptual Data Model.....	66
Gambar 1-8 Physical Data Model.....	67
Gambar 3.2-1 Alur Menentukan Reputasi Akademik.....	90
Gambar 3.2-2 Alur Menentukan Skor Akreditasi	91
Gambar 3.2-3 Alur Menentukan Skor Rasio Mahasiswa dan Tenaga Penagajar Aktif	92
Gambar 3.2-4 Alur Menentukan Skor Jumlah Sitasi Akademisi Penelitian.....	93
Gambar 3.2-5 Diagram Aplikasi	95
Gambar 3.2-6 Flowchart Sistem Keseluruhan	96
Gambar 3.2-7 Flowchart Metode L-LDA.....	97
Gambar 4.4-1 Antarmuka Home	118
Gambar 4.4-2 Antarmuka Pencarian Nama	118
Gambar 4.4-3 Antarmuka Hasil Pencarian	119
Gambar 4.4-4 Antarmuka Detail Informasi	119
Gambar 4.4-5 Antarmuka Informasi Latar Belakang.....	120
Gambar 4.4-6 Antarmuka Informasi Metode Penilaian	120
Gambar 4.4-7 Antarmuka Metode Penilaian QS General	121
Gambar 4.4-8 Antarmuka Metode Penilaian QS Asia	121
Gambar 4.4-9 Antarmuka Metode Penilaian Webometrics	121
Gambar 4.4-10 Antarmuka Pencarian Berdasar Region	122
Gambar 4.4-11 Antarmuka Hasil Pencarian Berdasar Region ..	122
Gambar 4.4-12 Antarmuka Pembobotan Pemingkatan.....	123

Gambar 4.4-13 Antarmuka Hasil Pemeringkatan QS General .	124
Gambar 4.4-14 Antarmuka Hasil Pemeringkatan QS Asia.....	124
Gambar 4.4-15 Antarmuka Hasil Pemeringkatan Webometrics	124
Gambar 4.4-16 Antarmuka Hasil Pemeringkatan 4ICU	125
Gambar 4.4-17 Antarmuka Hasil Pemeringkatan DIKTI	125
Gambar 4.4-18 Antarmuka Hasil Peringkat QS LLDA	125
Gambar 4.4-19 Antarmuka Error Handling Pembobotan	126
Gambar 5.3-1 Antarmuka Pembobotan Peringkat	181
Gambar 5.3-2 Antarmuka Tombol Cek Pemeringkatan	182
Gambar 5.3-3 Antarmuka Hasil Pemeringkatan	182

DAFTAR TABEL

Tabel 2.15-1 Struktur Kelas kata POS Tagging.....	34
Tabel 1-1 Daftar Kebutuhan Fungsional Sistem.....	54
Tabel 1-2 Daftar Kebutuhan Fungsional Sistem.....	56
Tabel 1-3 Case Memeasukkan Presentase Bobot Kriteria Penilaian Pemingkatan	57
Tabel 1-4 Use Case Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemingkatan	60
Tabel 1-5 Use Case Mencari Perguruan Tinggi Berdasarkan Potongan Nama	61
Tabel 1-6 Use Case Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi.....	62
Tabel 1-7 Use Case Melihat Detail Informasi Utama Perguruan Tinggi.....	63
Tabel 1-8 Penjelasan Setiap Kelas Label.....	68
Tabel 1-9 Peringkat QS World University Rankings 2015-2016	69
Tabel 1-10 Peringkat Reputasi Akademik QS 2015-2016.....	70
Tabel 1-11 Peringkat QS World University Rankings 2016-2017	70
Tabel 1-12 Peringkat Reputasi Akademik QS 2016-2017	71
Tabel 1-13 Peringkat QS World University Rankings 2017-2018	71
Tabel 1-14 Peringkat Reputasi Akademik QS 2017-2018.....	72
Tabel 1-15 Contoh Hasil Analisa Manual Sebagai Ground Truth Reputasi Akademik	73
Tabel 1-16 Daftar Rincian Jumlah Kata Dataset Abstrak Paper Perguruan Tinggi.....	74
Tabel 1-17 Daftar Rincian Jumlah Kata Corpus Taxonomy Bloom	75
Tabel 1-18 Perbandingan Jumlah Kata Corpus TKT	76
Tabel 5.2-1 Peringkat Perguruan Tinggi.....	128
Tabel 5.2-2 Daftar Reduksi Jumlah Kata Dataset.....	129
Tabel 5.2-3 Rincian Jumlah Kata Corpus Taxonomy Bloom ...	130
Tabel 5.2-4 Penggabungan Referensi Corpus Taxonomy Bloom	131

Tabel 5.2-5 Hasil Breakdown Menjadi 9 Corpus TKT.....	132
Tabel 5.2-6 Hasil Penambahan Kata Corpus TKT Menggunakan POS Tagging	133
Tabel 5.2-7 Hasil Penambahan Kata Corpus TKT Menggunakan Sinonim Word.....	134
Tabel 5.2-8 Daftar Jumlah Perguruan Tinggi Berdasarkan Jenis Wilayah Administrasi.....	136
Tabel 5.3-1 Hasil Reduksi Kata Tahap Preprocessing.....	150
Tabel 5.3-2 Tingkat Duplikasi Kata	152
Tabel 5.3-3 Rincian Jumlah Penambahan Kata dengan POS Tagging Setiap Perguruan Tinggi.....	153
Tabel 5.3-4 Rincian Jumlah Breakdown Hasil POS Tagging pada Corpus TKT	155
Tabel 5.3-5 Corpus TKT Final Berdasarkan Percobaan POS Tagging	156
Tabel 5.3-6 Analisa Kata Hasil Akuisisi POS Tagging 1 (-1, +1)	156
Tabel 5.3-7 Analisa Kata Hasil Akuisisi POS Tagging 2 (-2, +2)	157
Tabel 5.3-8 Analisa Kata Hasil Akuisisi POS Tagging 3 (-3, +3)	158
Tabel 5.3-9 Analisa Kata Hasil Akuisisi POS Tagging 4 (-4, +4)	158
Tabel 5.3-10 Analisa Kata Hasil Akuisisi POS Tagging 8 (-8, +8)	159
Tabel 5.3-11 Analisa Kata Hasil Akuisisi POS Tagging 10 (-10, +10).....	160
Tabel 5.3-12 Label Dokumen Hasil Percobaan Teknik POS Tagging Akuisisi 1	161
Tabel 5.3-13 Label Dokumen Hasil Percobaan Tanpa Menggunakan Teknik POS Tagging	162
Tabel 5.3-14 Label Dokumen Hasil Percobaan Sinonim Word Level 1	163
Tabel 5.3-15 Label Dokumen Hasil Percobaan Sinonim Word Level 2	164

Tabel 5.3-16 Label Dokumen Hasil Percobaan Sinonim Word Level 3	164
Tabel 5.3-17 Label Dokumen Hasil Percobaan Sinonim Word Level All	165
Tabel 5.3-18 Probabilitas Label dengan Dokumen dalam Percobaan 1	166
Tabel 5.3-19 Probabilitas Label dengan Dokumen dalam Percobaan 2	167
Tabel 5.3-20 Probabilitas Label dengan Dokumen dalam Percobaan 3	167
Tabel 5.3-21 Probabilitas Label dengan Dokumen dalam Percobaan 4	168
Tabel 5.3-22 Probabilitas Label dengan Dokumen dalam Percobaan 5	169
Tabel 5.3-23 Probabilitas Label dengan Dokumen dalam Percobaan 6	170
Tabel 5.3-24 Topik Akhir Dokumen Menggunakan Percobaan Teknik POS Tagging Akuisisi 1	170
Tabel 5.3-25 Topik Akhir Dokumen Menggunakan Percobaan Tanpa Teknik POS Tagging	171
Tabel 5.3-26 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level 1	172
Tabel 5.3-27 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level 2	172
Tabel 5.3-28 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level 3	173
Tabel 5.3-29 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level All	174
Tabel 5.3-30 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 1	174
Tabel 5.3-31 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 2	175
Tabel 5.3-32 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 3	175

Tabel 5.3-33 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 4	176
Tabel 5.3-34 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 5	176
Tabel 5.3-35 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 6	177
Tabel 5.3-36 Hasil Analisa Semua Percobaan dengan Ground Truth QS World University Rankings 2015-2016	178
Tabel 5.3-37 Hasil Analisa Semua Percobaan dengan Ground Truth QS World University Rankings 2016-2017	179
Tabel 5.3-38 Hasil Analisa Semua Percobaan dengan Ground Truth QS World University Rankings 2017-2018	180
Tabel 5.4-1 Rangkuman Hasil Pengujian	182

DAFTAR KODE SUMBER

Kode Sumber 2.4-1 Syntax cURL dengan setting option url.....	20
Kode Sumber 2.4-2 Syntax Fungsi file_get_contents	21
Kode Sumber 2.4-3 Syntax Penggunaan Library Simple HTML Dom Parser.....	21
Kode Sumber 4.2-1 Kode Sumber Create Database	100
Kode Sumber 4.2-2 Kode Sumber Create Table universitas.....	100
Kode Sumber 4.2-3 Create Tabel reputasi_akademik.....	101
Kode Sumber 4.2-4 Create Tabel akreditasi	101
Kode Sumber 4.2-5 Create Table jumlah_mahasiswa	102
Kode Sumber 4.2-6 Create Table jumlah_tenaga_pengajar_ jabatan_akademik.....	102
Kode Sumber 4.2-7 Create Table jumlah_tenaga_pengajar_tetap	103
Kode Sumber 4.2-8 Create Table jumlah_tenaga_pengajar_tidak _tetap.....	104
Kode Sumber 4.2-9 Create Table jumlah_sitasi_sinta	104
Kode Sumber 4.2-10 Create Table jumlah_author_sinta	105
Kode Sumber 4.2-11 Create Table precense.....	105
Kode Sumber 4.2-12 Create Table impact.....	106
Kode Sumber 4.2-13 Create Tabel alexa_rank	106
Kode Sumber 4.2-14 Create Table kumpulan_skor_final	107
Kode Sumber 4.2-15 Create Table kumpulan_skor_final	107
Kode Sumber 4.2-16 Create Table 4icu.....	108
Kode Sumber 4.2-17 Create Table dikti	108
Kode Sumber 4.2-18 Create Table qs_general	108
Kode Sumber 4.2-19 Create Table qs_asia.....	109
Kode Sumber 4.2-20 Create Table qs_llda	109
Kode Sumber 4.2-21 Create Table webometrics	109
Kode Sumber 4.3-1 Psudocode Teks Preprocessing.....	110
Kode Sumber 4.3-2 Psudocode Memperkaya Corpus Keyword TKT Dengan Synonym Word.....	111
Kode Sumber 4.3-3 Psudocode Memperkaya Corpus Keyword TKT dengan POS Tagging	112

Kode Sumber 4.3-4 Psudocode Update Konten Kata Corpus Keyword TKT	114
Kode Sumber 4.3-5 Menentukan Asumsi Label Topik Awal...	115
Kode Sumber 4.3-6 Pembobotan Skor Penilaian.....	117
Kode Sumber 6.2-1 Kode Sumber Text Preprocessing	202
Kode Sumber 6.2-2 Kode Sumber Memberkaya Corpus TKT .	209
Kode Sumber 6.2-3 Kode Sumber Pencocokan Term Asumsi Label Awal.....	216
Kode Sumber 6.2-4 Kode Sumber Metode L-LDA	224
Kode Sumber 6.2-5 Kode Sumber Pembobotan Skor Reputasi Akademik.....	227
Kode Sumber 6.2-6	227

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari dalam Tugas Akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Perguruan tinggi merupakan sebuah institusi pendidikan penyelenggara pendidikan tinggi. Pendidikan tinggi menjadi hal penting karena mencerminkan kondisi intelektual masyarakat dari sebuah negara. Membutuhkan perhatian dan evaluasi lebih dari pemerintah untuk dapat menghasilkan sumberdaya manusia yang baik dari perguruan tinggi. Hal ini tentunya untuk menghasilkan generasi yang mempunyai intelektual dan keterampilan tinggi ketika bersaing di dunia kerja kelak [1]. Kualitas pendidikan tinggi yang baik menjadi prioritas dan pusat perhatian pendidikan internasional. Pencapaian tujuan strategis perguruan tinggi bergantung pada faktor konstektual, yang meliputi kepemimpinan suatu negara, ideologi negara, finansial, sosial budaya, komunikasi dan bahasa, serta proses pengambilan keputusan dari pemerintahan. Selain itu perguruan tinggi juga didorong oleh norma yang terkait dengan lingkungan dan nilai pendidikan itu sendiri [1].

Selain itu perguruan tinggi menjadi salah satu sarana yang efektif untuk mengembangkan ilmu pengetahuan dan teknologi dengan berbagai cara, salah satu cara dengan melakukan penelitian. Penelitian merupakan salah satu wadah kreatifitas dan pengembangan potensi akademik bagi mahasiswa. Selain itu penelitian diselenggarakan untuk membuat seorang akademisi dapat menghasilkan sebuah karya dalam dunia pendidikan [1]. Jumlah populasi masyarakat suatu negara yang banyak maka harus berbanding lurus dengan pengembangan intelektual dan keterampilan generasi muda, sebagai investasi negara dalam dunia pendidikan.

Salah satu upaya untuk meningkatkan kualitas perguruan tinggi adalah dengan melihat hasil pemeringkatan perguruan tinggi di seluruh dunia [2]. Perguruan tinggi yang masuk dalam peringkat dunia akan menjadi acuan dan fokusannya bahwa kualitas pendidikan di negara tersebut telah bagus, dari berbagai segi kriteria. Ada banyak media publikasi pemeringkatan perguruan tinggi, salah satunya adalah publikasi tahunan *QS World University Rankings* [3]. Dalam pemeringkatan perguruan tinggi, tentunya ada banyak kriteria penilaian yang menjadikan perguruan tinggi tersebut berkualitas dan layak untuk menempati peringkat dalam daftar.

Tingkat kualitas pendidikan di Indonesia sejauh ini dapat dikatakan kurang baik. Indonesia masih berfokus pada jumlah partisipasi pendidikan, tidak pada pembangunan kualitas pendidikan. Berdasarkan posisi peringkat publikasi tahunan *QS World University Rankings*, perguruan tinggi di Indonesia masih berada pada urutan 67 regional Asia yang ditempati oleh Universitas Indonesia (UI). Berbeda jauh jika dibandingkan dengan negara tetangga, yaitu Singapura yang mempunyai jumlah penduduk lebih sedikit dari Indonesia berhasil berada pada peringkat 1 region Asia yang ditempati oleh National University of Singapore (NUS) [4]. Di Indonesia sendiri kualitas pendidikan yang menonjol hanya berada pada perguruan tinggi tertentu saja, hal ini yang menyebabkan kualitas pendidikan di Indonesia kurang merata.

Salah satu cara untuk mengetahui merata atau tidaknya kualitas pendidikan tinggi di Indonesia dengan melihat Tingkat Kesiapterapan Teknologi (TKT) dari perguruan tinggi. TKT menggambarkan tingkat kematangan perguruan tinggi dalam hal penelitian dan pengembangan teknologi [5]. Tujuan utama dari penelitian secara umum sejalan dengan tujuan perguruan tinggi itu sendiri, sebagai sarana yang efektif untuk mengembangkan ilmu pengetahuan dan pengembangan potensi sumberdaya manusia.

Kembali kepada publikasi tahunan *QS World University Rankings* terdapat berbagai macam kriteria penilaian yang secara umum menggambarkan kondisi perguruan tinggi [3], yang mana dapat menjadi *value* untuk meningkatkan tingkat kesiapterapan (TKT) dari perguruan tinggi. Kriteria kualitatif dalam *QS World University Rankings* menjadi *value* yang paling berpengaruh dalam melihat kualitas perguruan tinggi. Kriteria ini didapatkan dengan proses manual yang teliti dan akurat terhadap hasil, sehingga membutuhkan waktu yang lama untuk menghasilkan pemeringkatan yang ideal [3]. Salah satu pilihan untuk mengoptimasi kriteria kualitatif adalah dengan menerapkan metode kecerdasan komputasional berupa metode teks mining *Labeled Latent Dirichlet Allocation (L-LDA)* [6].

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengumpulkan data masukan sebagai bahan penilaian kriteria perguruan tinggi?
2. Bagaimana menghasilkan skor akhir dari kriteria penilaian perguruan tinggi yang bersifat kualitatif menggunakan topic modelling metode *Labeled Latent Dirichlet Allocation (L-LDA)*?
3. Bagaimana menghitung skor akhir dari data kriteria penilaian perguruan tinggi yang bersifat kuantitatif?
4. Bagaimana menentukan skor akhir secara keseluruhan untuk sebuah perguruan tinggi?
5. Bagaimana menganalisa hasil keluaran sistem dengan *Ground Truth QS University Rankings* yang sudah ditetapkan sehingga menghasilkan hasil akurasi?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan sebagai berikut:

1. Bahasa pemrograman yang digunakan adalah Python untuk proses *teks mining* dokumen abstrak *paper*, PHP

- dan HTML untuk membangun sistem *grabbing* dan membangun desain antarmuka.
2. Database management sistem (DBMS) yang digunakan adalah MySQL database.
 3. Perguruan tinggi yang dinilai hanya perguruan tinggi dalam wilayah negara Indonesia.
 4. Menggunakan acuan kriteria dari media publikasi tahunan *QS World University Rankings*.
 5. Data sebagai dasar daftar perguruan tinggi, data total mahasiswa, data total tenaga kerja perguruan tinggi berdasarkan data pada PDDIKTI.
 6. Data total publikasi akademisi sebagai salah satu kriteria penilaian kuantitatif berdasarkan database *Science and Technology Index* (SINTA) yang mengadopsi database dari Scopus dan Google Scholar.
 7. Data akreditasi perguruan tinggi berdasarkan data dari Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT).
 8. Metode *Labeled Latent Dirichlet Allocation* (L-LDA) untuk mengidentifikasi topik dari dokumen abstrak *paper* akademisi perguruan tinggi yang akan menjadi skor nilai reputasi akademik yang bersifat kualitatif.
 9. Menggunakan 50 dataset berupa dokumen abstrak *paper* akademisi dari masing-masing perguruan tinggi dengan kategori rumpun teknik.
 10. Menggunakan corpus *Taxonomy Bloom* sebagai dasar penetapan corpus kata (*term*) untuk setiap level Tingkat Kesiapterapan Teknologi (TKT).

1.4 Tujuan

- Tujuan dari pembuatan Tugas Akhir ini sebagai berikut:
1. Mampu menerapkan teknik *grabbing* terhadap data online berupa informasi *numeric* sebagai masukan kriteria penilaian perguruan tinggi yang bersifat kuantitatif.
 2. Mampu menerapkan metode *Labeled Latent Dirichlet Allocation* (L-LDA) untuk memodelkan topik dari sebuah

- dokumen abstrak paper akademisi perguruan tinggi sebagai masukan kriteria penilaian yang bersifat kualitatif.
3. Mampu menghasilkan skor dari setiap kriteria penilaian perguruan tinggi, serta skor final penilaian secara keseluruhan/kumulatif dari setiap kriteria.
 4. Mampu menganalisa hasil keluaran sistem pemeringkatan dengan *Ground Truth QS World University Rankings*.
 5. Mampu menyajikan informasi hasil pemeringkatan perguruan tinggi dalam antarmuka yang jelas dan mudah untuk dimengerti dan diakses oleh pengguna perangkat lunak website (*End User*).

1.5 Manfaat

Tugas Akhir ini diharapkan dapat membantu menghasilkan pemeringkatan perguruan tinggi di Indonesia yang mudah, efektif, dan sederhana berdasarkan Tingkat Kesiapterapan Teknologi (TKT) yang menjadi dasar pemetaan tingkat kematangan teknologi perguruan tinggi di Indonesia. Mengoptimasi kriteria diharapkan mampu menekan waktu kerja (*running time*) dan biaya (*cost*) untuk proses pemeringkatan perguruan tinggi.

Penilaian kriteria yang dibahas mengacu pada publikasi pemeringkatan perguruan tinggi internasional tahunan *QS World University Rankings* yang dapat digunakan sebagai tolok ukur kualitas pendidikan tinggi di Indonesia.

1.6 Metodologi

Tahapan-tahapan yang ditempuh dalam pengerjaan Tugas Akhir ini sebagai berikut:

1. Studi literatur

Pada tahap ini dilakukan studi literatur dengan mencari referensi yang berfokus pada pembahasan teknik *grabbing* data online dari website yang berfungsi sebagai penyedia informasi kriteria penilaian yang bersifat kuantitatif, mencari referensi proses teks mining metode *Labeled Latent Dirichlet Allocation (L-LDA)* untuk proses

topik modelling dokumen abstrak *paper* akademisi perguruan tinggi yang menjadi salah satu kriteria penilaian yang mempunyai bobot ketergantungan paling tinggi, serta mencari referensi literatur lain yang membahas mengenai dasar-dasar dan proses pemeringkatan perguruan tinggi yang sudah ada sebelumnya.

2. Analisis dan Perancangan Sistem

Pada tahap ini dijabarkan mengenai proses diagram kasus sistem (*use case*) dan aktor yang terlibat dalam alur sistem. Aktor yang menjadi pelaku dalam sistem adalah pengguna perangkat lunak website (*end user*). Kemudian beberapa kebutuhan fungsional dari sistem ini antara lain:

- i. Memasukkan prosentase bobot kriteria penilaian pemeringkatan.
- ii. Melihat peringkat perguruan tinggi berdasarkan tipe pemeringkatan.
- iii. Mencari perguruan tinggi berdasarkan potongan nama perguruan tinggi.
- iv. Mencari perguruan tinggi berdasarkan region wilayah provinsi.
- v. Melihat detail informasi akademik perguruan tinggi.

3. Implementasi Sistem

Pada tahap ini dilakukan pembuatan perangkat lunak untuk sistem server (*developer*) berupa penerapan teknik *grabbing* pada pengambilan data online, proses teks mining menggunakan metode *Labeled Latend Dirichlet Allocation* (L-LDA) untuk mengganti kriteria reputasi akademik yang sebelumnya dikerjakan secara manual, serta membangun database dan antarmuka untuk menampilkan informasi akhir kepada pengguna.

4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap perangkat lunak dengan menggunakan dataset final berupa 50 dokumen abstrak *paper* akademisi dari masing-masing perguruan tinggi sebagai pengujian kriteria penilaian yang sifat kualitatif, kemudian menggunakan data online dari database akademisi *Science and Technology Index* (SINTA) sebagai pengujian kriteria penilaian yang bersifat kuantitatif yang mempunyai nilai berupa numeric, serta menggunakan data hasil analisa Backlink, Domain Authority dan Page Authority dari kriteria berbasis analisa *webhost*.

Uji coba dan evaluasi dilakukan untuk mengevaluasi jalannya perangkat lunak, mengevaluasi fitur utama, mengevaluasi fitur tambahan, mencari kesalahan antara hasil peringkat yang dihasilkan oleh sistem perangkat lunak dengan peringkat *Ground Truth* yang sudah ditetapkan, menganalisa faktor-faktor yang mempengaruhi kebenaran sistem, dan mengadakan perbaikan jika ada kekurangan.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi sistem, rancangan antarmuka, proses lain yang telah dilakukan, hasil-hasil dan analisa yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan informasi dan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas teori sebagai penunjang yang berhubungan dengan pokok pembahasan yang menjadi dasar dari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan alur, proses dan perancangan antarmuka pada perangkat lunak.

Bab IV Implementasi

Bab ini membahas mengenai implementasi dari perancangan perangkat lunak dan implementasi fitur-fitur penunjang perangkat lunak.

Bab V Uji Coba dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak, serta melakukan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan, serta evaluasi terhadap fitur-fitur perangkat lunak.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan perangkat lunak lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan dalam proses membangun dan mengembangkan perangkat lunak dalam Tugas Akhir.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan Tugas Akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan.

2.1 *University Rankings*

Peringkat perguruan tinggi merupakan salah satu cara untuk mengukur kualitas pendidikan di sebuah negara yang diterapkan oleh seluruh perguruan tinggi di dunia. Namun saat ini tujuan pemeringkatan perguruan tinggi cenderung berbeda dari awalnya yaitu menjadi salah satu cara untuk melihat kepopuleran perguruan tinggi tersebut di mata dunia internasional. Dimana perguruan tinggi yang mempunyai posisi peringkat tinggi akan menjadi perhatian bahwa pendidikan tinggi di negara tersebut berkualitas [1]- [2].

Hingga saat ini, ada berbagai macam publikasi pemeringkatan perguruan tinggi yang populer untuk memberikan hasil pemeringkatan perguruan tinggi di seluruh dunia [7]- [8], diantaranya:

1. QS World University Rankings.
2. Timer Higher Education World University Rankings (THE).
3. Academics Ranking of World University (ARWU).
4. Global Universities Rangkaing by US News.
5. Reuters World's Top 100 Innovative University.
6. SCImago Institution Rankings.
7. Webometrics.
8. 4ICU (4 International Colleges and Universities).

2.2 *QS World University Rankings*

Quacquarelli Symonds atau lebih sering disebut dengan *QS World University Rangkaing* merupakan publikasi peringkat perguruan tinggi seluruh dunia yang berlokasi di Inggris [4]. *QS World University Rangkaing* diterbitkan setiap tahun dengan

menghasilkan perubahan peringkat yang sangat sensitif, sehingga persaingan antar perguruan tinggi begitu kuat [1].

Awal mula terbentuknya *QS World University Rankings* pada tahun 2004 merupakan hasil kolaborasi dengan THE (*Times Higher Education*) dengan nama publik THE-QS, namun sejak tahun 2009 QS dan THE sepakat untuk berjalan sendiri-sendiri. QS berjalan hingga sekarang menggunakan metode yang diterapkan sejak awal dan lebih populer disebut publikasi *QS World University Ranking*. Sedangkan THE membangun publikasi peringkat perguruan tinggi sendiri dengan nama THES yang menggunakan metode lain yang hampir serupa dengan *QS World University Ranking* [3].

QS World University Ranking mempunyai 5 cakupan region sebagai observasi pemeringkatan, diantaranya Asia, Amerika Latin, negara Eropa berkembang - Asia Tengah, region Arab dan BRICS (Brasil, Rusia, India, China, South Africa). Untuk menghasilkan peringkat perguruan tinggi, *QS World University Ranking* mempunyai 6 kriteria utama [4], diantaranya:

1. Reputasi Akademik (40%)

Kriteria ini merupakan hasil survey responden akademisi perguruan tinggi di seluruh dunia. Survey ini diharapkan akan mendapatkan hasil yang objektif dari semua responden yang tersebar diseluruh dunia. Mereka mendapatkan bagian survey sesuai dengan latar belakang akademisi tersebut. Terdapat 5 garis besar bidang latar belakang akademisi sebagai responden, yaitu bidang ilmu seni dan sastra, ilmu teknik dan teknologi, ilmu biologi dan kesehatan, ilmu alam, dan ilmu sosial [3].

2. Reputasi Employer (10%)

Kriteria ini merupakan hasil survey responden perusahaan yang menaungi alumni perguruan tinggi di seluruh dunia [3]. Sama dengan kriteria reputasi akademik, kriteria ini ditujukan agar memperoleh hasil yang objektif terhadap kualitas lulusan (fresh graduate) perguruan tinggi. Salah

satu faktor penting agar perguruan tinggi dapat diperhatikan dalam kriteria ini adalah dengan menjalin banyak kerjasama dengan perusahaan sebagai rekan kerja maupun penelitian [2].

3. Jumlah Mahasiswa Aktif (20%)

Kriteria ini merupakan hasil jumlah mahasiswa yang sedang menempuh pendidikan di perguruan tinggi. Tujuan dari kriteria ini adalah untuk menganalisa pengawasan jumlah infrastruktur dengan jumlah mahasiswa, agar jumlah infrastruktur sebanding dengan jumlah mahasiswa. Perguruan tinggi yang baik merupakan perguruan tinggi yang memberikan fasilitas dan kenyamanan belajar [3].

4. Jumlah Sitasi Akademisi (20%)

Kriteria ini merupakan hasil jumlah sitasi maupun publikasi dari akademisi perguruan tinggi yang ter-*index* oleh publikasi Scopus. Sitasi/publikasi yang dihitung berusia maksimal terbit 5 tahun sebelumnya. Tujuan dari kriteria ini adalah untuk melihat seberapa banyak dampak penelitian akademisi perguruan tinggi terhadap lingkungan [3]. Karena tujuan perguruan tinggi adalah yang dapat memberikan dampak baru bagi masyarakat.

5. Jumlah Tenaga Pengajar Internasional (5%)

Kriteria ini merupakan hasil dari jumlah tenaga pengajar (staff/dosen) asing di sebuah perguruan tinggi. Lama waktu minimal tenaga pengajar asing menetap selama 3 bulan per tahun [2]. Tujuan dari kriteria ini adalah melihat kerjasama dan ketertarikan tenaga asing untuk perguruan tinggi tersebut.

6. Jumlah Mahasiswa Asing (5%)

Kriteria ini merupakan hasil dari jumlah mahasiswa asing yang belajar di perguruan tinggi, baik dalam rangka pertukaran pelajar, maupun program pendidikan lainnya

[3]. Tujuan dari kriteria ini adalah untuk melihat ketertarikan mahasiswa asing terhadap metode pembelajaran perguruan tinggi [2].

Dalam penilaian kriteria *QS World University Ranking* terdapat 2 jenis kriteria, yaitu kriteria yang bersifat kualitatif dan kuantitatif. Untuk kriteria yang bersifat kualitatif merupakan kriteria yang membutuhkan pengolahan data untuk mendapatkan nilai serta cenderung bersifat deskriptif dan membutuhkan analisa terlebih dahulu. Kriteria yang termasuk dalam jenis kualitatif adalah kriteria reputasi akademik dan reputasi employer.

Sedangkan untuk kriteria yang bersifat kuantitatif merupakan kriteria yang dapat ditentukan nilainya secara langsung tanpa melakukan analisa terlebih dahulu dan tidak bersifat deskriptif. Kriteria yang termasuk dalam jenis kuantitatif adalah kriteria jumlah mahasiswa aktif, jumlah sitasi/publikasi akademisi, jumlah staff internasional dan jumlah mahasiswa asing.

Untuk kriteria pertama yang bersifat kualitatif, yaitu reputasi akademik dalam *QS World University Rankings* ada beberapa poin yang menjadi acuan untuk digali dari responden survey untuk menghasilkan penilaian yang objektif [9], diantaranya:

1. Biodata personal responden akademisi

Poin penilaian ini, responden diminta menjelaskan mengenai biodata personal responden yang berkaitan dengan dimana akademisi bekerja, diantaranya seperti nama lengkap, alamat email institusional, jobdesk pekerjaan, lama bekerja dalam institusi, nama institusi dimana akademisi bekerja, pendapat mengenai negara yang paling berpengaruh dalam pendidikan tinggi dan penelitian, serta menanyakan mengenai pengetahuan akademisi terhadap Massive Open Online Courses (MOOCs) [9].

2. Knowledge specification

Poin penilaian ini, responden diminta menjelaskan berdasar pengetahuan responden mengenai pengetahuan dasar tentang region dan bidang ilmu yang akan dibahas/ditanyakan, diantaranya seperti pengetahuan terhadap kualitas akademik berdasarkan region (amerika, asia-australia-new zealand, eropa-timur tengah-afrika atau semua region), pengetahuan terhadap kategori bidang ilmu pengajaran, pengetahuan terhadap sub bidang ilmu yang lebih spesifik, serta mengenai pembagian waktu akademisi untuk melakukan riset (*research*), pengajaran (*teaching*), administrasi (*administration*) dan lainnya dalam bentuk persen [9].

3. Penelitian terbaik

Poin penilaian ini, responden diminta memilih beberapa perguruan tinggi yang dianggap mempunyai tingkat penelitian (*research*) yang baik, diantaranya seperti memilih 10 institusi dalam negeri sesuai dengan region negara responden yang mempunyai kualitas penelitian terbaik, memilih 30 institusi luar negeri yang mempunyai kualitas penelitian terbaik, serta dalam memilih institusi dalam negeri maupun luar negeri tersebut harus mencakup kategori bidang ilmu yang diteliti. Ada 5 bidang ilmu, maka total responden harus memilih 50 perguruan tinggi dalam negeri terbaik yang mencakup 5 bidang ilmu, serta 150 perguruan luar negeri yang mencakup 5 bidang ilmu.

4. Inovasi dalam penelitian

Poin penilaian ini, responden diminta mengidentifikasi perguruan tinggi mana yang mempunyai tingkat inovasi yang baik dalam penelitian, diantaranya seperti mengidentifikasi institusi yang paling cepat dan dinamis dalam perkembangan dan inovasi penelitian, serta bidang ilmu apa yang dimiliki institusi tersebut yang paling cepat berkembang dalam hal inovasi [9].

5. Leading company

Poin penilaian ini, responden diminta mengidentifikasi perusahaan mana yang paling berkontribusi terhadap penelitian, diantaranya seperti mengidentifikasi 5 perusahaan yang menurut responden akademisi paling berpengaruh terhadap pendidikan sesuai dengan bidang pendidikan responden, serta menjelaskan alasan memilih 5 perusahaan tersebut [9].

6. MOOCs

Poin penilaian ini, responden diminta menjelaskan mengenai penerapan MOOCs dalam institusi responden [10], diantaranya seperti apakah institusi pendidikan responden sudah menerapkan pembelajaran berbasis MOOCs, apakah institusi pendidikan responden dapat merencanakan pembelajaran berbasis MOOCs untuk 3 tahun kedepan, apakah responden telah menerapkan pembelajaran MOOCs berdasarkan inisiatif pribadi, serta pendapat responden mengenai keuntungan pembelajaran berbasis MOOCs [9].

7. Profil penelitian responden akademisi

Poin penilaian ini, responden diminta menjelaskan profil publikasi/penelitian responden selama ini jika ada, diantaranya seperti artikel jurnal, konferensi, paper/buku, kemampuan (*performance*) responden, pengalaman (*exhibitions*) responden, serta paten yang dimiliki [9].

Untuk kriteria kedua yang bersifat kualitatif yaitu reputasi employer, ada beberapa poin yang menjadi acuan untuk digali dari responden perusahaan untuk menghasilkan penilaian yang objektif [11], diantaranya:

1. Biodata personal responden perusahaan

Poin penilaian ini, responden perusahaan diminta menjelaskan mengenai biodata personal yang berkaitan dengan institusi dimana responden bekerja, diantaranya

nama lengkap, nama perusahaan, alamat email resmi berdomain perusahaan, website perusahaan, jabatan pekerjaan, negara lokasi perusahaan berjalan, sektor utama perusahaan, ukuran perusahaan (kuantitas staff), kategori lulusan yang diterima sebagai staff (internship, sarjana-magister (S1/S2), MBA), dan yang terakhir lokasi negara yang paling banyak kandidat diterima sebagai staff perusahaan [11].

2. Analisa perguruan tinggi bisnis terbaik

Poin penilaian ini, responden perusahaan diminta menjelaskan mengenai pendapat mereka terhadap perguruan tinggi yang mempunyai lulusan terbaik, baik dalam hal bisnis maupun keseluruhan, diantaranya adalah memilih 10 perguruan tinggi domestik sesuai tempat tinggal responden yang mempunyai lulusan terbaik, memilih 30 perguruan tinggi luar negeri yang mempunyai lulusan terbaik, memberi tahu daftar perguruan tinggi dari staff perusahaan yang bekerja dalam perusahaan tersebut, serta bidang disiplin ilmu yang paling banyak direkrut oleh perusahaan tersebut [11].

Selain itu mengidentifikasi juga sekolah bisnis dimana responden perusahaan diminta untuk menjelaskan beberapa hal, diantaranya adalah mengidentifikasi 10 sekolah bisnis yang alumninya berpotensi direkrut oleh perusahaan, perekrutan staff bergelar MBA (Master of Business Administration), membiayai staff pada program executive MBA untuk mengembangkan potensi, menjelaskan pentingnya spesialisasi MBA dalam perusahaan, serta memilih 5 perguruan tinggi yang unggul dalam gelar MBA (ada 10 kategori spesialisasi, yang berarti ada 50 perguruan tinggi yang dipilih).

3. Kualifikasi untuk staff bergelar MBA

Poin penilaian ini, responden perusahaan diminta menjelaskan pendapatnya mengenai kualifikasi staff yang

bergelar MBA, diantaranya adalah memilih region negara yang mempunyai sekolah bisnis terbaik dan menciptakan dampak tinggi untuk bisnis, lalu menjelaskan mengenai ketentuan yang ideal untuk staff bergelar MBA untuk bekerja (lama waktu kerja per hari, gaji, bonus, tunjangan), divisi/posisi kerja yang sesuai dengan staff bergelar MBA, rata-rata gaji yang ditawarkan oleh perusahaan untuk freshgraduate.

4. Pembahasan umum

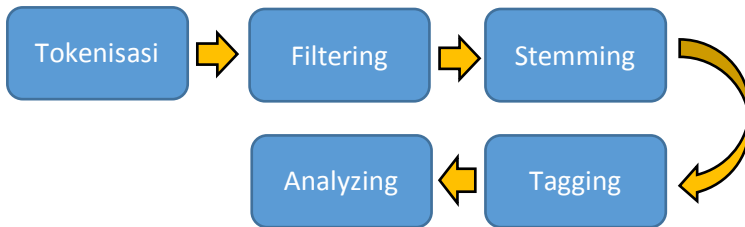
Poin penilaian ini, responden perusahaan diminta menjelaskan pendapatnya mengenai kebutuhan skill apa yang harus dimiliki oleh calon staff (bergelar MBA maupun reguler), diantaranya adalah skill yang dibutuhkan untuk freshgraduate, seperti pengalaman internasional, pengetahuan tentang subjek, kepemimpinan, pengalaman magang, kemampuan individu, dan kemampuan bahasa. Selain itu diminta menjelaskan mengenai skill yang dibutuhkan oleh calon staff MBA, seperti kemampuan individu, interaksi dengan lingkungan, kepemimpinan, prestasi akademik, pemikiran strategis, kemampuan bahasa, kewiraswastaan, keterampilan IT, manajemen resiko, pengalaman, pemasaran, keterampilan keuangan, komunikasi, tanggung jawab sosial, dan e-business [11].

Untuk kriteria ketiga hingga keenam, poin yang dijadikan penilaian adalah jumlah dari masing-masing. Kriteria ini tidak mempunyai ketentuan poin khusus seperti kriteria sebelumnya yang bersifat kualitatif.

2.3 Teks Mining

Teks mining merupakan proses menggali data teks yang didapatkan dari sumber data berupa dokumen (word, pdf, kutipan, dan sejenisnya). Teks mining disebut juga dengan teknik mengekstraksi pola dari sebuah dokumen. Tujuan utama dari teks mining adalah untuk mendapatkan informasi yang berguna dari

sekumpulan dokumen dengan melakukan pencarian kata-kata yang dapat mewakili isi dari sebuah dokumen yang kemudian dapat dianalisa. Kegunaan dari penggunaan teks mining adalah sebagai pengkategorian teks (*text categorization*) dan pengelompokan teks (*text clustering*) [12]. Proses umum dalam teks mining seperti pada Gambar 2.3-1.



Gambar 2.3-1 Proses Utama Teks Mining

2.4 Teknik Grabbing

Merupakan salah satu cara untuk mengambil konten teks atau object yang ditampilkan dalam situs website yang kemudian akan diletakkan dalam website milik pribadi. Secara umum teknik ini akan menduplikasi konten website tujuan untuk ditampilkan dalam website kita, akan tetapi dari sini banyak informasi dan kegunaan lain yang dapat dikembangkan. Salah satunya adalah sebagai salah satu bahan teks mining dimana dengan tujuan untuk menganalisa konten teks dalam website tersebut. Teknik grabbing lebih banyak dikembangkan menggunakan bahasa pemrograman PHP, terdapat library yang mendukung analisa ini. Teknik grabbing dapat dikembangkan sendiri oleh developer sesuai dengan kebutuhan yaitu kegunaan dan konten situs website sasaran, oleh karena itu teknik ini bisa dikatakan tidak fleksibel karena jika ingin melakukan grabbing pada situs website tertentu maka harus merubah pola konten yang akan diambil. Beberapa cara menggunakan teknik grabbing diantaranya adalah fungsi cURL, fungsi `file_get_contents`, library simple HTML dom parser, dan pemanfaatan ReGex (*Regular Expression*).

2.4.1 Fungsi cURL

Merupakan sebuah fungsi yang berguna untuk mengambil konten dari sebuah situs website. cURL dapat digunakan untuk melakukan simulasi request method POST ke server dan juga menggunakan Cookie. Fungsi ini dijalankan pada Bahasa pemrograman PHP. Keuntungan menggunakan cURL adalah beberapa situs website yang melindungi kontennya dengan menggunakan cookie dan post method, dengan menggunakan cURL dapat mengakses konten yang terproteksi tersebut. Beberapa fungsi yang diakomodasi oleh cURL:

1. `curl_init()` untuk inialisasi curl library.
2. `curl_setopt()` untuk mengatur pilihan dari curl.
3. `curl_exec()` untuk mengeksekusi/menjalankan query.
4. `curl_close()` untuk menghentikan curl system.

```
$options = array(
    CURLOPT_CUSTOMREQUEST =>"GET",
    CURLOPT_POST           =>false,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_CONNECTTIMEOUT => 120,
    CURLOPT_TIMEOUT       => 120
);

$ch = curl_init( $url );
curl_setopt_array( $ch, $options );
$content = curl_exec( $ch );
curl_close( $ch );
$header['content'] = $content;
```

Kode Sumber 2.4-1 Syntax cURL dengan setting option url

2.4.2 Fungsi File_get_contents

Merupakan cara untuk membaca isi file ke dalam string. Karena akan menggunakan teknik pemetaan memori, jika hal ini

didukung oleh server, untuk meningkatkan kinerja. Isi file dalam konteks grabbing adalah, bagian konten situs website yang akan diterjemahkan menjadi string, kemudian akan dianalisa sesuai kebutuhan mengambil informasi dari tag konten bagian tertentu menggunakan ReGex (*Regular Expression*).

```
file_get_contents(path,include_path,context,start,max_length)
```

Kode Sumber 2.4-2 Syntax Fungsi file_get_contents

2.4.3 Simple HTML Dom Parser

Merupakan sebuah syntax dalam bahasa pemrograman PHP yang berfungsi untuk menguraikan elemen-elemen HTML untuk diambil dan dirubah atau dihapus pada tag/elemen-elemen tertentu menjadi konten baru yang diinginkan. Dalam PHP sendiri sudah disediakan library HTML DOM yang dapat diunduh terpisah dengan banyak keuntungan.

```
<?php
    require_once 'simple_html_dom.php';

    $html = file_get_html("http://hieppies.blogspot.com");
    $html = file_get_html("index.html");
    $html = file_get_html("/menu/index.html");

    $html = str_get_html("<html>
        <body>
            <p>Hi, Cantik!.</p>
            <p>Lagi ngapain?.</p>
        </body>
    </html>");
?>
```

Kode Sumber 2.4-3 Syntax Penggunaan Library Simple HTML Dom Parser

2.4.4 Regular Expression (ReGex)

Merupakan teknik yang digunakan untuk pencocokan string teks, seperti karakter tertentu, kata-kata, atau pola karakter. ReGex memiliki 2 fungsi utama yaitu untuk mencari dan mengganti, mencari suatu pola tertentu dalam text lalu menggantinya menjadi pola yang lain. Dalam penggunaannya untuk grabbing situs website, ReGex biasanya dikombinasikan dengan fungsi `file_get_contents` dari PHP. Mengambil konten menggunakan fungsi `file_get_contents` sedangkan menganalisa hasil dan mencari string konten yang dibutuhkan menggunakan regular expression.

2.5 Pusat Data Scopus

Scopus merupakan database bibliografi yang meng-*index* abstrak dan sitasi/kutipan untuk artikel jurnal akademik. Scopus merupakan salah satu produk bibliograsi dari Elsevier. Scopus muncul sejak tahun 1995, kurang lebih 21 tahun Scopus meng-*index* jurnal dari para akademisi di seluruh dunia. Hingga saat ini kurang lebih terdapat 22.000 judul artikel jurnal dan lebih dari 5.000 penerbit [12].

Scopus meng-*index* artikel jurnal akademik dari berbagai bidang ilmu. Maka dari itu banyak publikasi pemeringkatan perguruan tinggi lain seperti Scimago menggunakan jasa dari Scopus sebagai pusat database jurnal akademik. Scopus sendiri sangat berhati-hati dan selektif untuk meng-*index* artikel jurnal akademik. Tidak semua artikel jurnal di database lain seperti Google Scholar, ataupun Web of Science dapat ter-*index* oleh Scopus. Scopus hanya meng-*index* berdasarkan kualitas artikel jurnal tersebut.

Scopus juga menyediakan layanan untuk developer berupa Scopus API (*Application Program Interfaces*) sebagai sarana untuk mengembangkan informasi berdasarkan database Scopus, hanya saja untuk menggunakan layanan ini hanya untuk institusi yang berlangganan (*Subscribe*) saja [13]. Tidak semua orang dan institusi dapat menggunakan layanan ini. Keuntungan

dari database Scopus ini adalah mempunyai atribut data dan informasi yang lengkap, diantaranya nama author, jumlah publikasi, jumlah sitasi, ORCID, dan grafik perkembangan keaktifan author [13]. Selain itu dukungan layanan API sangat lengkap dan juga kontennya dapat dipercaya.

2.6 *Google Scholar*

Google Scholar merupakan layanan database metadata publikasi yang dibuat oleh Google. Dalam Google Scholar memungkinkan pengguna melakukan pencarian materi berupa teks dalam berbagai format publikasi. Index Google Scholar mencakup jurnal online dan publikasi ilmiah sehingga semua jurnal online bisa tercover dalam Google Scholar [14]. Tujuan utama dari Google Scholar adalah menyusun paper seperti yang dilakukan peneliti, dengan memperhatikan kelengkapan teks, nama penulis yang menampilkan artikel, serta frekuensi penggunaan kutipan artikel dalam literatur akademis lainnya.

Dalam Google Scholar, seorang peneliti dapat mendaftarkan akun dirinya dalam sistem sebagai biodata untuk setiap paper yang dipublikasikan. Dalam setiap biodata penulis terdapat beberapa indikator yang cukup unik sebagai bahan acuan bahwa paper yang dipublikasikan oleh penulis tersebut berkualitas dan terpercaya atau tidak [15]. Hal ini dapat dilihat dari jumlah paper yang dipublikasikan penulis yang terindex oleh Google Scholar, jumlah sitasi (*citation*), nilai h-index, dan nilai i10-index [16]. Berikut penjelasan masing-masing indikator dalam Google Scholar:

1. Jumlah Publikasi

Merupakan jumlah paper yang dipublikasikan oleh penulis yang terindex oleh Google Scholar. Disini dapat diketahui juga tahun publikasi dan jumlah sitasi/kutipan terhadap paper tersebut. Semakin banyak orang lain yang mengutip paper tersebut akan semakin tinggi peringkat posisi paper dalam daftar publikasi penulis tersebut. Jumlah publikasi ini dihitung sejak penulis aktif dalam Google Scholar.

2. Jumlah Sitasi

Merupakan jumlah total kutipan dari orang lain yang mengutip paper penulis tersebut. Semakin banyak orang yang mengutip paper penulis maka akan semakin banyak pula total sitasinya. Terdapat dua versi dalam jumlah sitasi pada Google Scholar, terdapat jumlah total (*all citation*) yang merupakan jumlah keseluruhan sejak penulis aktif dalam Google scholar, serta jumlah sejak 5 tahun terakhir (*Since 2012*) yang dapat dilihat bahwa kualitas sitasi dari paper tersebut baik atau tidak sejak 5 tahun terakhir.

3. Nilai H-Index

Merupakan ukuran index yang diusulkan oleh seorang fisikawan di Universitas California bernama Jorge E. Hirsch pada tahun 1985. H-Index ini masih terus diteliti keakuratannya. H-Index sendiri adalah nilai index yang didapatkan dari jumlah paper sebanyak h paper penulis dengan jumlah kutipan untuk setiap *paper* minimal sama dengan nilai h tersebut [16].

4. Nilai i10-Index

Merupakan nilai index yang didapatkan dari jumlah 2 *paper* dari penulis yang dikutip minimal oleh 10 *paper* dari orang lain. Sehingga kondisi minimal dari kualitas *paper* tersebut dapat diketahui oleh cara ini.

2.7 *Science and Technology Index (SINTA)*

Science and Tecnology Index atau biasa disebut dengan SINTA merupakan pusat index, sitasi dan kepakaran terbesar di Indonesia. Sistem informasi riset berbasis website ini menawarkan akses cepat, mudah dan komprehensif untuk mengukur unjuk kerja peneliti, institusi dan jurnal di Indonesia. Sinta menyediakan benchmark dan analisis, identifikasi kekuatan riset tiap institusi, memperlihatkan kolaborasi penelitianm menganalisisi tren penelitian dan direktori pakar. SINTA memberikan informasi mengenai tolok ukur institusi, kolaborasi, analisis penelitian serta direktori akademisi di Indonesia [17].

Beberapa informasi yang dapat digali dari database online SINTA [18], antara lain:

1. Jurnal yang diterbitkan oleh institusi pendidikan dan penelitian tinggi yang disajikan dalam bentuk peringat.
2. Institusi pendidikan tinggi yang mempunyai kontribusi dalam penelitian, berisikan informasi jumlah author akademisi serta total sitasi yang dihasilkan dari masing-masing institusi pendidikan dan penelitian tinggi yang disajikan dalam bentuk peringat.
3. Author akademisi yang mempunyai kontribusi dalam penelitian, berisikan informasi jumlah sitasi yang dimiliki, skor h-index, dan i10-index.
4. Detail informasi setiap institusi pendidikan tinggi:
 - a. Jumlah sitasi Google Scholar.
 - b. Jumlah dokumen Google Scholar.
 - c. Jumlah artikel pada Scopus database.
 - d. Jumlah non artikel pada Scopus database.
 - e. Jumlah sitasi dari Scopus.
 - f. Jumlah akademisi.
 - g. Daftar akademisi beserta kualitas sitasi (h-index, i10-index) dari Google Scholar dan Scopus database.
 - h. Jumlah jurnal artikel, book chapters, dan conference paper dari akademisi maupun institusi.
 - i. Skor penilaian menurut SINTA.

2.8 Pangkalan Data Pendidikan Tinggi (PDDIKTI)

Pangkalan Data Pendidikan Tinggi atau lebih umum disebut PDDIKTI merupakan pusat kumpulan data penyelenggara pendidikan tinggi seluruh Indonesia. Kementerian Riset, Teknologi, dan Pendidikan Tinggi merupakan kementerian dalam pemerintahan Indonesia yang berfokus pada penyelenggaraan urusan bidang riset, teknologi dan pendidikan tinggi [19].

Data dalam PDDIKTI merupakan hasil sinkronisasi yang dikelola oleh masing-masing perguruan tinggi di Indonesia, sehingga akan lebih update dalam penyebaran informasi, serta

terintegrasi secara nasional. Pangkalan data ini menjadi salah satu instrumen pelaksanaan penjaminan mutu. Data yang terangkum dalam PDDIKTI harus mudah diakses oleh *stakeholder* (lembaga akreditasi, pemerintah, masyarakat), sehingga menerapkan prinsip transparansi perguruan tinggi.

Beberapa data yang disediakan oleh PDDIKTI dalam sistem yang dapat dilihat [19], diantaranya:

1. Jumlah perguruan tinggi di Indonesia.
2. Jumlah program studi setiap perguruan tinggi.
3. Jumlah mahasiswa (jenis kelamin, kelompok bidang).
4. Jumlah dosen (jenis kelamin, ikatan kerja/status kepegawaian, pendidikan terakhir, jabatan akademik).

2.9 *Alexa Traffic Rank*

Alexa Traffic Rank merupakan sistem yang menyediakan fasilitas informasi tentang peringkat *webhost*, yang dilihat dari jumlah *traffic* pengunjung yang masuk kedalam situs *webhost* tersebut. Dengan kata lain Alexa Traffic Rank merupakan salah satu *tools* untuk *Search Engine Optimization* (SEO) website yaitu sitem untuk menganalisa dan mempengaruhi tingkat keterlihatan (*visibilitas*) sebuah situs pada mesin pencari [20].

Alexa Traffic Rank mempunyai nilai peringkat dari nilai terkecil yaitu nilai angka 1 (satu) hingga seterusnya (berjuta-juta) yang mana dalam perhitungan sistem di Alexa Traffic Rank apabila semakin kecil nilai Alexa Traffic Rank, nilai kepopuleran situs akan semakin bagus. Selain itu Alexa Traffic Rank juga menyediakan API untuk pengembangan (*developer*) perangkat lunak berbasis analisa SEO yang tidak berbayar (*free*).

2.10 *MOZ Analytics*

MOZ Analytics merupakan salah satu *tools* yang digunakan untuk mengetahui jumlah pencarian *webhost*. MOZ Analytics dikembangkan oleh SEOMoz yang berbasis di Seattle, Washington, Amerika Serikat. Salah satu product dari MOZ yang berfokus pada pemeringkatan dan analisa adalah MOZ Rank.

MOZ Rank mengkuantifikasi kepopuleran link *webhost*. Halaman *webhost* mendapatkan skor MOZ Rank berdasarkan halaman lain yang terhubung dengan halaman *webhost* tersebut (ditautkan). Semakin banyak tautan dari halaman *webhost*, maka akan semakin tinggi skor dari MOZ Rank. Dengan kata lain MOZ Analytics dengan berbagai macam produk turunannya merupakan sebuah tools untuk *Search Engine Optimization* (SEO). MOZ Analytics mengembangkan API untuk *developer* SEO yang tidak berbayar (*free*).

2.11 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python sendiri mulai diluncurkan di masyarakat sejak tahun 1991 oleh Guido van Rossum dengan nama vendor Python Software Foundation. Python diklaim sebagai bahasa pemrograman yang menggabungkan kapabilitas, kemampuan dengan sintaksis kode yang jelas dan dilengkapi dengan pustaka (*library*) yang lengkap dan komprehensif.

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi yang diluncurkan. Namun pada prinsipnya python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan dengan definisi Open Source dan General Public License [21]. Python terdiri dari dua buah versi terbaru yaitu 3.4.2, dan 2.7.8 yang bebas digunakan oleh developer. Versi Python ini rilis pada tahun 2014. Kelebihan yang dimiliki oleh bahasa pemrograman Python sebagai berikut:

1. Memiliki kepustakaan (*library*) yang luas, dalam Python sudah disediakan modul-modul siap pakai sesuai kebutuhan developer.
2. Memiliki tata bahasa yang mudah dipelajari.
3. Memiliki aturan layout (*indentation*) yang mudah untuk dilakukan pengecekan.
4. Berfokus pada orientasi objek.
5. Memiliki sistem pengolahan memori otomatis.

6. Modular, mudah dikembangkan dengan menciptakan modul-modul baru sesuai kebutuhan.

2.12 Library Python

Beberapa pustaka (*library*) yang digunakan dalam membangun sistem Tugas Akhir yang berhubungan dengan *teks mining*, diantaranya:

1. Natural Language ToolKit (NLTK)

NLTK merupakan pustaka utama untuk proses pengolahan bahasa alami *natural language process* (NLP). Pustaka ini berperan penting dalam proses teks mining seperti klasifikasi, tagging, tokenisasi, dan pemrosesan semantic. Pustaka ini diimplementasikan pada bahasa pemrograman Python, selain itu pustaka ini juga didukung oleh dokumentasi yang lengkap, mudah untuk dipahami, dan forum diskusi yang banyak membantu [22]. Beberapa pustaka NLTK yang diterapkan dalam Tugas Akhir ini, antara lain:

- a. NLTK corpus stopwords

Merupakan pustaka yang menyediakan corpus berisi keyword kata-kata stopwords, kata umum yang kurang penting keberadaannya didalam konteks dokumen. Corpus stopwords ini dikembangkan dalam berbagai bahasa yang disediakan oleh NLTK.

- b. NLTK Tokenize

Merupakan pustaka untuk melakukan preprocessing dokumen dengan melakukan tokenize dokumen menjadi sebuah kata tunggal.

- c. Porter Stemmer

Merupakan pustaka untuk proses stemming dalam sebuah dokumen, dengan menghapus imbuhan kata. Stemmer ini mengadopsi berbagai tipe bahasa utam yang ada didunia.

- d. Snowball Stemmer

Merupakan pustaka lain untuk proses stemming. Pustaka stemming yang dimiliki NLTK bervariasi

dengan tujuan pengembangan untuk mencapai hasil yang lebih bagus. Dari hasil penelitian sebelumnya, proses stemming dalam NLP menjadi isu utama yang terus dipecahkan. Snowball Stemmer ini merupakan salah satu pengembangan dari Porter Stemmer yang mempunyai keuntungan lebih.

e. WordNet

WordNet merupakan database leksikan untuk bahasa Inggris. Fungsi utama pustaka ini adalah untuk mengelompokkan sebuah kata kepada kata lain yang terhubung dalam rule sinonim atau biasa disebut synsets [24]. Penggunaan utama dari WordNet sebagai kombinasi antara kamus dan Thesaurus.

f. Postag Word Tokenize

Merupakan pustaka untuk pemrosesan tagging *Part of Speech* (POS Tagger) kata dalam sebuah kalimat dalam dokumen. Pustaka ini berperan penting dalam penerapan perangkat lunak pada Tugas Akhir ini.

2. Sckit Learn

Merupakan pustaka dalam bahasa Python yang berfokus pada proses pembelajaran numeric, seperti algoritma klasifikasi, clustering, dan regresi. Pustaka ini dilengkapi dengan berbagai macam algoritma dan didesain berjalan dalam sistem vector Numpy dan SciPy [24].

Beberapa algoritma dari pustaka Sckit Learn yang digunakan dalam Tugas Akhir ini, antara lain:

a. CountVectorizer

Pustaka ini ada pada modul `sklearn.feature_extraction` yang digunakan untuk mengekstrak fitur dalam format yang didukung oleh algoritma pembelajaran dari Sckit Learn dari kumpulan data berformat teks maupun gambar. Fungsi utama CountVectorizer adalah mengkonversi kumpulan dokumen teks kedalam matriks token (*count term*).

- b. `TfidfTransformer`

Pustaka ini berfungsi sebagai transformasi matriks perhitungan menjadi representasi *Term Frequency-Invers Document Frequency* (TF-IDF) yang dinormalisasi. Merupakan skema pembobotan untuk mengurangi term yang sering muncul di corpus tertentu.
3. `Numpy`

Merupakan modul yang menyediakan objek perhitungan mathematic yang memudahkan dalam melakukan perhitungan. Objek utama yang disediakan adalah array yang dapat berperan sebagai matriks.
4. `TextBlob`

`Textblob` merupakan pustakan yang dapat melakukan berbagai proses terhadap data teks, seperti tokenisasi, analisa sentiment hingga proses translate bahasa kedalam beberapa bahasa yang umum didunia. Penggunaan `textblob` dalam Tugas Akhir ini sebagai pustaka untuk melakukan uji coba proses translate dokumen berbahasa Indonesia menjadi bahasa Inggris sebelum dilakukan preprocessing [25].
5. `Option Parser`

Merupakan pustaka yang fleksibel untuk mengurai opsi dalam baris perintah pada command line terminal python. Keuntungan dari `Option Parser` adalah menggunakan gaya deklaratif bahasa yang lebih baik daripada modul yang sama sebelumnya. `Option Parser` atau `Optparse` memungkinkan user untuk menentukan opsi dalam sintaks GNU/POSIX dengan didukung oleh pilihan bantuan (*help*).
6. `MySQL Python dan MyQSL Client`

Merupakan modul dari `MySQL` untuk konektor bahasa pemrograman python kepada basis data `MySQL`. `MySQL`

Python dan MySQL Client digunakan untuk menyimpan hasil akhir teks mining dalam database.

7. Pustaka import tambahan

Selain menggunakan pustaka utama diatas yang membutuhkan konfigurasi lebih lanjut, ada beberapa pustaka default dari Python, diantaranya:

a. String

Modul ini berisi beberapa konstanta dan kelas yang berfungsi sebagai metode pengolah string untuk tipe data string dalam pengolahan kata, seperti fungsi str, unicode, list, tuple, bytearray, buffer, xrange section.

b. Re (regular expression)

Modul ini menyediakan operasi pencocokan reguler expression untuk pola tipe data string. Modul ini sesuai untuk pengolahan kata yang tanpa menggunakan library tambahan.

c. Os (Operating System)

Modul ini menyediakan cara portabel untuk menggunakan fungsi yang bergantung pada sistem operasi. Menjadi jalan untuk proses programming untuk memodifikasi fungsi pada sistem operasi.

d. Math

Modul ini menyediakan akses untuk mengeksplere fungsi matematika yang didefinisikan oleh standar bahasa C. Modul ini tidak dapat digunakan dengan bilangan kompleks.

e. Random

Modul ini mengimplementasikan proses generator bilangan acak untuk berbagai macam distribusi. Generator bilangan acak dapat ditentukan sesuai kebutuhan pengguna.

f. Time

Modul ini menyediakan berbagai fungsi yang berhubungan dengan waktu. Meskipun modul ini selalu tersedia, tidak semua fungsi tersedia di semua platform.

Sebagian besar fungsi yang didefinisikan dalam modul ini memanggil fungsi library bahasa C.

g. Csv

Format CSV (Comma Separated Values) merupakan format impor dan ekspor paling umum untuk spreadsheet dan database. Penggunaannya yang sederhana dan kurangnya standar untuk norma penulisan menjadi perbedaan disetiap platform yang menggunakan CSV.

2.13 PHP dan HTML

Hypertext Preprocessor atau yang biasa disebut dengan PHP merupakan bahasa script yang dapat disisipkan dalam kumpulan kode HTML [26]. PHP banyak dipakai dalam pemrograman website selain dengan HTML. PHP sendiri ditemukan oleh Rasmus Lerdorf pada tahun 1995, dan hingga sekarang PHP dikembangkan oleh vendor utama yaitu The PHP Group. Hingga saat ini PHP telah masuk dalam versi ke 7, atau biasa yang disebut dengan PHP 7. Kelebihan dari bahasa pemrograman PHP sebagai berikut:

1. Bahasa pemrograman yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web server yang mendukung PHP dapat ditemukan dimana-mana, diantaranya apache, IIS, Lighttpd, dan Xitami.
3. Dalam sisi pengembangan (*developer*) lebih mudah karena banyak organisasi maupun developer lain yang siap membantu mengembangkan variasi dari PHP sendiri.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang baik dan banyak.
5. Open Source yang dapat digunakan di berbagai *Operating System* (OS) dan dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah sistem.

2.14 Database Management System (DBMS) MySQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis (DBMS) data SQL. MySQL disebut database yang multithread, multi user dengan jumlah instalasi 6 juta kali di seluruh dunia yang membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL). MySQL sendiri dikembangkan oleh Oracle pada tahun 1995. MySQL sendiri bisa dijalankan disemua tipe sistem operasi [27].

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional RDBMS yaitu sistem basis data yang entitas utamanya terdiri dari tabel-tabel yang mempunyai relasi dari satu tabel ke tabel yang lain. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu *Structured Query Language (SQL)*. SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.15 Part-Of-Speech Tagger (POS Tagger)

Merupakan teknik yang dapat menerapkan penanda tag dari dokumen teks (corpus) dalam beberapa bahasa untuk menjadi kelas kata berdasarkan tipe struktur kalimat dalam dokumen. Struktur kata ini seperti kata benda (*noun*), kata kerja (*verb*), kata sifat (*adjective*) dan sebagainya. POS Tagger disebut juga *Grammatical Tagging*. Proses menandai sebuah kata dalam sebuah teks (corpus) yang sesuai dengan bagian pembicaraan tertentu. Berdasarkan definisi konteks, yaitu hubungan dengan kata-kata yang berdekatan yang terkait dalam frase, kalimat atau paragraf [28].

POS Tagging pada zaman dahulu dilakukan menggunakan cara manual yaitu dengan menggunakan jasa ahli bahasa untuk memetakan kelas dari kata dalam corpus. Untuk saat ini POS Tagging dilakukan dalam konteks komputasi linguistik,

menggunakan algoritma yang mengasosiasikan istilah diskrit, serta bagian pembicaraan yang tersembunyi (*latent*). Salah satu pengembangan POS Tagger yang populer saat ini adalah Stanford POS Tagger yang dibangun dalam bahasa Java. Struktur kelas kata POS Tagging digambarkan pada Tabel 2.15-1 [28].

Tabel 2.15-1 Struktur Kelas kata POS Tagging

Tag	Deskripsi	Contoh Penerapan
CC	Coordin, conjunction	And, but, or
CD	Cardinal number	One, two
DT	Determiner	A, the
EX	Existential 'there'	There
FW	Foreign word	Mea culpa
IN	Preposition/sub-conjunction	Of, in, by
JJ	Adjective	Yellow
JJR	Adj., comparative	Bigger
JJS	Adj., superlative	Wildest
LS	List item marker	1, 2, one
MD	Modal	Can, should
NN	Noun, sing. Or mass	Llama
NNS	Noun, plural	Llamas
NNP	Proper noun, sing.	IBM
NNPS	Proper noun, plural	Carolinas
PDT	Predeterminer	All, both
POS	Possessive ending	's
PRP	Personal pronoun	I, you, he
PRP\$	Possessive pronoun	Your, one's
RB	Adverb	Quickly, never
RBR	Adverb, comparative	Faster
RBS	Adverb, superlative	Fastest
RP	Participle	Up, off
SYM	Symbol	_, %, &
TO	"to"	to
UH	Interjection	Ah, oops
VB	Verb base form	Eat

VBD	Verb past tense	Ate
VBG	Verb gerund	Eating
VBN	Verb past participle	Eaten
VBP	Verb on-3sg pres	Eat
VBZ	Verb 3-sg pres	Eats
WDT	Wh-determiner	Which, that
WP	Wh-pronoun	What, who
WP\$	Possessive wh-	Whose
WRB	Wh-adverd	How, where
\$	Dollar sign	\$
#	Pound sign	#
“	Left quote	‘ or “
”	Right quote	‘ or “
(Left parenthesis	[, (, {, <
)	Right parenthesis],), }, >
,	Comma	,
.	Sentence-final punc	. ! ?
:	Mid-sentence punc	: ; ... -- -

2.16 Bloom Taxonomy

Bloom Taxonomy merupakan sebuah cara untuk membagi cara belajar seseorang menjadi tiga ranah (*domain*) utama. Taksonomi ini dibuat untuk tujuan pendidikan. Dibuat pertama kali oleh Benjamin S. Bloom pada tahun 1956. Tiga ranah (*domain*) utama diantaranya [29]:

1. Cognitive Domain (ranah kognitif) berisi mengenai perilaku yang menekankan pada aspek intelektual, seperti pengetahuan, pengertian, dan keterampilan berpikir.
2. Affective Domain (ranah afektif) berisi mengenai perilaku yang menekankan pada aspek perasaan dan emosi, seperti minat, sikap, apresiasi, dan cara penyediaan diri.
3. Psychomotor Domain (ranah psychomotor) berisi mengenai aspek ketrampilan motorik seperti tulisan tangan, mengetik, dan mengoperasikan mesin.

Dari ketiga domain diatas, domain yang paling berpengaruh terhadap hasil intelektual adalah domain kognitif. Domain ini selanjutnya dibagi kedalam 6 level, diantaranya [29]:

1. Knowledge
Menunjukkan materi yang telah dipelajari sebelumnya dengan mengingat fakta, istilah, konsep dasar dan jawaban.
2. Comprehension
Menunjukkan pemahaman tentang fakta dan gagasan dengan mengatur, membandingkan, menerjemahkan, menafsirkan, memberikan deskripsi dan mengemukakan gagasan utama.
3. Application
Memecahkan masalah dengan menerapkan pengetahuan, fakta, teknik dan peraturan yang diperoleh dengan cara yang berbeda.
4. Analysis
Memeriksa dan memecahkan informasi menjadi beberapa bagian dengan mengidentifikasi motif atau sebab, serta membuat kesimpulan dan menemukan bukti untuk mendukung generalisasi.
5. Synthesis
mengkompilasi informasi bersama-sama dengan cara yang berbeda dengan menggabungkan elemen dalam pola baru atau mengusulkan solusi alternatif.
6. Evaluation
Menyajikan dan mempertahankan pendapat dengan membuat penilaian tentang informasi, validitas gagasan atau kualitas kerja berdasarkan seperangkat kriteria.

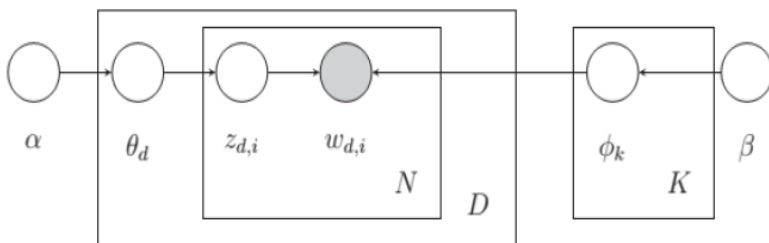
2.17 Metode *Labeled Latent Dirichlet Allocation (LLDA)*

Metode *Labeled Latent Dirichlet Allocation* atau biasa disebut dengan metode L-LDA merupakan model umum probabilitas untuk multilabel corpora pada kumpulan dokumen. Metode L-LDA merupakan perkembangan dari metode *Latent Dirichlet Allocation (LDA)* yang digagas oleh David Blei pada tahun 2003 [30].

Metode LDA sendiri merupakan metode yang bersifat *unsupervised*, dimana tidak membutuhkan pembelajaran sebelumnya sebagai acuan kebenarannya. Metode LDA merupakan salah satu model untuk pemodelan topik [30]. Tujuan utama dari metode LDA ini adalah untuk menyimpulkan informasi utama yang tersembunyi dalam dokumen. Dalam model ini, terdapat 2 buah aspek bernilai penting dalam menyimpulkan informasi utama, yaitu aspek laten (topik/tersembunyi) dapat menjelaskan aspek observasi (teramati). Aspek observasi merupakan dokumen yang diamati, sedangkan aspek laten merupakan topik yang ditentukan dari setiap kata pada dokumen yang diamati [31].

Tujuan dari penggunaan metode L-LDA adalah sebagai pengganti penggunaan kriteria yang bersifat kualitatif dalam pemeringkatan perguruan tinggi yaitu reputasi akademik. Reputasi akademik sebelumnya menggunakan metode survey yang cenderung dikerjakan secara manual [4], dengan metode ini akan diotomatisasi dengan memodelkan topik dari dataset abstrak paper akademisi perguruan tinggi. Abstrak tersebut termasuk dalam kategori/topik apa, yang nantinya setiap topik akan mempunyai bobot tersendiri, sehingga setiap topik dapat dikategorikan untuk menghasilkan scoring pada kriteria reputasi akademik.

Model generatif LDA dalam diagram model generatif [30] ditunjukkan pada Gambar 2.17-1.



Gambar 2.17-1 Diagram Representasi Model Generatif LDA

Berdasarkan diagram diatas didapat distribusi bersama (*joint-distribution*) yang dapat direpresentasikan dalam Persamaan (1) [32]- [30].

$$p(w, z, \theta, \phi|\alpha, \beta) = p(\phi|\beta) p(\theta|\alpha) p(z|\theta) p(w|\phi_z) \quad (1)$$

Distribusi Dirichlet merupakan distribusi atas vektor, dimana elemen vektor merupakan bilangan diantara 0 dan 1, dengan kata lain dapat dianggap sebagai bilangan probabilitas. Persamaan distribusi Dirichlet seperti pada Persamaan (2) [32].

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha)}{\prod_{i=1}^k \Gamma(\alpha)} \theta^{\alpha_1-1} \dots \theta^{\alpha_k-1} \quad (2)$$

Untuk menemukan aspek laten (topik/teramati) sebagai estimasi parameter metode LDA dapat ditemukan dengan metode estimasi Gibbs Sampling. Metode estimasi yang dapat digunakan dalam model LDA berdasarkan metode analisis informasi sampel dan informasi prior (distribusi posterior). Cara ini merupakan pendekatan simulasi untuk membangun distribusi bersama berdasarkan distribusi marginal, dalam proses estimasi parameter LDA [32].

Gibbs Sampling merupakan algoritma dari Markov Chain Monte Carlo (MCMC) dimana mempunyai konsep untuk menghitung joint probability distribution dengan cara melakukan sampling satu persatu terhadap setiap variabel berdasarkan nilai variabel lainnya. Gibbs Sampling ini memerlukan nilai peluang topik yang diasosiasikan untuk sebuah kata [33]. Persamaan Gibbs Sampling pada model LDA seperti pada Persamaan (3).

$$P(z_i = t | z_{-i}, w_i) \propto \frac{C_{-i,j}^{(w)} + \beta}{\sum_{w=1}^W C_{-i,t}^{(w)} + W\beta} \frac{C_{-i,t}^{(m_i)} + \alpha}{\sum_{m_i=1}^{m_i} C_{-i}^{m_i} + T\alpha} \quad (3)$$

Keterangan:

$C_{-i,j}^{(w)}$ = jumlah kata terhadap topik

$C_{-i,t}^{(m_i)}$ = jumlah dokumen terhadap topik

β = parameter penyeimbang jumlah kata terhadap topik, parameter ini diisikan bebas oleh user

α = parameter penyeimbang jumlah dokumen terhadap topik, parameter ini diisikan bebas oleh user

T = jumlah topik yang diinisiasi

W = jumlah kata yang diinisiasi

Berikut adalah proses generativ dari metode LDA [33] [30]:

Fase inisialisasi:

1. untuk setiap dokumen (dm) pada kumpulan dokumen (kd):
 - a. untuk setiap kata (w) dalam dokumen (dm):
 - i. membuat topik sample (z) secara random $\sim Mult(\frac{1}{K})$
 - ii. menambahkan $n_{-i,t}^{(w)}$, $n_{-i,t}^{(\cdot)}$, $n_{-i,t}^{(dm_i)}$, dan $n_{-i}^{(dm_i)}$

Fase inferensi kesimpulan Gibbs Sampling:

1. untuk setiap iterasi:
 - a. untuk semua dokumen (dm) dalam kumpulan dokumen (kd):
 - i. untuk semua kata (w) dalam dokumen (dm):
 1. menghapus nilai topik sample (z) saat ini pada kata (w)
 2. mengurangi $n_{-i,t}^{(w)}$, $n_{-i,t}^{(\cdot)}$, $n_{-i,t}^{(dm_i)}$, dan $n_{-i}^{(dm_i)}$
 3. untuk setiap topik (j), perulangan dari 0 hingga K-1:
 - a. menjumlahkan $p(z_i = t | z_{-i}, w_i)$
 4. membuat topik baru $z \sim p(z_i = t | z_{-i}, w_i)$
 5. memasukkan topik (z) pada kata (w)
 6. menambahkan $n_{-i,t}^{(w)}$, $n_{-i,t}^{(\cdot)}$, $n_{-i,t}^{(dm_i)}$, dan $n_{-i}^{(dm_i)}$

Dalam proses generative LDA, proses komputasi distribusi posterior pada fase inferensi Gibbs Sampling pada langkah ke-4 menggunakan persamaan 3. Setelah estimasi inferensi Gibbs Sampling selesai, maka akan masuk dalam proses distribusi probabilitas topik pada kumpulan dokumen dengan menghitung nilai dari probabilitas topik terhadap dokumen $\vartheta^{(d_i)}$ dan probabilitas topik terhadap kata φ_j^w seperti pada Persamaan (4) dan (5) [33].

$$\vartheta^{(d_i)} = \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i}^{(d_i)} + T\alpha} \quad (4)$$

$$\varphi_j^w = \frac{n_{-i,j}^{(w)} + \beta}{n_{-i,j}^{(c)} + W\beta} \quad (5)$$

Keterangan:

$\vartheta^{(d_i)}$ = probabilitas dokumen terhadap topik

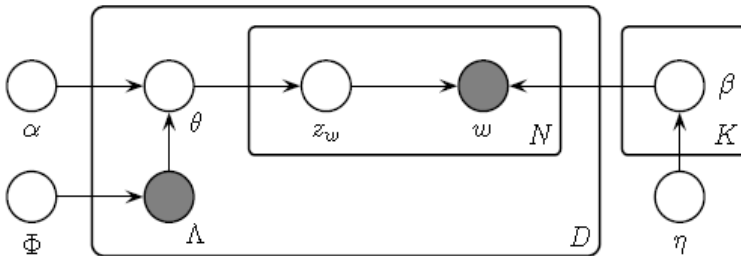
φ_j^w = probabilitas kata terhadap topik

Dalam tugas akhir ini menerapkan metode L-LDA sebagai metode pemodelan topik utama. Metode L-LDA sendiri merupakan model generatif untuk multilabel corpora, dimana menekankan pada penyediaan label sebelum melakukan komputasi [6]. Tujuan utama dari metode L-LDA sama seperti model LDA, yaitu sama-sama menyimpulkan informasi utama yang tersembunyi dalam dokumen.

Dalam model L-LDA terlebih dahulu membuat dokumen pelabelan untuk menghimpun topik yang akan digunakan sebagai langkah pengelompokan/klasifikasi. Setiap label berhubungan langsung dengan satu topik, sehingga setiap topik digambarkan

sebagai satu label [6]. Setiap dokumen dalam metode L-LDA sebagai campuran topik yang mendasari serta topik yang dihasilkan pasti sesuai dalam lingkup label topik yang ditentukan [6].

Untuk representasi model generative *Labeled Latent Dirichlet Allocation* (L-LDA) dijelaskan pada Gambar 2.17-2.



Gambar 2.17-2 Diagram model generativ L-LDA

Konsep metode L-LDA adalah setiap dokumen direpresenyasikan dari kumpulan list kata dan list indikator binary topik (0, 1). Berikut adalah proses generative dari metode L-LDA [6]:

Fase inisialisasi topik:

1. untuk setiap topik (k), $k \in \{1, \dots, K\}$:
 - a. menghasilkan kosakata (vocabulary) dengan topik (k)

$$\sim \text{Dirichlet}(\cdot | \eta)$$

$$\beta_k = (\beta_{k,1}, \dots, \beta_{k,V})^T \sim \text{Dirichlet}(\cdot | \eta)$$

Fase inferensi kesimpulan Gibbs Sampling:

1. untuk setiap dokumen (dm):
 - i. untuk setiap topik (k), $k \in \{1, \dots, K\}$:
 - i. generate label indikator binary topik $A^{(dm)} \in \{0, 1\} \sim \text{Ber}(\cdot | \Phi_k)$

- b. generate dimensi vektor yang lebih rendah untuk memproyeksikan parameter vektor dari distribusi Dirichlet $\alpha^{(dm)} = L^{(dm)} \alpha$
- c. menghasilkan distribusi multinomial campuran $\theta^{(dm)} = (\theta_{t_1}, \dots, \theta_{t_{M^{dm}}})^T \sim Dir(\cdot | \alpha^{(dm)})$
- d. untuk setiap iterasi sejumlah panjang kata dalam dokumen $\{1, \dots, N_d\}$:
 - i. menghasilkan topik baru $z_i \in \{\lambda_1^{(d)}, \dots, \lambda_{M^d}^{(d)}\} \sim Mult(\cdot | \theta^{(d)})$
 - ii. menghasilkan list kata baru $w_i \in \{1, \dots, V\} \sim Mult(\cdot | \beta_{z_i})$

Persamaan Gibbs Sampling yang digunakan pada L-LDA mengacu pada (Griffiths dan Steyvers, 2004) pada Persamaan (6).

$$P(z_i = j | z_{-i}) \propto \frac{n_{-i,j}^{w_i} + \eta_{w_i}}{\binom{\cdot}{n_{-i,j} + \eta^T \mathbf{1}}} \times \frac{n_{-i,j}^{(dm)} + \alpha_j}{n_{-i,\cdot}^{(dm)} + \alpha^T \mathbf{1}} \quad (6)$$

2.18 Macam-macam Pengembangan Metode LDA

Metode Latent Dirichlet Allocation (LDA) merupakan salah satu metode topik modeling yang mudah untuk dikembangkan. Penambahan terhadap variabel penilaian, aspek keakuratan, serta kondisi kasus yang berkembang merupakan beberapa hal yang mengharuskan terus adanya perkembangan dalam sebuah metode *machine learning*. Keunggulan dan hasil yang lebih bagus menjadi target yang harus dikejar oleh akademisi pengembang. Beberapa versi metode LDA yang sudah ada diantaranya sebagai berikut:

1. Regular Latent Dirichlet Allocation (LDA) [30]
2. Supervised Latent Dirichlet Allocation (sLDA) [34]
3. Discriminative Latent Dirichlet Allocation (DiscLDA) [35]
4. Mixture Latent Dirichlet Allocation (MM-LDA) [36]
5. Labeled Latent Dirichlet Allocation (L-LDA) [6]

2.19 *Term Frequency – Invers Document Frequency (TF-IDF)*

Term Frequency – Invers Document Frequency atau biasa disebut dengan TF-IDF merupakan sebuah metode untuk mendapatkan bobot sebuah term dari sebuah corpus. Metode ini terkenal efisien, mudah dan memiliki hasil yang akurat. TF-IDF didapatkan dengan mencari nilai *Term Frequency* (TF) dan nilai *Invers Document Frequency* (IDF) yang terlebih dahulu mencari nilai *Document Frequency* (DF) [37].

2.19.1 **Term Frequency (TF)**

Merupakan jumlah kemunculan suatu *term* dalam dokumen. Nilai TF kemudian dilogaritmikkan untuk mengurangi besarnya bilangan. Persamaan mencari TF seperti Persamaan (9).

$$Tf = tdfd \quad (9)$$

2.19.2 **Invers Documenten Frequency (IDF)**

Sebelum mencari IDF terlebih dahulu mencari nilai DF yaitu jumlah dokumen dimana sebuah term muncul. Setelah mendapatkan nilai DF kemudian menghitung nilai IDF yang berfungsi mengurangi bobot suatu term jika kemunculannya banyak tersebar di seluruh koleksi dokumen [37]. Persamaan DF seperti pada Persamaan (10), dan IDF pada Persamaan (11).

$$Df = dft \quad (10)$$

$$Idf_t = \log_{10}\left(\frac{N}{df_t}\right) \quad (11)$$

Keterangan:

- Tf = term frequency
- Df = dokumen frequency
- IDF = invers dokumen requency
- N = jumlah dokumen

2.20 Fase Teks Mining (Metode L-LDA) Studi Kasus Perneriksaan Perguruan Tinggi

Terdapat 6 tahap utama dalam alur metode *Labeled Latent Dirichlet Allocation* (L-LDA), dari dokumen mentah hingga menjadi bobot final reputasi akademik perguruan tinggi, diantaranya sebagai berikut:

1. Tahap preprocessing

Tahap preprocessing merupakan proses untuk membersihkan kata (*term*), dimana kumpulan dokumen abstrak paper yang terdiri dari banyak dokumen akan dikelola diambil kata bakunya saja. Kata baku menunjukkan komposisi keberadaan kata (*term*) dalam dokumen [22]. Tahap preprocessing terdiri dari beberapa sub-proses diantaranya:

1.1. Case folding

Merupakan proses merubah huruf bertipe besar (*uppercase*) menjadi bertipe kecil (*lowercase*) dalam sebuah corpus dokumen.

1.1. Tokenisasi

Merupakan proses pemecahan kumpulan kalimat menjadi kata (*term*) yang berdiri sendiri-sendiri (satu suku kata).

1.2. Remove punctuation

Merupakan proses penghapusan karakter yang dapat disesuaikan pengguna. Pada umumnya karakter yang dihapus adalah karakter non-huruf (tanda baca - angka).

1.3. Stopwords

Merupakan proses penghapusan kata (*term*) yang tidak begitu penting keberadaanya didalam dokumen. Sehingga dapat memaksimalkan keberadaan kata yang penting.

1.4. Stemming

Merupakan proses pemotongan imbuhan (*prefix, suffix, infix*) dari sebuah kata (*term*) dalam sebuah dokumen.

Hasil dari tahap preprocessing berupa kumpulan kata baku dan telah bersih kemudian hasil kata ini akan menjadi bahan

pairing dalam pencocokan Term Frequency (TF) untuk proses Menentukan asumsi label awal dokumen.

2. Tahap memperkaya corpus TKT

Corpus TKT merupakan dokumen kumpulan kata kunci (*keyword*) yang merepresentasikan kelas dari *term* pada dokumen dataset untuk masuk kedalam topik Tingkat Kesiapterapan Teknologi (TKT) yang mempunyai 9 kelas. Corpus TKT mengadopsi corpus Bloom Taxonomy yang terdiri dari 6 kelas [29]. Untuk dapat digunakan sebagai corpus dalam teks mining metode L-LDA, kelas corpus dalam Bloom Taxonomy perlu dipecah menjadi 9 kelas sesuai kelas TKT [5]. Breakdown menggunakan pengurutan aspek konten dari setiap *keyword*, dimana *keyword* yang mempunyai aspek tipe dasar berada pada level bawah dan aspek tipe penerapan berada pada level atas.

Selain menggunakan breakdown terhadap 6 corpus Bloom Taxonomy menjadi 9 corpus TKT, dalam Tugas Akhir ini dilakukan juga pengembangan corpus dari beberapa cara. Cara pertama mengembangkan berdasarkan adopsi teknik POS Tagging dokumen abstrak. POS tagging akan memecah sebuah kalimat untuk diambil kelas tipe kata seperti *verb*, *noun*, *adjective*, *adverb* dan sebagainya [28]. Sebelum melakukan POS Tagging, dokumen abstrak paper akan dilakukan pencocokan terhadap *keyword* di masing-masing corpus TKT. setelah melakukan pencocokan dan ditemukan kecocokan, selanjutnya akan dicek terlebih dahulu kata (*term*) tersebut tergolong pada kelas kata apa dari prinsip POS Tagging, setelah diketahui kelas tipe kata kemudian akan mengakuisisi kata (*term*) lain yang berdekatan dengan kata (*term*) tersebut dengan jangkauan akuisisi beragam, mulai dari akuisisi 1 hingga 10. Dalam percobaan selanjutnya acuan menggunakan akuisisi 1. Kelas kata (*term*) hasil akuisisi akan masuk pada corpus yang sama dengan kata yang dicocokkan sebelumnya.

Untuk cara kedua mengembangkan berdasarkan melihat sinonim dari setiap kata yang ada pada 9 corpus TKT. Setiap kata (*term*) akan dicari sinonimnya menggunakan library WordNet dalam bahasa pemrograman Python [24]. Terdapat 4 tipe percobaan dalam menghasilkan corpus baru dengan mengacu pada pengembangan corpus dari segi sinonim, percobaan pertama dengan melihat sinonim level 1 dari setiap kata yang ada pada 9 corpus TKT. Sinonim level 1 didapatkan dari satu nilai similaritas paling tinggi antara sinonim. Percobaan kedua dengan melihat sinonim level 2 dari setiap kata. Sinonim level 2 didapatkan dari dua nilai similaritas tertinggi. Percobaan 3 menggunakan prinsip yang sama hanya saja merubahnya menjadi tiga nilai similaritas tertinggi. Sedangkan yang terakhir adalah menggunakan sinonim level all, dimana dalam percobaan ini akan diambil semua sinonim dari setiap kata yang ada pada 9 corpus TKT. hasil dari pencarian sinonim akan ditambahkan pada corpus TKT final.

Dalam Tugas Akhir ini disimpulkan menggunakan proses memperkaya corpus TKT dengan cara mengembangkan dari segi sinonim pada percobaan sinonim level 3 dan all sebagai perbandingan. Karena berdasar hasil analisa akhir menunjukkan bahwa *distance* atau bisa disebut hasil akhir dari kedua percobaan ini lebih bagus dari percobaan lainnya.

3. Tahap penentuan asumsi topik awal menggunakan Term Frequency (TF)

Tahap ini dilakukan dengan menghitung kemunculan term paling banyak dari dokumen abstrak *paper* pada 9 corpus TKT final [37]. Corpus TKT final yang digunakan berasal dari beberapa percobaan dalam tahap sebelumnya, diantaranya menggunakan corpus TKT hasil memperkaya menggunakan teknik POS Tagging akuisisi 1, percobaan sinonim WordNet 1, percobaan sinonim WordNet 2, percobaan sinonim WordNet 3, dan percobaan sinonim

WordNet all. Dari semua percobaan ini disimpulkan bahwa menggunakan percobaan 3 dan all lebih bagus dalam menghasilkan distance hasil akhir.

Dokumen yang mempunyai kecocokan berdasarkan *Term Frequency* (TF) akan dicatat jumlah term yang sesuai, dan diurutkan berdasarkan jumlah kata yang paling tinggi hingga yang paling rendah. Kecocokan ini menandakan bahwa dokumen relevan berdasarkan keyword dengan ke-9 corpus TKT. 3 kecocokan teratas antara dokumen dan corpus TKT akan digunakan sebagai asumsi label awal dokumen untuk masukan tahap inferensi Gibbs Sampling L-LDA.

4. Tahap generative model L-LDA dan inferensi Gibbs Sampling

Tahap ini merupakan proses ini untuk menghasilkan probabilitas akhir setiap label yang menjadi topik final. Masukan pada tahap ini berupa asumsi label topik yang diharuskan lebih dari satu label beserta dokumen abstrak paper. Asumsi label topik ini akan diproses menggunakan threshold, iterasi dan ketentuan variabel masukan lainnya yang ditentukan oleh user [6]. Penggunaan metode L-LDA berguna untuk memastikan asumsi label mempunyai probabilitas tertinggi agar berada pada satu label saja dari sebuah dokumen uji. Hasil keluaran proses L-LDA berupa probabilitas term dari dokumen abstrak dengan asumsi label topik, serta probabilitas dokumen abstrak dengan asumsi label topik. Dalam Tugas Akhir ini, keluaran yang berperan penting untuk proses selanjutnya adalah probabilitas dokumen abstrak dengan label topik.

5. Tahap penentuan label topik final dan pembobotan

Pada tahap ini telah didapatkan topik final dari asumsi label topik yang dijadikan batasan uji Gibbs Sampling. Masing-masing asumsi label topik ini berguna untuk proses pembobotan. Setiap asumsi label topik hasil keluaran L-LDA beserta probabilitas akan dihitung sebagai masukan tahap

penghitungan skor final. Hasil penghitungan ini akan mencerminkan skor kriteria reputasi akademik dari perguruan tinggi.

2.21 Akreditasi Perguruan Tinggi

Akreditasi perguruan tinggi merupakan salah satu cara untuk menilai kualitas perguruan tinggi secara keseluruhan, yang bertujuan untuk mengetahui komitmen perguruan tinggi terhadap kapasitas dan efektifitas pendidikan yang diselenggarakan, yang didasarkan pada standar akreditasi yang telah ditetapkan oleh badan akreditasi nasional [38].

Di Indonesia pihak yang berwenang untuk melakukan akreditasi institusi perguruan tinggi adalah Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Akreditasi ini dilakukan oleh tim asesor yang turun langsung kepada perguruan tinggi terkait untuk dinilai dari beberapa hal. Berikut adalah poin-poin penting dalam penilaian akreditasi institusi oleh BAN-PT [38].

1. Menilai kesesuaian visi, misi, tujuan dan sasaran, serta strategi pencapaian dalam perguruan tinggi dengan kondisi yang sudah ada sekarang.
2. Menilai tata pamong (kepengurusan), kepemimpinan, sistem pengelolaan, dan penjaminan mutu.
3. Menilai kualitas mahasiswa dan lulusan.
4. Menilai sumber daya manusia (tenaga pengajar dan administrasi).
5. Menilai kesesuaian kurikulum, model pembelajaran, serta suasana akademik.
6. Menilai kebutuhan pembiayaan, sarana dan prasarana, serta sistem informasi dalam perguruan tinggi.
7. Menilai tentang penelitian, pelayanan/pengabdian masyarakat, serta kerjasama.

Penilaian akreditasi ini dilakukan dengan cara survey langsung terhadap perguruan tinggi, sehingga data yang didapatkan dapat objektif terhadap perguruan tinggi tersebut. Akreditasi yang

dilakukan oleh BAN-PT tersebut mempunyai 2 jenis, yaitu akreditasi institusi dan akreditasi profesi.

Akreditasi institusi yang merupakan penilaian kualitas institusi dengan memperhatikan poin-poin penilaian diatas. Sedangkan akreditasi profesi merupakan penilaian kualitas terhadap bidang studi/departemen/jurusan dalam perguruan tinggi tersebut, yang mana akan menentukan kualitas untuk akreditasi institusi juga. Berikut adalah score penilaian akreditasi BAN-PT.

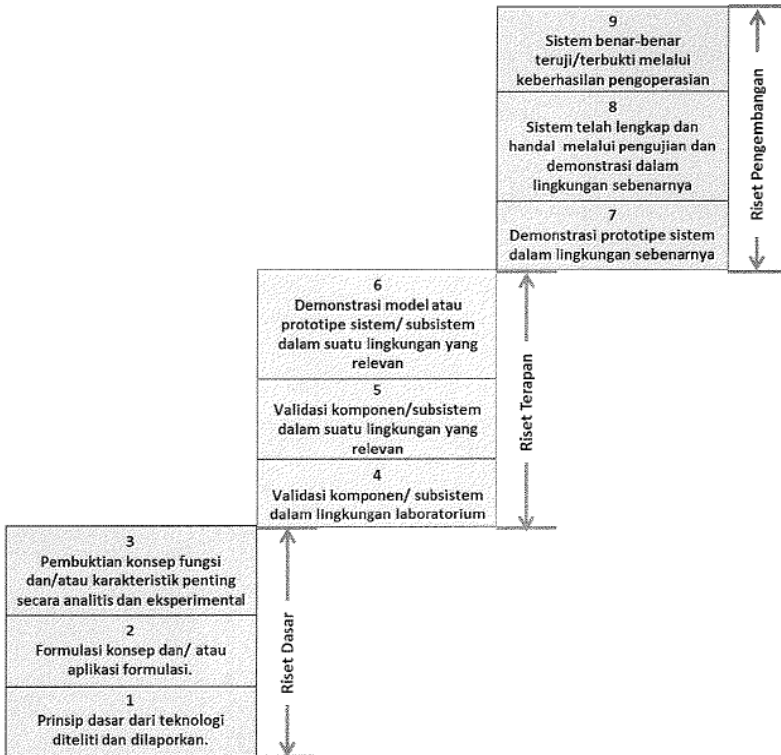
- a) Nilai 601 – 700, mendapatkan peringkat A
- b) Nilai 501 – 600, mendapatkan peringkat B
- c) Nilai 401 – 500, mendapatkan peringkat C
- d) Nilai 0 – 400, mendapatkan peringkat N/A (not applicable)

Dalam Tugas Akhir ini, salah satu kriteria penilaian yang menjadi acuan adalah kriteria reputasi employer. Dengan menggunakan kriteria akreditasi perguruan tinggi oleh BAN-PT yang dijelaskan pada poin penilaian kualitas mahasiswa dan lulusan sudah mengandung kriteria reputasi employer yang dibutuhkan.

2.22 *Tingkat Kesiapterapan Teknologi (TKT)*

Tingkat Kesiapterapan Teknologi atau yang biasa disebut dengan TKT merupakan tingkat kondisi kematangan atau kesiapterapan suatu hasil penelitian (*research*) dan pengembangan *prototype* tertentu yang selanjutnya akan masuk pada tahap komersialisasi, yang diukur secara sistematis dengan tujuan untuk dapat dimanfaatkan oleh pengguna, baik oleh pemerintah, maupun masyarakat luas [5].

Setiap jenis penelitian (*research*) dan pengembangan *prototype* dikelompokkan menjadi 9 tingkatan dengan masing-masing memiliki indikator yang berbeda-beda. Seperti pada Gambar 2.22-1.



Gambar 2.22-1 Hirarki Tingkat Kesiapterapan Teknologi (TKT)

Berdasarkan hirarki tingkatan diatas, dapat dikelompokkan menjadi 3 kelompok tingkatan besar [5], diantaranya.

1. Riset dasar untuk TKT tingkat 1-3

Dalam kelompok ini, penelitian yang dimaksud adalah penelitian yang berfokus pada pembahasan metode, pengembangan teori, pembahasan gagasan produk. Penelitian ini akan bertujuan untuk membuat produk yang bermanfaat bagi masyarakat.

2. Riset terapan untuk TKT tingkat 4-6

Dalam kelompok ini, penelitian yang dimaksud adalah penelitian yang berfokus pada yang menghasilkan produk, demonstrasi model atau *prototype* subsistem dalam lingkungan yang relevan, dan berfokus pada validasi dalam lingkungan laboratorium. Penelitian ini sudah membuat produk/prototype.

3. Riset pengembangan untuk TKT tingkat 7-9

Dalam kelompok ini, penelitian yang dimaksud adalah penelitian yang berfokus pada pengembangan produk yang sudah ada sehingga membuat riset pendukung baru untuk produk yang sudah ada.

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini diuraikan tahap analisis kebutuhan dan perancangan perangkat lunak dalam Tugas Akhir. Tahap analisis membahas mengenai analisis kebutuhan yang menjadi dasar dari tahap perancangan sistem. Perancangan sistem direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1. Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan perangkat lunak pemeringkatan perguruan tinggi. Terdapat beberapa bagian antara lain, deskripsi umum, aktor dalam sistem, spesifikasi kebutuhan sistem, kasus penggunaan, dan dataset pengujian sistem.

3.1.1 Deskripsi Umum

Pada Tugas Akhir ini dibangun perangkat lunak pemeringkatan perguruan tinggi di Indonesia untuk mengukur Tingkat Kesiapterapan Teknologi (TKT) menggunakan metode *Labeled Latent Dirichlet Allocation* (L-LDA). Perangkat lunak yang dibangun berbasis pada aplikasi website.

Data masukan yang digunakan adalah data yang mewakili berbagai macam kriteria penilaian perguruan tinggi yang mengacu pada *QS World University Rankings*, diantaranya adalah dokumen abstrak perguruan tinggi yang akan diolah menggunakan proses teks mining, data hasil akreditasi institusi yang didapatkan dari data online BAN-PT, data rekap jumlah mahasiswa perguruan tinggi, data rekap jumlah tenaga pengajar dari berbagai spesifikasi (status ketenagakerjaan, pendidikan terakhir, dan jabatan akademik), data jumlah tenaga kerja Phd (S3) yang didapatkan dari database PDDIKTI, data jumlah sitasi akademisi perguruan tinggi, data kualitas sitasi yang dilihat dari skor h-index dan i10-index akademisi perguruan tinggi, data jumlah publikasi yang terindex oleh Scopus yang didapatkan dari database SINTA, jumlah backlink *webhost*, serta skor Page Authority dan Domain

Authority *webhost* perguruan tinggi yang didapatkan dengan analisa *webhost*. Data keluaran dari perangkat lunak ini merupakan informasi peringkat perguruan tinggi di Indonesia berdasarkan pembobotan pada data inputan yang digunakan sebagai penilaian posisi peringkat, selain itu juga menghasilkan detail informasi tambahan mengenai kualitas institusi, akademisi dan kemahasiswaan.

3.1.2 Aktor Dalam Sitem

Terdapat 1 aktor dalam sistem pemeringkatan perguruan tinggi ini, yaitu pengguna sistem perangkat lunak website (*End User*). End User mempunyai hak akses dalam menentukan bobot masukan untuk pembobotan perguruan tinggi, melihat pemeringkatan berdasarkan metode dan tipe pemeringkatan lain, mencari perguruan tinggi berdasarkan kebutuhan, serta melihat detail informasi akademik perguruan tinggi. End User sebagai pengguna tunggal dalam sirkulasi sistem perangkat lunak.

3.1.3 Spesifikasi Kebutuhan Sistem

Pada perangkat lunak pemeringkatan perguruan tinggi di Indonesia untuk mengukur Tingkat Kesiapterapan Teknologi (TKT) menggunakan metode *Labeled Latent Dirichlet Allocation* (L-LDA) ini mempunyai spesifikasi kebutuhan yang mencakup kebutuhan fungsional sistem. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan agar menghasilkan pemeringkatan yang akurat terhadap *Ground Truth* yang telah ditetapkan. Kebutuhan fungsional tersebut dideskripsikan dalam Tabel 1-1.

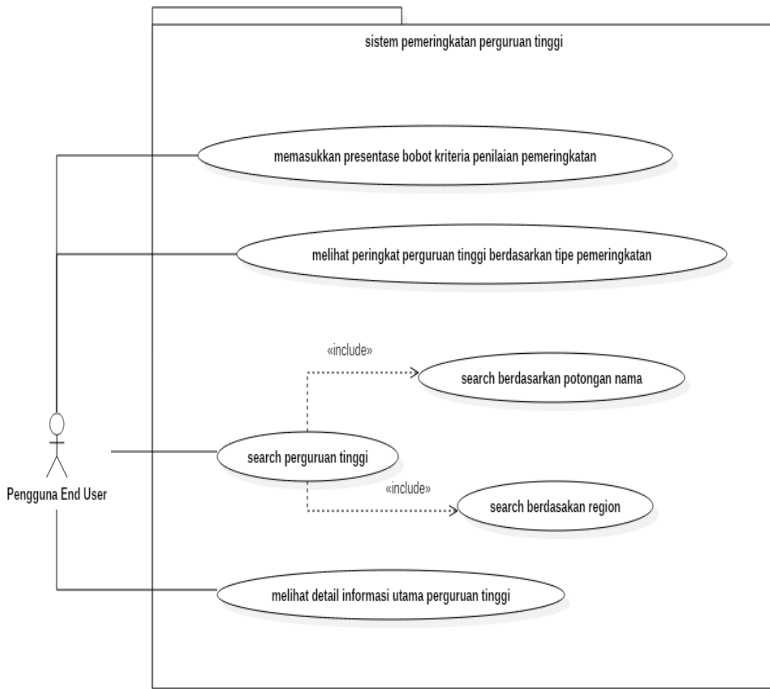
Tabel 1-1 Daftar Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi	Aktor
F-001	Memasukkan presentase bobot kriteria penilaian	Pengguna dapat memasukkan presentase bobot pemeringkatan sesuai dengan kebutuhan.	End User

	pemeringkatan		
F-002	Melihat peringkat perguruan tinggi berdasarkan tipe pemeringkatan	Pengguna dapat melihat peringkat perguruan tinggi dengan menggunakan bobot yang dipilih sesuai dengan tipe pemeringkatan.	End User
F-003	Mencari perguruan tinggi dari database berdasarkan potongan nama	Pengguna dapat mencari perguruan tinggi dengan memasukkan keyword berupa potongan nama perguruan tinggi dalam kotak pencarian search box.	End User
F-004	Mencari perguruan tinggi dari database berdasarkan region wilayah provinsi	Pengguna dapat mencari perguruan tinggi dengan memilih region wilayah provinsi yang disediakan dalam map peta Indonesia dalam 33 provinsi.	End User
F-005	Melihat detail informasi akademik perguruan tinggi	Pengguna dapat melihat informasi detail dari perguruan tinggi, yang dapat dikelompokkan menjadi 6 kategori, diantaranya kepopuleran webhost, mahasiswa, akreditasi, produktifitas institusi, kecukupan dosen pengajar, dan kualitas akademisi.	End User

3.1.4 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang sudah dilakukan. Diagram kasus penggunaan yang mewakili kebutuhan sistem dapat dilihat pada Gambar 1-1 dan kode kasus penggunaan pada Tabel 1-2.



Gambar 1-1 Diagram Kasus Penggunaan

Tabel 1-2 Daftar Kebutuhan Fungsional Sistem

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Memasukkan presentase bobot kriteria penilaian pemeringkatan
UC-002	Melihat peringkat perguruan tinggi berdasarkan tipe pemeringkatan
UC-003	Mencari perguruan tinggi berdasarkan potongan nama
UC-004	Mencari perguruan tinggi berdasarkan region wilayah provinsi

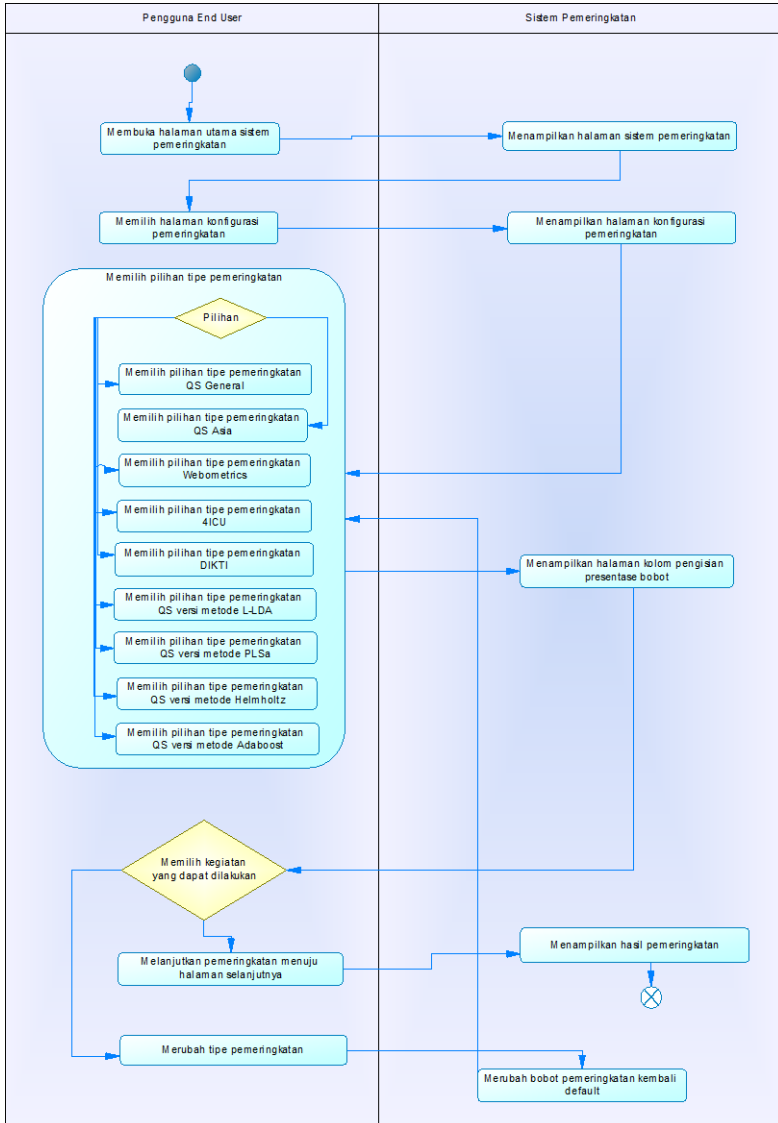
UC-005	Melihat detail informasi utama perguruan tinggi
--------	---

3.1.4.1 UC-001 Memasukkan Presentase Bobot Kriteria Penilaian Pemingkatan

Tabel 1-3 Case Memeasukkan Presentase Bobot Kriteria Penilaian Pemingkatan

Nama	Memasukkan Presentase Bobot Kriteria Penilaian Pemingkatan
Nomor	UC-001
Deskripsi	Kasus penggunaan ini digunakan untuk menentukan bobot pemingkatan pada tipe pemingkatan versi metode LLDA dengan menentukan bobot pemingkatan disetiap kriteria yang diinginkan oleh pengguna
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem
Kondisi Akhir	Sistem menampilkan halaman pemilihan kriteria dan tipe pemingkatan yang sudah mempunyai bobot
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman utama sistem pemingkatan 2. Sistem menampilkan halaman utama sistem pemingkatan 3. Pengguna memilih halaman konfigurasi pemingkatan dengan melakukan scroll kursor kebawah 4. Sistem menampilkan halaman konfigurasi pemingkatan 5. Pengguna End User memilih salah satu pilihan tipe pemingkatan yang terdapat pada halaman konfigurasi pemingkatan <ol style="list-style-type: none"> a. Tipe pemingkatan QS General b. Tipe pemingkatan QS versi Asia c. Tipe pemingkatan Webometrics d. Tipe pemingkatan 4ICU e. Tipe pemingkatan DIKTI f. Tipe pemingkatan QS versi metode LLDA g. Tipe pemingkatan QS versi metode PLSA h. Tipe pemingkatan QS versi metode Helmholtz i. Tipe pemingkatan QS versi metode Adaboost LDA 6. Sistem menampilkan halaman kolom pengisian presentase

	<p>bobot pada tipe pemeringkatan yang dipilih</p> <p>7. Pengguna End User memilih kegiatan yang dapat dilakukan</p> <ul style="list-style-type: none">j. Melanjutkan pemeringkatan menuju halaman selanjutnyak. Merubah tipe pemeringkatan
Alur Alternatif	-

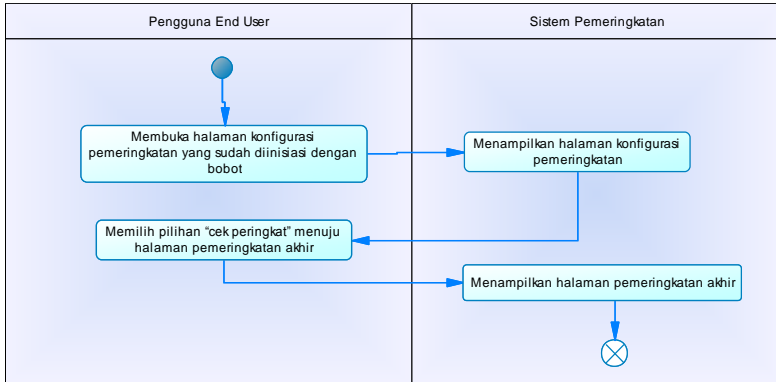


Gambar 1-2 Diagram Aktifitas Use Case UC-001

3.1.4.2 UC-002 Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan

Tabel 1-4 Use Case Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan

Nama	Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan
Nomor	UC-002
Deskripsi	Kasus penggunaan ini digunakan untuk melihat pemeringkatan akhir perguruan tinggi
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem dan memasukkan bobot kriteria penilaian
Kondisi Akhir	Sistem menampilkan halaman akhir yang menampilkan peringkat perguruan tinggi sesuai dengan bobot dan pilihan tipe pemeringkatan yang diinisiasi sebelumnya
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman konfigurasi pemeringkatan yang sudah diinisiasi dengan bobot 2. Sistem menampilkan halaman konfigurasi pemeringkatan yang sudah diinisiasi dengan bobot 3. Pengguna End User memilih pilihan “cek peringkat” menuju halaman pemeringkatan akhir 4. Sistem menampilkan halaman pemeringkatan akhir
Alur Alternatif	-



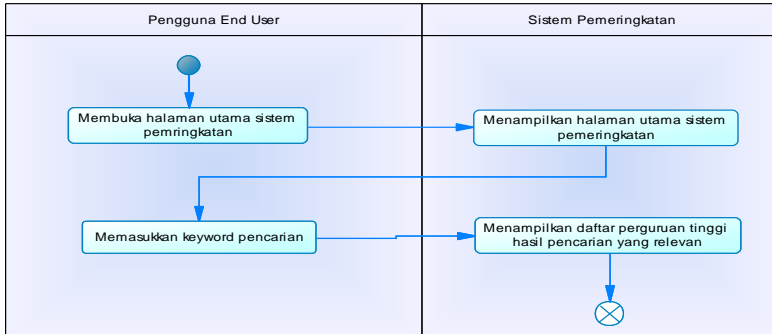
Gambar 1-3 Diagram Aktifitas Use Case UC-002

3.1.4.3 UC-003 Mencari Perguruan Tinggi Berdasarkan Potongan Nama

Tabel 1-5 Use Case Mencari Perguruan Tinggi Berdasarkan Potongan Nama

Nama	Mencari Perguruan Tinggi Berdasarkan Potongan Nama
Nomor	UC-003
Deskripsi	Kasus penggunaan ini digunakan untuk mencari informasi perguruan tinggi berdasarkan masukan berupa potongan nama perguruan tinggi pada kolom search pada halaman utama
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem
Kondisi Akhir	Sistem menampilkan halaman hasil pencarian berdasarkan keyword potongan nama yang dimasukkan dalam kolom search yang berisikan list perguruan tinggi yang relevan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman utama sistem pemingkatan 2. Sistem menampilkan halaman utama sistem pemingkatan 3. Pengguna End User memasukkan keyword pencarian berupa potongan nama pada kolom pencarian

	4. Sistem menampilkan daftar perguruan tinggi hasil pencarian yang relevan dalam halaman result
Alur Alternatif	-



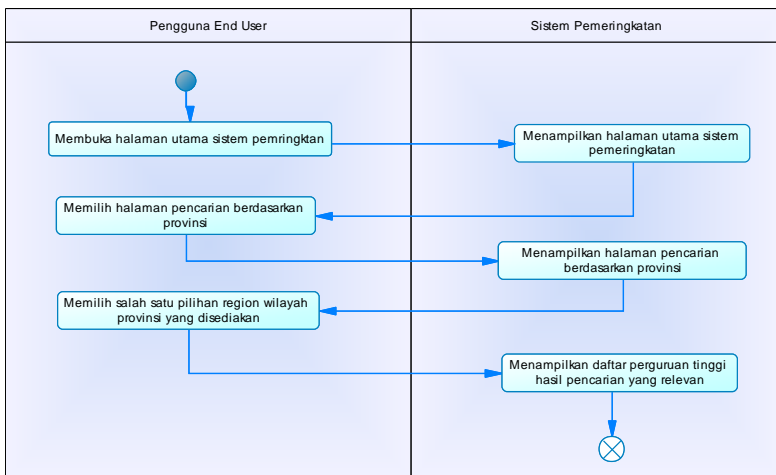
Gambar 1-4 Diagram Aktifitas Use Case UC-003

3.1.4.4 UC-004 Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi

Tabel 1-6 Use Case Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi

Nama	Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi
Nomor	UC-004
Deskripsi	Kasus penggunaan ini digunakan untuk mencari informasi perguruan tinggi berdasarkan kategori region wilayah provinsi perguruan tinggi pada halaman pencarian provinsi
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem
Kondisi Akhir	Sistem menampilkan halaman hasil pencarian berdasarkan region wilayah provinsi yang dipilih pada map provinsi
Alur Normal	1. Pengguna End User membuka halaman utama sistem pemeringkatan

	<ol style="list-style-type: none"> 2. Sistem menampilkan halaman utama sistem pemeringkatan 3. Pengguna End User memilih halaman pencarian berdasarkan provinsi dengan melakukan scroll kursor kebawah 4. Sistem menampilkan halaman pencarian berdasarkan provinsi 5. Pengguna End User memilih salah satu pilihan region wilayah provinsi yang disediakan 6. Sistem menampilkan daftar perguruan tinggi yang berada pada region wilayah provinsi yang dipilih
Alur Alternatif	-



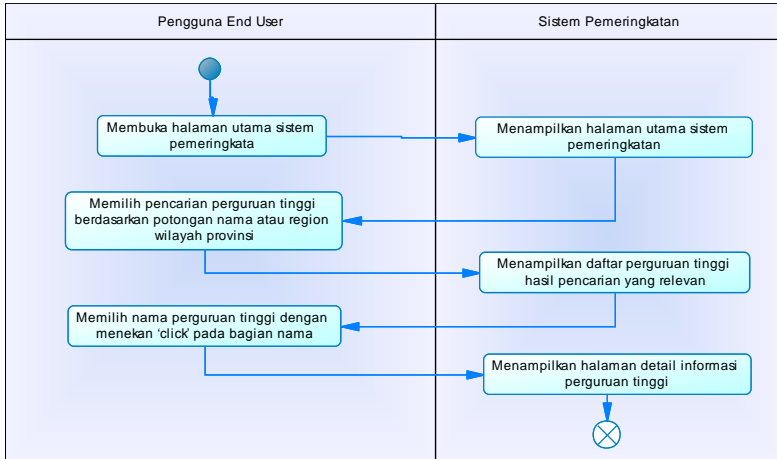
Gambar 1-5 Diagram Aktifitas Use Case UC-004

3.1.4.5 UC-005 Melihat Detail Informasi Utama Perguruan Tinggi

Tabel 1-7 Use Case Melihat Detail Informasi Utama Perguruan Tinggi

Nama	Melihat Detail Informasi Utama Perguruan Tinggi
Nomor	UC-005

Deskripsi	Kasus penggunaan ini digunakan untuk melihat detail informasi yang dimiliki perguruan tinggi, meliputi kepopuleran webhost, mahasiswa, status akreditasi, produktifitas institusi, kecukupan dosen pengajar, dan kualitas akademisi serta dilengkapi dengan grafik tipe pemeringkatan yang dimiliki
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah melakukan pencarian perguruan tinggi berdasarkan potongan nama atau region wilayah provinsi dan sistem menampilkan daftar perguruan tinggi
Kondisi Akhir	Sistem menampilkan halaman detail informasi yang mengandung banyak informasi, diantaranya kepopuleran webhost, mahasiswa, status akreditasi, produktifitas institusi, kecukupan dosen pengajar, dan kualitas akademisi serta dilengkapi dengan grafik tipe pemeringkatan yang dimiliki
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman utama sistem pemeringkatan 1. Sistem menampilkan halaman utama sistem pemeringkatan 2. Pengguna End User memilih pencarian perguruan tinggi berdasarkan potongan nama atau region wilayah provinsi 3. Sistem menampilkan daftar perguruan tinggi hasil pencarian yang relevan pada halaman pencarian 4. Pengguna End User memilih nama perguruan tinggi dengan menekan 'click' pada bagian nama 5. Sistem menampilkan halaman detail informasi perguruan tinggi
Alur Alternatif	-



Gambar 1-6 Diagram Aktifitas Use Case UC-005

3.1.7 Label Kelas dalam Teks Mining Metode L-LDA

Penggunaan metode L-LDA yang bersifat supervised mengharuskan adanya keberadaan kelas atau bisa disebut juga dengan label. Label ini digunakan sebagai penanda pengetahuan dokumen masuk kedalam kondisi seperti apa. Kondisi dalam studi kasus ini adalah hirarki dari Tingkat Kesiapterapan Teknologi (TKT) yang berjumlah 9 level. Berikut adalah daftar kelas yang digunakan dalam Tugas Akhir ini.

Tabel 1-8 Penjelasan Setiap Kelas Label

No.	Hirarki TKT	Kelas/Label	Keterangan
1	Level 1	T1	Prinsip dasar dari teknologi diteliti dan dilaporkan.
2	Level 2	T2	Formulasi konsep dan/atau aplikasi formulasi.
3	Level 3	T3	Pembuktian konsep fungsi dan/atau karakteristik penting secara analisa dan eksperimental
4	Level 4	T4	Validasi komponen atau subsistem dalam lingkup laboratorium
5	Level 5	T5	Validasi komponen atau subsistem dalam lingkungan yang relevan
6	Level 6	T6	Demonstrasi model atau prototype sistem atau subsistem dalam lingkungan yang relevan
7	Level 7	T7	Demonstrasi model dalam lingkungan yang sebenarnya
8	Level 8	T8	Sistem telah lengkap dan handal melalui pengujian dan demonstrasi dalam lingkungan sebenarnya
9	Level 9	T9	Sistem benar-benar teruji

			melalui keberhasilan pengoperasian
--	--	--	------------------------------------

3.1.8 Ground Truth Pengujian Sistem

3.1.8.1 Peringkat QS World University Rankings

Dalam Tugas Akhir ini pemeringkatan keseluruhan dengan menggunakan 6 indikator penilaian mengacu pada kebenaran peringkat QS World University Rankings tahun 2016-2017 sesuai dengan waktu pengerjaan Tugas Akhir ini dimulai. Selain itu melakukan uji coba perbandingan jika mengacu pada Ground Truth peringkat QS World University Rankings dari beberapa tahun yang berbeda. Berikut adalah beberapa peringkat QS World University Rankings dan peringkat reputasi akademik dari berbagai jangka waktu.

1. QS 2015-2015

Peringkat universitas wilayah Indonesia berdasarkan QS World University Rankings 2015-2016.

Tabel 1-9 Peringkat QS World University Rankings 2015-2016

Peringkat QS	Nama Perguruan Tinggi
1	Universitas Indonesia (UI)
2	Institut Teknologi Bandung (ITB)
3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Padjadjaran (UNPAD)
7	Universitas Diponegoro (UNDIP)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Institut Teknologi Sepuluh Nopember (ITS)
10	Universitas Brawijaya (UB)
11	Universitas Bina Nusantara (BINUS)

Tabel 1-10 Peringkat Reputasi Akademik QS 2015-2016

Peringkat Reputasi Akademik	Nama Perguruan Tinggi
1	Institut Teknologi Bandung (ITB)
2	Universitas Indonesia (UI)
3	Universitas Gadjah Mada (UGM)
4	Universitas Muhammadiyah Surakarta (UMS)
5	Universitas Airlangga (UNAIR)
6	Institu Pertanian Bogor (IPB)
7	Universitas Diponegoro (UNDIP)
8	Universitas Brawijaya (UB)
9	Institut Teknologi Sepuluh Nopember (ITS)
10	Universitas Padjadjaran (UNPAD)
11	Universitas Islam Indonesia (UII)
12	Universitas 17 Agustus 1945 (UNTAG)
13	Universitas Bina Nusantara (BINUS)
14	Universitas Pelita Harapan (UPH)
15	Universitas Sebelas Maret (UNS)
16	Universitas Mataram (UNRAM)

2. QS 2016-2017

Peringkat universitas wilayah Indonesia berdasarkan QS World University Rankings 2016-2017.

Tabel 1-11 Peringkat QS World University Rankings 2016-2017

Peringkat QS	Nama Perguruan Tinggi
1	Universitas Indonesia (UI)
2	Institut Teknologi Bandung (ITB)
3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)

5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

Tabel 1-12 Peringkat Reputasi Akademik QS 2016-2017

Peringkat Reputasi Akademik	Nama Perguruan Tinggi
1	Institut Teknologi Bandung (ITB)
2	Universitas Indonesia (UI)
3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

3. QS 2017-2018

Peringkat universitas wilayah Indonesia berdasarkan QS World University Rankings 2017-2018.

Tabel 1-13 Peringkat QS World University Rankings 2017-2018

Peringkat QS	Nama Perguruan Tinggi
1	Universitas Indonesia (UI)
2	Institut Teknologi Bandung (ITB)
3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)

5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

Tabel 1-14 Peringkat Reputasi Akademik QS 2017-2018

Peringkat Reputasi Akademik	Nama Perguruan Tinggi
1	Institut Teknologi Bandung (ITB)
2	Universitas Gadjah Mada (UGM)
3	Universitas Indonesia (UI)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

3.1.8.2 Analisa Label Topik Dokumen

Penggunaan teks mining menggunakan metode *Labeled Latent Dirichlet Allocation* (L-LDA) membutuhkan acuan kebenaran sebagai perbandingan apakah keluaran sistem yang dibangun sudah sesuai dengan kebenaran dalam melakukan penentuan topik dalam indikator reputasi akademik.

Ground Truth yang digunakan adalah hasil analisa manual terhadap dokumen abstrak *paper* akademisi perguruan tinggi. Terdapat 800 dokumen sebagai dataset metode teks mining dari masing-masing perguruan tinggi, dan dianalisa manual untuk

mencari label topik apa yang sesuai. Berikut adalah beberapa contoh hasil analisa manual terhadap dataset.

Tabel 1-15 Contoh Hasil Analisa Manual Sebagai Ground Truth Reputasi Akademik

No	Nama Perguruan Tinggi	No Dok	Nama Dokumen	Hasil Analisa
1	Institut Pertanian Bogor	1	IPB_001_01	T1, T2
		2	IPB_001_02	T1, T2
		3	IPB_001_03	T1, T2, T3
		4	IPB_05_10	T3, T4, T5
		5	IPB_05_14	T4, T5, T6, T7
2	Institut Teknologi Bandung	1	ITB_01_01	T4, T3, T5
		2	ITB_01_05	T4, T5
		3	ITB_01_22	T1, T2, T3
		4	ITB_01_09	T4, T5, T6, T7
		5	ITB_01_07	T4, T5

3.1.9 Dataset Pengujian Sistem

3.1.9.1 Data Dokumen Abstrak Akademisi Perguruan Tinggi

Dokumen abstrak paper akademisi didapatkan dari database metadata publikasi Google Scholar. Dalam Tugas Akhir ini memfokuskan dokumen abstrak paper pada akademisi yang mempunyai bidang fokus teknik. Masing-masing perguruan tinggi diambil 50 dokumen abstrak *paper* teknik berdasarkan sitasi tertinggi dari akademisi yang terdaftar atas institusi perguruan tinggi di Google Scholar.

Dalam proses teks mining, 50 buah dataset abstrak paper perguruan tinggi digabung menjadi satu dokumen utuh yang memungkinkan adanya kata yang sama berjumlah lebih dari satu (duplikasi). Satu dokumen utuh ini selanjutnya akan menjadi masukan untuk proses tek mining tahap preprocessing. Penentuan dataset dokumen abstrak paper akademisi dengan *rule* seperti berikut:

1. Pencarian daftar akademisi dalam Google Scholar dengan keyword email yang terhubung dengan institusi atau keyword hosting institusi. Misal Institut Teknologi Sepuluh Nopember dicari menggunakan keyword “@its.ac.id” atau www.its.ac.id
2. Daftar akademisi yang diambil adalah yang mempunyai tingkat sitasi tertinggi dan berfokus pada bidang teknik.
3. Masing-masing dari akademisi yang mempunyai sitasi tertinggi diambil paper yang membahas teknik.
4. Dokumen abstrak paper difokuskan pada dokumen berbahasa Inggris.

Tahap proses teks mining dalam Tugas Akhir ini akan menghasilkan reduksi terhadap kata (*term*) yang ada di dalam dokumen abstrak *paper* akademisi. Reduksi terhadap kata (*term*) berperan penting untuk menyeleksi kata (*term*) yang tidak penting. Dilakukan analisa terhadap keberadaan kata (*term*) yang penting dan tidak penting dalam dataset untuk melihat rata-rata duplikasi *term*. Rincian jumlah kata pada dataset abstrak paper akademisi (16 perguruan tinggi teratas berdasarkan *Ground Truth*) seperti pada Tabel 1-16.

Tabel 1-16 Daftar Rincian Jumlah Kata Dataset Abstrak Paper Perguruan Tinggi

No.	Perguruan Tinggi	Total Kata
1.	Institus Pertanian Bogor	10539
2.	Institut Teknologi Bandung	9287
3.	Institut Teknologi Sepuluh Nopember	9630
4.	Universitas 17 Agustus 1945	9146
5.	Universitas Airlangga	9457
6.	Universitas Bina Nusantara	7236
7.	Universitas Brawijaya	8651
8.	Universitas Diponegoro	8397
9.	Universitas Gadjah Mada	10214
10.	Universitas Indonesia	8823
11.	Universitas Islam Indonesia	7897

12.	Universitas Mataram	7527
13.	Universitas Muhammadiyah Surakarta	10330
14.	Universitas Padjadjaran	7774
15.	Universitas Pelita Harapan	7204
16.	Universitas Sebelas Maret	7807

3.1.9.2 Data Corpus Bloom Taxonomy

Corpus Bloom Taxonomy berfungsi sebagai dasar pengetahuan corpus Tingkat Kesiapterapan Teknologi (TKT). Corpus Bloom Taxonomy mempunyai 6 level tingkatan yaitu Knowledge, Comprehension, Application, Analysis, Synthesis, dan Evaluation [29]. 6 level corpus Bloom Taxonomy akan digunakan untuk pengetahuan corpus TKT yang nantinya akan dikembangkan lagi menjadi 9 level (*class*). Dilihat dari jumlah keyword yang tercakup dalam masing-masing level Bloom Taxonomy dapat dikatakan kurang banyak dalam jumlah keyword jika dibandingkan dengan jumlah kata (*term*) pada dokumen dataset abstrak *paper* akademisi. Untuk selanjutnya akan diadakan proses breakdown corpus Bloom Taxonomy menjadi corpus TKT dengan melalui berbagai tahap. Rincian jumlah dari masing-masing corpus Bloom Taxonomy seperti pada Tabel 1-17.

Tabel 1-17 Daftar Rincian Jumlah Kata Corpus Taxonomy Bloom

No.	Level Taxonomy	Total Kata
1.	Knowledge	35
2.	Comprehension	29
3.	Application	36
4.	Analysis	51
5.	Synthesis	51
6.	Evaluation	46

3.1.9.3 Data Corpus Tingkat Kesiapterapan Teknologi (TKT)

Corpus Tingkat Kesiapterapan Teknologi (TKT) merupakan corpus utama untuk proses pengenalan kata (*term*)

dalam dokumen abstrak *paper* akademisi agar masuk dalam kategori level TKT yang mana. Corpus TKT ini berjumlah 9 level, dan didapatkan dari proses breakdown 6 level corpus Bloom Taxonomy.

Metode breakdown yang digunakan dalam penentuan corpus TKT ini adalah sebagai berikut:

1. Menggabungkan 6 level corpus Bloom Taxonomy menjadi satu dokumen besar.
2. Mengurutkan kata (*term*) dari corpus Bloom Taxonomy besar berdasarkan bobot konteks kata (*term*) dari yang paling dasar hingga tingkat penerapan. Tahap ini menggunakan kemampuan manusia (*expert*) untuk membedakan dan mensorting sesuai urutan.
3. Menghapus kata yang berjumlah lebih dari satu (duplikasi).
4. Memecah urutan konten kata (*term*) tersebut menjadi 9 bagian. Pemecahan ini didasarkan pada setiap urutan level TKT akan bergantung pada TKT selanjutnya. Sebagai contoh corpus TKT 2 pasti berkorelasi dengan TKT 1 dan TKT 3. Sehingga pemecahan ini mudah untuk dilakukan. Secara tidak langsung level TKT dapat diperoleh kata-kata dari berbagai macam kata pada level Bloom Taxonomy.
5. Dokumen hasil breakdown berjumlah 9 corpus kata TKT baru tanpa mengandung kata duplikasi.

Perbandingan dari 6 level corpus Bloom Taxonomy dengan 9 level corpus TKT seperti pada Tabel 1-18.

Tabel 1-18 Perbandingan Jumlah Kata Corpus TKT

Level Bloom Taxonomy	Jumlah Kata	Level TKT	Jumlah Kata
Knowledge	35	TKT 1	31
		TKT 2	24
Comprehension	29	TKT 3	32
Application	36	TKT 4	15
		TKT 5	32
Analysis	51	TKT 6	24

Synthesis	51	TKT 7	26
		TKT 8	30
Evaluation	46	TKT 9	34
Total	256	Total	256

3.2 Tahap Perancangan

Tahap ini meliputi perancangan basis data, perancangan tampilan antarmuka (*interfaces*), perancangan alur proses penggunaan sistem antar pengguna, dan proses utama dalam proses teks mining topik modeling menggunakan metode L-LDA. Diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini.

3.2.1 Perancangan Basis Data

Perangkat lunak yang dibangun menggunakan basis data MySQL dengan bahasa syntax SQL. Terdapat 28 tabel untuk menampung hasil proses grabbing data online, hasil proses teks minning berupa skor reputasi akademik, hasil penampung skor masing-masing kriteria, serta data perguruan tinggi. Terdapat 2 buah kegunaan utama dari basis data yang digunakan dalam pembuatan perangkat lunak, yaitu sebagai penyimpan data final, dan penyimpanan data temporary update. Dengan rincian 16 tabel utama dan 12 tabel temporary sebagai penampung sementara data hasil Teknik grabbing.

3.2.1.1 Tabel Universitas

Identifikasi : universitas

Deskripsi isi : tabel untuk menyimpan data nama perguruan tinggi beserta informasi detail

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama	Variabel penampung nama	Varchar(255)	No
Domain	Variabel penampung domain	Varchar(150)	No

	webhost		
url_web	Variabel penampung alamat link webhost	Varchar(150)	No
Kota	Variabel kota	Varchar(150)	No
provinsi	Variabel provinsi	Varchar(150)	No

3.2.1.2 Tabel Reputasi Akademik

Identifikasi : reputasi_akademik

Deskripsi isi : tabel untuk menyimpan nilai hasil proses teks mining metode L-LDA sebagai representasi nilai reputasi akademik

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(255)	No
Skor_universitas	Variabel penampung skor final reputasi akademik	Float	No
Bobot_total	Variabel penampung jumlah hasil reputasi akademik	Int(2)	No
Jumlah_dokumen	Variabel penampung jumlah dokumen olah	Int(2)	No
Tanggal_akses	Variabel timestamp	Datetime	No

3.2.1.3 Tabel Akreditasi

Identifikasi : tampung_akreditasi

Deskripsi isi : tabel untuk menyimpan nilai hasil proses grabbing akreditasi perguruan tinggi

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(255)	NO
Skor_universitas	Variabel penampung skor final reputasi akademik	Float	NO
Bobot_total	Variabel penampung jumlah hasil reputasi akademik	Int(2)	No

Jumlah_dokumen	Variabel penampung jumlah dokumen olah	Int(2)	No
Tanggal_akses	Variabel waktu akses	Datetime	No

3.2.1.4 Tabel Jumlah Mahasiswa

Identifikasi : jumlah_mahasiswa

Deskripsi isi : tabel untuk menyimpan jumlah mahasiswa yang didapatkan dari data PDDIKTI

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Kode_universitas	Variabel penampung kode status diakui universitas	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(255)	No
Jumlah_tahun1_semester1	Variabel jumlah mahasiswa tahun1 pada semester1	Int(2)	No
Jumlah_tahun1_semester2	Variabel jumlah mahasiswa tahun1 pada semester2	Int(2)	No
Jumlah_tahun2_semester1	Variabel jumlah mahasiswa tahun2 pada semester1	Int(2)	No
Jumlah_tahun2_semester2	Variabel jumlah mahasiswa tahun2 pada semester2	Int(2)	No
Status_universitas	Variabel berisi keterangan status universitas	Varchar(5)	No

3.2.1.5 Tabel Jumlah Sitasi Sinta

Identifikasi : jumlah_sitasi_sinta

Deskripsi isi : tabel untuk menyimpan jumlah sitasi akademisi yang didapatkan dari database online SINTA

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No

Kode_universitas	Variabel penampung kode status diakui universitas	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(250)	No
Jumlah_tahun1_semester1	Variabel jumlah mahasiswa tahun1 pada semester1	Int(2)	No
Jumlah_tahun1_semester2	Variabel jumlah mahasiswa tahun1 pada semester2	Int(2)	No
Jumlah_tahun2_semester1	Variabel jumlah mahasiswa tahun2 pada semester1	Int(2)	No
Jumlah_tahun2_semester2	Variabel jumlah mahasiswa tahun2 pada semester2	Int(2)	No
Status_universitas	Variabel berisi keterangan status universitas	Varchar(5)	No

3.2.1.6 Tabel Jumlah Author Sinta

Identifikasi : jumlah_author_sinta

Deskripsi isi : tabel untuk menyimpan jumlah h-index dan i10-index akademisi yang didapatkan dari database online SINTA

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Id_data	Variabel tampung id data secara kolektif	Int(2)	No
Nomor	Variabel nomor akademisi dalam range satu kali iterasi	Int(2)	No
Nama_author	Variabel penampung nama akademisi	Varchar(250)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(250)	No
Jumlah_sitasi_scholar	Variabel jumlah sitasi	Int(2)	No
I10_index_scholar	Vaiabel jumlah i10-index	Int(2)	No
H_index_s	Variabel jumlah h_index	Int(2)	No

cholar			
Waktu_akses	Variabel penampung waktu akses	Tinytext	No

3.2.1.7 Tabel Precense

Identifikasi : precense

Deskripsi isi : tabel untuk menyimpan jumlah domain, subdomain dan repository webhost perguruan tinggi yang terindex Google

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
url_web	Variabel penampung link webhost	Varchar(255)	No
Domain_search	Variabel penampung link search webhost	Varchar(150)	No
Hasil_search	Variabel penampung hasil search webhost	Int(2)	No
Tanggal_akses	Variabel waktu akses	Datetime	No

3.2.1.8 Tabel Impact

Identifikasi : impact

Deskripsi isi : tabel untuk menyimpan jumlah page authority, domain authority dan backlink webhost perguruan tinggi

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
url_web	Variabel penampung link webhost	Varchar(255)	No
Domain_web	Variabel penampung link webhost tanpa “www”	Varchar(150)	No
Page_authority	Variabel penampung nilai page authority	Int(2)	No
Domain_authority	Variabel penampung nilai domain authority	Int(2)	No
Backlink_eksterna	Variabel penampung jumlah backlink	Int(2)	No

Tanggal_akses	Variabel waktu akses	Datetime	No
---------------	----------------------	----------	----

3.2.1.9 Tabel Alexa Rank

Identifikasi : alexa_rank

Deskripsi isi : tabel untuk menyimpan peringkat webhost berdasarkan alexa rank

<u>Id_field</u>	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
url_web	Variabel penampung link webhost	Varchar(255)	No
Rank_world	Variabel penampung peringkat webhost skala dunia	Int(2)	No
Rank_country	Variabel penampung peringkat webhost skala nasional	Int(2)	No
Country_name	Variabel penampung nilai domain authority	Varchar(150)	No
Tanggal_akses	Variabel waktu akses	Datetime	No

3.2.1.10 Tabel Kumpulan Skor Final

Identifikasi : kumpulan_skor_final

Deskripsi isi : tabel untuk menyimpan skor dari setiap kriteria pemeringkatan

<u>Id_field</u>	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Kode_universitas	Variabel penampung kode status diakui universitas	Float	No
Nama_universitas	Variabel nama perguruan tinggi	Varchar(255)	No
Skor_reputasi_akademik_normalisasi	Variabel penampung skor reputasi akademik yang dinormalisasi	Float	No
Skor_nilai_akreditasi_normalisasi	Variabel penampung skor nilai akreditasi yang dinormalisasi	Float	No

Skor_jumlah_mahasiswa_normalisasi	Variabel penampung skor jumlah mahasiswa yang dinormalisasi	Float	No
Skor_jumlah_sitasi_normalisasi	Variabel penampung skor jumlah sitasi yang dinormalisasi	Float	No
Skor_jumlah_tenaga_pengajar_normalisasi	Variabel penampung skor jumlah tenaga pengajar yang dinormalisasi	Float	No
Skor_kualitas_sitasi_normalisasi	Variabel penampung skor kualitas sitasi yang sudah dinormalisasi	Float	No
Skor_jumlah_phd_normalisasi	Variabel penampung skor jumlah tenaga pengajar S3 (Phd) yang sudah dinormalisasi	Float	No
Skor_jumlah_paper_normalisasi	Variabel penampung skor jumlah paper yang dinormalisasi	Float	No
Skor_jumlah_jurnal_artikel	Variabel penampung skor jumlah jurnal artikel yang dinormalisasi	Float	No
Skor_jumlah_paper_conference	Variabel penampung skor jumlah paper conference yang dinormalisasi	Float	No
Skor_persen_normalisasi	Variabel penampung skor persentase yang dinormalisasi	Float	No
Skor_impact_normalisasi	Variabel penampung skor impact yang dinormalisasi	Float	No
Skor_transparency_normalisasi	Variabel penampung skor transparency yang dinormalisasi	Float	No
Skor_excellent_normalisasi	Variabel penampung skor excellence yang dinormalisasi	Float	No
Skor_moz_analytic_normalisasi	Variabel penampung skor moz analytic yang	Float	No

malisasi	dinormalisasi		
Skor_alexarank_normalisasi	Variabel penampung skor alexa rank yang dinormalisasi	Float	No
Skor_kualitass_dosen_normalisasi	Variabel penampung skor kualitas dosen yang dinormalisasi	Float	No
Skor_kecukupan_dosen_normalisasi	Variabel penampung skor kecukupan dosen yang dinormalisasi	Float	No
Skor_kualitass_manajemen_normalisasi	Variabel penampung skor kualitas manajemen yang dinormalisasi	Float	No
Skor_kualitass_kegiatan_kemahasiswaan	Variabel penampung skor kualitas kegiatan mahasiswa yang dinormalisasi	Float	No
Sko_kualitass_kegiatan_penelitian	Variabel penampung skor kualitas kegiatan penelitian yang dinormalisasi	Float	No

3.2.1.11 Tabel Peringkat 4ICU

Identifikasi : 4icu

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemeringkatan 4ICU

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_4icu	Variabel penampung skor 4ICU	Float	No
Peringkat_4icu	Variabel penampung peringkat 4ICU	Int(2)	No

3.2.1.12 Tabel Peringkat DIKTI

Identifikasi : dikti

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemerinkkatan DIKTI Indonesia

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_uni versitas	Variabel penampung nama universitas	Varchar(150)	No
Skor_dikti	Variabel penampung skor DIKTI	Float	No
Peringkat_ dikti	Variabel penampung peringkat DIKTI	Int(2)	No

3.2.1.13 Tabel Peringkat QS General

Identifikasi : qs_general

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemerinkkatan QS rank versi general

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_uni versitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_qs	Variabel penampung skor QS General	Float	No
Peringkat_ qs	Variabel penampung peringkat QS General	Int(2)	No

3.2.1.14 Tabel Peringkat QS Asia

Identifikasi : qs_asia

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemerinkkatan QS rank versi region Asia

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_uni versitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_qs_a	Variabel penampung skor	Float	No

sia	QS Asia		
Peringkat_qs-asia	Variabel penampung peringkat QS Asia	Int(2)	No

3.2.1.15 Tabel Peringkat QS Metode L-LDA

Identifikasi : qs_llda

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemeringkatan QS rank berdasarkan metode L-LDA

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_llda	Variabel penampung skor QS L-LDA	Float	No
Peringkat_llda	Variabel penampung peringkat QS L-LDA	Int(2)	No

3.2.1.16 Tabel Peringkat Webometrics

Identifikasi : webometrics

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemeringkatan Webometrics

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_webometrics	Variabel penampung skor Webometrics	Float	No
Peringkat_webometrics	Variabel penampung peringkat Webometrics	Int(2)	No

3.2.2 Perancangan Antarmuka

Perangkat lunak yang dibangun berbasis pada aplikasi website. Antarmuka perangkat lunak dibangun menggunakan

bahasa pemrograman PHP dan HTML. Berikut adalah rincian antarmuka yang dibangun dalam perngkat lunak.

3.2.2.1 Antarmuka Home

Identifikasi : antarmuka home (index)

Deskripsi isi : antarmuka ini berfungsi sebagai halaman root utama dalam perangkat lunak website. Terdapat satu kebutuhan fungsional dalam antarmuka ini yaitu pencarian perguruan tinggi berdasarkan potongan nama. Terdapat kolom search perguruan tinggi.

3.2.2.2 Antarmuka Latar Belakang Pemingkatan

Identifikasi : antarmuka sekilas pemingkatan

Deskripsi isi : antarmuka ini berfungsi sebagai penampil informasi sederhana yang menyatakan latar belakang kegunaan dari sistem pemingkatan secara general.

3.2.2.3 Antarmuka Metode Penilaian

Identifikasi : antarmuka metode penilaian

Deskripsi isi : antarmuka ini berfungsi sebagai penampil informasi mengenai kriteria penilaian apa saja yang digunakan dalam sistem pemingkatan. Terdapat informasi secara general dan informasi secara detail dari setiap tipe pemingkatan, mulai dari pemingkatan QS General, QS Asia, Webometrics, 4ICU, pemingkatan DIKTI, dan QS menggunakan metode L-LDA. Terdapat tombol untuk melihat tipe pemingkatan yang diinginkan.

3.2.2.4 Antarmuka Peringkat Berdasarkan Region

Identifikasi : antarmuka peringkat berdasarkan region

Deskripsi isi : antarmuka ini berfungsi sebagai halaman pencarian peringkat perguruan tinggi berdasarkan region wilayah provinsi perguruan tinggi. Dengan menampilkan desain map

negara Indonesia sebagai inputan pencarian yang mudah untuk dipahami. Pencarian dilakukan dengan cara menekan salah satu bagian provinsi dari map negara Indonesia.

3.2.2.5 Antarmuka Hasil Pencarian Berdasar Region Provinsi

Identifikasi : antarmuka hasil pencarian berdasarkan region provinsi

Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil hasil pencarian perguruan tinggi berdasarkan region provinsi yang dimiliki. Ditampilkan juga peringkat dari perguruan tinggi dalam hasil pencarian dari beberapa tipe pemeringkatan seperti QS General, QS Asia, Webometrics, 4ICU, pemeringkatan DIKTI, dan QS L-LDA. Terdapat tombol untuk melihat tipe pemeringkatan yang diinginkan, dan terdapat kolom filter untuk mencari berdasarkan kota.

3.2.2.6 Antarmuka Hasil Pencarian Berdasar Pencarian Keyword Potongan Nama

Identifikasi : antarmuka hasil pencarian berdasarkan pencarian keyword potongan nama

Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil hasil pencarian perguruan tinggi berdasar keyword potongan nama. Terdapat informasi peringkat dari masing-masing perguruan tinggi yang dihasilkan. Terdapat tombol “Lihat Detail” untuk melihat detail informasi akademik perguruan tinggi.

3.2.2.7 Antarmuka Detail Informasi Akademik

Identifikasi : antarmuka detail informasi akademik

Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil informasi akademik perguruan tinggi. Terdapat beberapa informasi diantaranya mahasiswa, kecukupan dosen, status akreditasi, kualitas akademisi, produktifitas institusi, dan

kepopuleran website. Ditampilkan juga grafik peringkat dari berbagai tipe pemeringkatan.

3.2.2.8 Antarmuka Pembobotan Pemeringkatan

Identifikasi : antarmuka pembobotan pemeringkatan

Deskripsi isi : antarmuka ini berfungsi sebagai halaman masukan bobot pemeringkatan yang diinginkan. Terdapat tabel tipe pemeringkatan dan mempunyai daftar kriteria yang digunakan. Pengguna dapat memasukkan bobot pemeringkatan sesuai kebutuhan. Nilai maksimal pembobotan dari satu tipe pemeringkatan adalah 100%, jika lebih dari itu maka akan muncul pesan error bobot berlebihan. Terdapat tombol pencarian berdasarkan tipe pemeringkatan, diantaranya “Cek QS-General”, “Cek QS-Asia”, “Cek 4ICU”, “Cek Webometrics”, “Cek DIKTI”, “Cek QS L-LDA”, “Cek QS PLSa”, “Cek QS Helmholtz”, “Cek QS Adaboost”.

3.2.2.9 Antarmuka Hasil Pemeringkatan Berdasar Pembobotan Tipe Pemeringkatan yang Diinginkan

Identifikasi : antarmuka hasil pemeringkatan berdasarkan hasil pembobotan

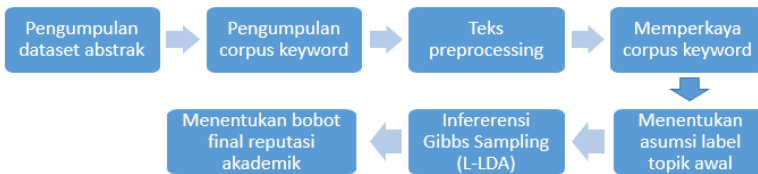
Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil hasil pembobotan perguruan tinggi sesuai dengan tipe pemeringkatan yang dikehendaki. Terdapat kolom filter untuk mencari berdasarkan kota.

3.2.3 Perancangan Kriteria Penilaian Perguruan Tinggi

Dalam menentukan peringkat perguruan tinggi menggunakan metode L-LDA, dibutuhkan beberapa kriteria penilaian lain sebagai penunjang aspek kriteria kuantitatif dan kualitatif [4]. Penjelasan masing-masing kriteria akan dijelaskan dibawah ini.

3.2.3.1 Reputasi Akademik

Merupakan kriteria utama yang bersifat kualitatif yang didapatkan dari proses teks mining dokumen abstrak perguruan tinggi menggunakan metode *Labeled Latent Dirichlet Allocation* (L-LDA) [6]. Mempunyai bobot pemeringkatan 40% dari total bobot keseluruhan. Alur proses menghasilkan kriteria reputasi akademik seperti pada Gambar 3.2-1.



Gambar 3.2-1 Alur Menentukan Reputasi Akademik

Langkah pertama dengan mengumpulkan dataset berupa abstrak paper perguruan tinggi, kemudian membangun corpus keyword yang didapatkan dari Bloom Taxonomy dan dikembangkan menjadi corpus TKT seperti pada langkah 4 pada bagan diatas. Selanjutnya melakukan tahap teks preprocessing untuk menghasilkan kata inputan yang baku dari dokumen dataset. Hasil dari tahap teks preprocessing akan digunakan sebagai penentuan asumsi label topik awal, dimana setiap dokumen harus mempunyai label sebagai acuan topiknya. Kemudian masuk pada tahap inferensi Gibbs Sampling untuk menghasilkan probabilitas terbaik dari dokumen terhadap topik. Hasil dari tahap ini berupa probabilitas akhir yang nantinya akan dihitung menggunakan pembobotan final reputasi akademik. Hasil ini kemudian disimpan dalam database sebagai nilai indicator reputasi akademik.

3.2.3.2 Akreditasi Perguruan Tinggi

Merupakan kriteria yang bersifat kuantitatif. Didapatkan dari proses grabbing data online pada website BAN-PT. Data

yang diambil adalah data nama perguruan tinggi, masa berlaku, nilai huruf akreditasi, serta status perguruan tinggi. Mempunyai bobot 10% dari total bobot keseluruhan. Alur proses menghasilkan kriteria akreditasi perguruan tinggi seperti pada Gambar 3.2-2.



Gambar 3.2-2 Alur Menentukan Skor Akreditasi

Langkah pertama dengan menyiapkan situs website yang akan dijadikan sasaran *grabbing*, dalam Tugas Akhir ini menggunakan 2 percobaan pada website BAN-PT dan website penyedia informasi akreditasi. Kemudian Menentukan teknik *grabbing* apa dan library yang akan digunakan, dalam Tugas Akhir ini menggunakan Bahasa pemrograman PHP, library HTML DOM Parser, dan beberapa fungsi bawahan dari PHP sendiri. Kemudian melakukan *grabbing* dengan ketentuan harus mempunyai bandwidth yang cukup. Karena proses *grabbing* membutuhkan waktu yang cukup lama melihat konten yang ingin diambil cukup besar dan banyak. Dalam proses *grabbing* data yang masuk akan langsung disimpan dalam table temporary pada database, sebagai penampung sementara karena tujuan utama disini adalah mengambil sebisa mungkin. Dalam database table temporary ini kemudian akan dianalisa dan dilakukan normalisasi hasil untuk menghasilkan skor akreditasi yang paling akhir. Skor akreditasi ini kemudian akan disimpan lagi dalam database dengan table yang berbeda sebagai implementasi skor final akreditasi perguruan tinggi.

3.2.3.3 Jumlah Rasio Mahasiswa dan Tenaga Pengajar Aktif

Merupakan kriteria yang bersifat kuantitatif. Didapatkan dari data rekap database online PDDIKTI. Data yang diambil merupakan rekap yang berisi jumlah mahasiswa, nama perguruan

tinggi serta jumlah tenaga pengajar keseluruhan. Mempunyai bobot 20% dari total bobot keseluruhan. Alur proses menghasilkan kriteria rasio mahasiswa dan tenaga pengajar aktif seperti pada Gambar 3.2-3.



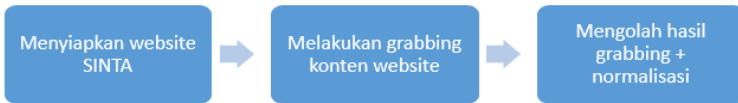
Gambar 3.2-3 Alur Menentukan Skor Rasio Mahasiswa dan Tenaga Penagajar Aktif

Langkah pertama dengan mengumpulkan data rakap jumlah mahasiswa dan rekap jumlah tenaga pengajar dari seluruh perguruan tinggi di Indonesia. Data ini disediakan oleh PDDIKTI sebagai pusat databasenya. Untuk mempermudah proses analisa, dalam Tugas Akhir ini menggunakan data rekap berformat .xlsx kemudian akan dibaca menggunakan library excel dalam Bahasa pemrograman python. Karena file excel berupa row dan column maka akan diambil beberapa variable penting sebagai acuan penilaian. Variable jumlah mahasiswa semester 1 tahun 1, mahasiswa semester 2 tahun 1, mahasiswa semester 1 tahun 2, dan semester 2 tahun 2 menjadi acuan penilaian mahasiswa. Serta data jumlah tenaga pengajar secara keseluruhan dari setiap perguruan tinggi akan menjadi acuan penilaian jumlah tenaga pengajar. Nilai dari variable ini akan dinormalisasi terlebih dahulu dengan membagi dengan nilai tertinggi. Kemudian akan dirasioakan dengan jumlah tenaga pengajar, kemudian disimpan dalam database sebagai implementasi skor final rasio mahasiswa dan tenaga pengajar aktif.

3.2.3.4 Jumlah Sitasi Akademisi Penelitian

Merupakan kriteria yang bersifat kuantitatif. Didapatkan dari proses grabbing data online *Science and Technology Index* (SINTA). Data yang digunakan adalah nama perguruan tinggi, serta jumlah sitasi dan author akademisi yang ter-*index* SINTA.

Mempunyai bobot 20% dari total bobot keseluruhan. Alur proses menghasilkan kriteria jumlah sitasi akademisi penelitian seperti pada Gambar 3.2-4.



Gambar 3.2-4 Alur Menentukan Skor Jumlah Sitasi Akademisi Penelitian

Langkah pertama dengan menyiapkan website sasaran grabbing yaitu situs website *Science and Technology Index* (SINTA) dari Kemenristek Dikti. Dalam situs ini terdapat beberapa variable penting sebagai acuan penilaian indikator. Setiap variable akan dilakukan teknik *grabbing* dan sama seperti pada proses menentukan indikator akreditasi perguruan tinggi. Dalam proses ini juga membutuhkan database table temporary sebagai penampung sementara. Variable yang digunakan adalah jumlah sitasi dari google scholar dan scopus. Kemudian dari table temporary ini akan dianalisa dan dinormalisasi dan dicari rata-ratanya sebagai skor final jumlah sitasi akademisi penelitian dan disimpan lagi dalam table final.

3.2.3.5 Jumlah Tenaga Pengajar (Dosen)

Merupakan kriteria yang bersifat kuantitatif yang didapatkan dari data rekap database online PDDIKTI. Data yang diambil merupakan jumlah tenaga pengajar dengan berbagai status ketenaga kerjaan, seperti pegawai tetap maupun tidak tetap. Mempunyai bobot 5% dari total bobot keseluruhan. Alur proses menghasilkan kriteria jumlah tenaga pengajar (dosen) seperti pada Gambar 3.2-3 diatas.

Prinsip kerja Menentukan indikator ini sama seperti indikator jumlah rasio mahasiswa dan tenaga pengajar atif. Hanya saja disini ada beberapa data rekap yang digunakan sebagai pendukung yaitu rekap jumlah tenaga pengajar yang bersifat tetap

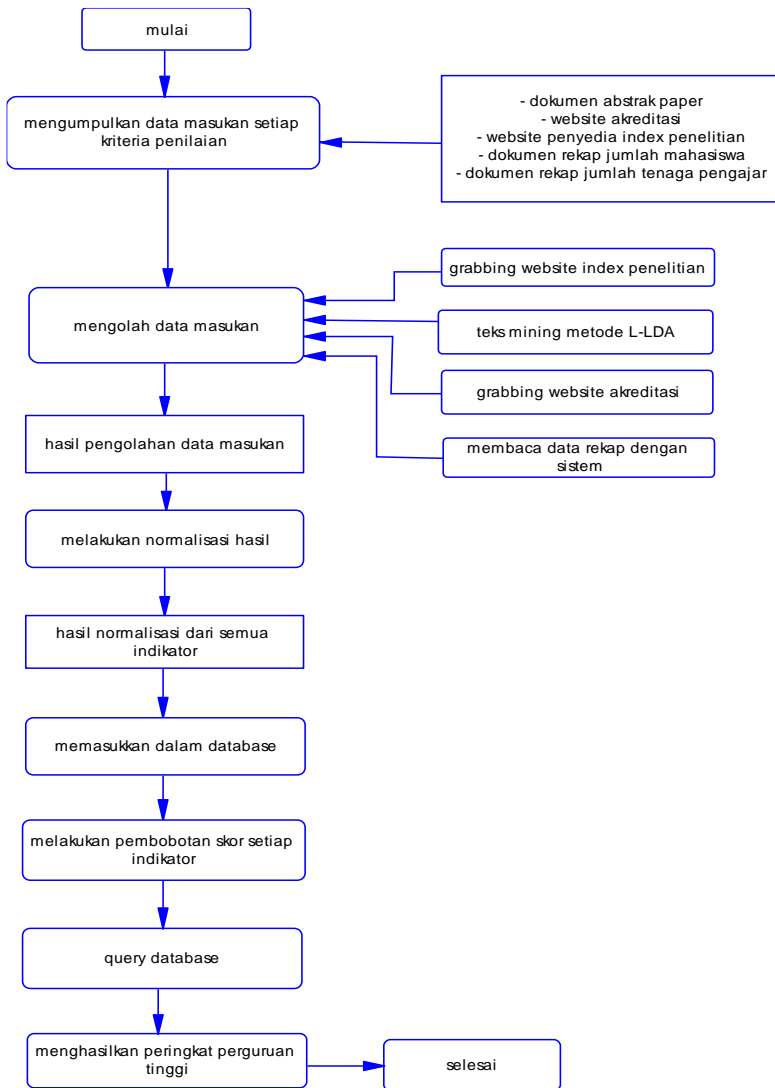
dan tidak tetap (honorer). Data ini akan dibaca menggunakan library excel pada Bahasa pemrograman PHP kemudian akan dianalisa dan dinormalisasi untuk menghasilkan jangkauan nilai yang rata. Terdapat variable tanga pengajar tetap dan tidak tetap yang nilai masing-masing telah dinormalisasi, kemudian dijumlahkan dan dirata-rata. Nilai rata-rata ini akan menjadi skor indikator jumlah tenaga pengajar (dosen).

3.2.3.6 Kualitas Sitasi

Merupakan kriteria yang bersifat kuantitatif yang didapatkan dari proses grabbing data online *Science and Technology Index* (SINTA). Data yang diambil merupakan nilai h-index dan i10-index Google Scholar dari akademisi perguruan tinggi. Mempunyai bobot 5% dari total bobot keseluruhan. Alur proses menghasilkan kriteria kualitas sitasi seperti pada Gambar 3.2-4 diatas.

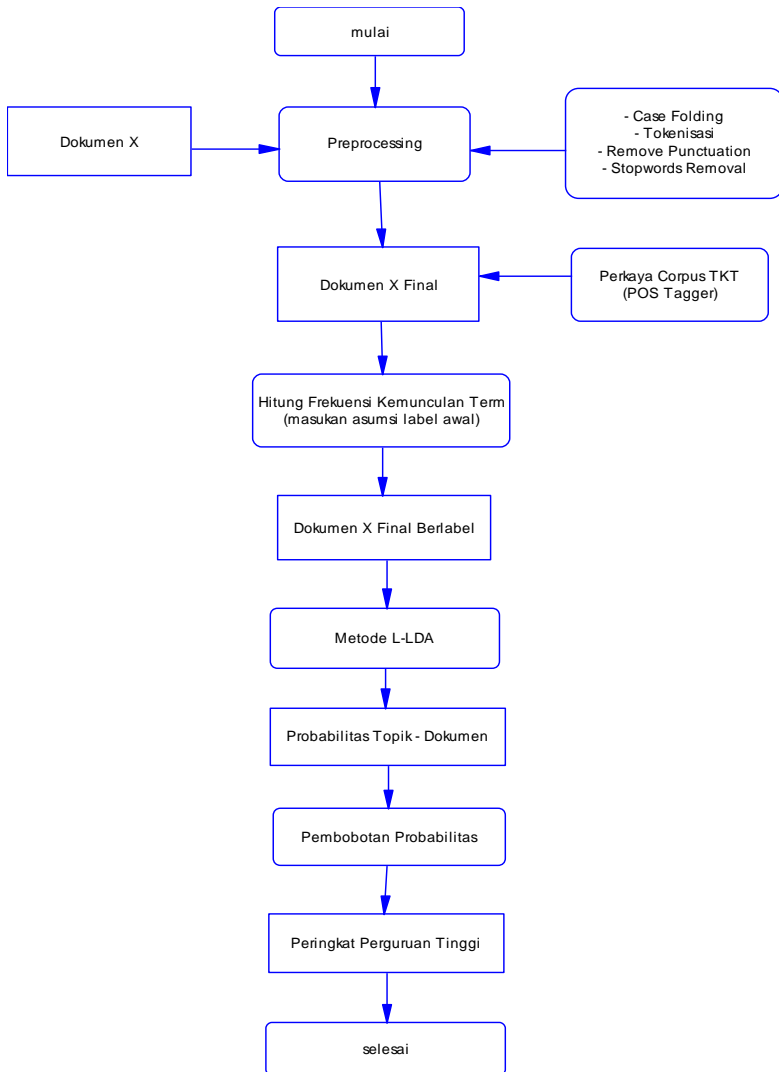
Prinsip kerja dalam Menentukan indikator ini sama seperti pada indikator jumlah sitasi akademisi penelitian. Hanya saja disini yang diambil adalah variable jumlah h-index dan i10-index dari google scholar. Kedua variable ini akan dilakukan normalisasi nilai dan dicari rata-ratanya. Dan kemudian disimpan dalam table database baru sebagai implementasi skor indikator kualitas sitasi.

3.2.5 Proses Kerja Sistem Secara Keseluruhan (Flowchart Sistem)



Gambar 3.2-6 Flowchart Sistem Keseluruhan

3.2.6 Proses Teks Mining Metode L-LDA (Flowchart Metode L-LDA)



Gambar 3.2-7 Flowchart Metode L-LDA

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI SISTEM

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi lingkungan implementasi, syntax basis data, dan kode program dalam perangkat lunak. Selain itu implementasi dari setiap proses, parameter masukan, keluaran, dan beberapa keterangan pendukung yang berhubungan dengan perangkat lunak.

4.1 Lingkungan Implementasi

Lingkungan implementasi perangkat lunak yang digunakan untuk mengembangkan Tugas Akhir ini sebagai berikut:

1. 1 Laptop untuk developer dan running perangkat lunak server menggunakan Intel® Core™ i7-5500U CPU @2.40GHz (4CPU) , RAM 12GB dengan Sistem Operasi Windows 8.1 Profesional x64 (64-bit).
2. 1 Laptop untuk pengguna End User menggunakan Intel® Core™ i7-5500U CPU @2.40GHz (4CPU) , RAM 12GB dengan Sistem Operasi Windows 8.1 Profesional x64 (64-bit).
3. Database yang digunakan pada localhost adalah MySQL database dengan IDE MySQLyog Community v11.52 (64-bit) sebagai uji coba localhost.
4. Database yang digunakan pada server online adalah MySQL.
5. Webhosting menggunakan server hosting dari namecheap.com
6. Mozilla Firefox 53.0.3 (32-bit) sebagai antarmuka untuk pengujian perangkat lunak bagi pengguna End User.
7. SublimeText 2 v2.0.2 sebagai IDE text editor dari programmer developer.

8. XAMPP v3.2.1 sebagai webserver sebagai server uji coba localhost dari programmer developer.
9. Python v2.7.12 dan Anaconda 4.2.0 (64-bit) sebagai IDE dan bahasa pemrograman utama untuk pengolahan teks minning.
10. StarUML sebagai tools pembuatan diagram Use Case perangkat lunak.
11. Power Designer sebagai tools pembuatan diagram activity CDM dan PDM perangkat lunak.

4.2 Implementasi Syntax Basis Data

4.2.1 Create Database “database_qs_rank”

```
CREATE DATABASE /*!32312 IF NOT
EXISTS*/`database_qs_2` /*!40100 DEFAULT CHARACTER SET
latin1 */;

USE `database_qs_2`;
```

Kode Sumber 4.2-1 Kode Sumber Create Database

4.2.2 Create Table “universitas”

```
CREATE TABLE `universitas` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama` varchar(255) NOT NULL,
  `domain` varchar(150) NOT NULL,
  `url_web` varchar(150) NOT NULL,
  `kota` varchar(150) NOT NULL,
  `provinsi` varchar(150) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=410 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-2 Kode Sumber Create Table universitas

4.2.3 Create Table “reputasi_akademik”

```
CREATE TABLE `reputasi_akademik` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_universitas` float NOT NULL,
  `bobot_total` int(2) NOT NULL,
  `jumlah_dokumen` int(2) NOT NULL,
  `tanggal_akses` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=484 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-3 Create Tabel reputasi_akademik

4.2.4 Create Tabel “akreditasi”

```
CREATE TABLE `tampung_akreditasi` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nomor` int(3) NOT NULL,
  `nama_universitas` varchar(255) NOT NULL,
  `peringkat` varchar(150) NOT NULL,
  `masa_berlaku` varchar(150) NOT NULL,
  `status` varchar(150) NOT NULL,
  `skor` int(2) NOT NULL,
  `tmp_nama_universitas` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=13095
DEFAULT CHARSET=latin1;
```

Kode Sumber 4.2-4 Create Tabel akreditasi

4.2.5 Create Tabel “jumlah_mahasiswa”

```
CREATE TABLE `jumlah_mahasiswa` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `kode_universitas` int(4) NOT NULL,
  `nama_universitas` varchar(255) NOT NULL,
  `jumlah_tahun1_semester1` int(2) NOT NULL,
```

```

`jumlah_tahun1_semester2` int(2) NOT NULL,
`jumlah_tahun2_semester1` int(2) NOT NULL,
`jumlah_tahun2_semester2` int(2) NOT NULL,
`status_universitas` varchar(50) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=42783
DEFAULT CHARSET=latin1;

```

Kode Sumber 4.2-5 Create Table jumlah_mahasiswa

4.2.6 Create Tabel “jumlah_tenaga_pengajar_jabatan_akademik”

```

CREATE TABLE
`jumlah_tenaga_pengajar_jabatan_akademik` (
  `id` int(2) NOT NULL AUTO_INCREMENT,
  `kode_universitas` int(2) NOT NULL,
  `nama_universitas` varchar(100) NOT NULL,
  `asisten_ahli_10000` int(2) NOT NULL,
  `asisten_ahli_15000` int(2) NOT NULL,
  `lektor_20000` int(2) NOT NULL,
  `lektor_30000` int(2) NOT NULL,
  `lektor_kepala_40000` int(2) NOT NULL,
  `lektor_kepala_55000` int(2) NOT NULL,
  `lektor_kepala_70000` int(2) NOT NULL,
  `profesor_85000` int(2) NOT NULL,
  `profesor_105000` int(2) NOT NULL,
  `tanpa_jabatan` int(2) NOT NULL,
  `status_universitas` tinytext NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=16840 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-6 Create Table jumlah_tenaga_pengajar_jabatan_akademik

4.2.7 Create Tabel “jumlah_tenaga_pengajar_tetap”

```

CREATE TABLE `jumlah_tenaga_pengajar_tetap` (
  `id` int(2) NOT NULL AUTO_INCREMENT,
  `kode_universitas` int(2) NOT NULL,
  `nama_universitas` varchar(100) NOT NULL,
  `D1` int(2) NOT NULL,
  `D2` int(2) NOT NULL,
  `D3` int(2) NOT NULL,
  `D4` int(2) NOT NULL,
  `S1` int(2) NOT NULL,
  `S2` int(2) NOT NULL,
  `S3` int(2) NOT NULL,
  `non_formal` int(2) NOT NULL,
  `informal` int(2) NOT NULL,
  `lainnya` int(2) DEFAULT NULL,
  `SP1` int(2) NOT NULL,
  `SP2` int(2) NOT NULL,
  `profesi` int(2) NOT NULL,
  `tanpa_jenjang` int(2) NOT NULL,
  `status_universitas` tinytext NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=59738 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-7 Create Table jumlah_tenaga_pengajar_tetap

4.2.8 Create Tabel “jumlah_tenaga_pengajar_tidak_tetap”

```

CREATE TABLE `jumlah_tenaga_pengajar_tidak_tetap`
(
  `id` int(2) NOT NULL AUTO_INCREMENT,
  `kode_universitas` int(2) NOT NULL,
  `nama_universitas` varchar(100) NOT NULL,
  `D1` int(2) NOT NULL,
  `D2` int(2) NOT NULL,
  `D3` int(2) NOT NULL,
  `D4` int(2) NOT NULL,
  `S1` int(2) NOT NULL,

```

```

`S2` int(2) NOT NULL,
`S3` int(2) NOT NULL,
`non_formal` int(2) NOT NULL,
`informal` int(2) NOT NULL,
`lainnya` int(2) DEFAULT NULL,
`SP1` int(2) NOT NULL,
`SP2` int(2) NOT NULL,
`profesi` int(2) NOT NULL,
`tanpa_jenjang` int(2) NOT NULL,
`status_universitas` tinytext NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8885 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-8 Create Table jumlah_tenaga_pengajar_tidak_tetap

4.2.9 Create Tabel “jumlah_sitasi_sinta”

```

CREATE TABLE `jumlah_sitasi_sinta` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `jumlah_author_scholar` int(2) NOT NULL,
  `jumlah_paper_scholar` int(2) NOT NULL,
  `jumlah_sitasi_scholar` int(2) NOT NULL,
  `jumlah_jurnal_artikel` int(2) NOT NULL,
  `jumlah_book_chapter` int(2) NOT NULL,
  `jumlah_paper_conference` int(2) NOT NULL,
  `jumlah_artikel_scopus` int(2) NOT NULL,
  `jumlah_non_artikel_scopus` int(2) NOT NULL,
  `jumlah_sitasi_scopus` int(2) NOT NULL,
  `sinta_score` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4006 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-9 Create Table jumlah_sitasi_sinta

4.2.10 Create Tabel “jumlah_author_sinta”

```

CREATE TABLE `jumlah_author_sinta` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `id_data` int(2) NOT NULL,
  `nomor` int(2) NOT NULL,
  `nama_author` varchar(150) NOT NULL,
  `nama_universitas` varchar(150) NOT NULL,
  `jumlah_sitasi_scholar` int(2) NOT NULL,
  `i10_index_scholar` int(2) NOT NULL,
  `h_index_scholar` int(2) NOT NULL,
  `waktu_akses` tinytext NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2189 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-10 Cretae Table jumlah_author_sinta

4.2.11 Create Tabel “precense”

```

CREATE TABLE `precense` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `url_web` varchar(255) NOT NULL,
  `domain_search` varchar(150) NOT NULL,
  `hasil_search` int(2) NOT NULL,
  `tanggal_akses` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=526 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-11 Create Table precense

4.2.12 Create Tabel “impact”

```
CREATE TABLE `impact` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `url_web` varchar(255) NOT NULL,
  `domain_web` varchar(150) NOT NULL,
  `page_authority` int(2) NOT NULL,
  `domain_authority` int(2) NOT NULL,
  `backlink_eksternal` int(2) NOT NULL,
  `tanggal_akses` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1960 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-12 Create Table impact

4.2.13 Create Tabel “alexa_rank”

```
CREATE TABLE `alexa_rank` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `url_web` varchar(255) NOT NULL,
  `rank_world` int(2) NOT NULL,
  `rank_country` int(2) NOT NULL,
  `country_name` varchar(150) NOT NULL,
  `tanggal_akses` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=444 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-13 Create Tabel alexa_rank

4.2.14 Create Tabel “kumpulan_skor_final”

```

CREATE TABLE `kumpulan_skor_final` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `kode_universitas` int(3) NOT NULL,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_reputasi_akademik_normalisasi` float NOT
NULL,
  `skor_nilai_akreditasi_normalisasi` float NOT
NULL,
  `skor_jumlah_mahasiswa_normalisasi` float NOT
NULL,
  `skor_jumlah_sitasi_normalisasi` float NOT NULL,
  `skor_jumlah_tenaga_pengajar_normalisasi` float
NOT NULL,
  `skor_kualitas_sitasi_normalisasi` float NOT
NULL,
  `skor_jumlah_phd_normalisasi` float NOT NULL,
  `skor_jumlah_paper_normalisasi` float NOT NULL,
  `skor_jumlah_jurnal_artikel` float NOT NULL,
  `skor_jumlah_paper_conference` float NOT NULL,
  `skor_precense_normalisasi` float NOT NULL,
  `skor_impact_normalisasi` float NOT NULL,
  `skor_transparency_normalisasi` float NOT NULL,
  `skor_excellence_normalisasi` float NOT NULL,
  `skor_moz_analytic_normalisasi` float NOT NULL,
  `skor_alex_rank_normalisasi` float NOT NULL,
  `skor_kualitas_dosen_normalisasi` float NOT
NULL,
  `skor_kecukupan_dosen_normalisasi` float NOT
NULL,
  `skor_kualitas_manajemen_normalisasi` float NOT
NULL,
  `skor_kualitas_kegiatan_kemahasiswaan_normalisasi`
  float NOT NULL,
  `skor_kualitas_kegiatan_penelitian_normalisasi`
float NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=11356 DEFAULT
CHARSET=latin1;

```

Kode Sumber 4.2-14 Create Table kumpulan_skor_final

4.2.15 Create Tabel “4icu”

```
CREATE TABLE `4icu` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_4icu` float NOT NULL,
  `peringkat_4icu` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2128 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-16 Create Table 4icu

4.2.16 Create Tabel “dikti”

```
CREATE TABLE `dikti` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_dikti` float NOT NULL,
  `peringkat_dikti` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1773 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-17 Create Table dikti

4.2.17 Create Tabel “qs_general”

```
CREATE TABLE `qs_general` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_qs` float NOT NULL,
  `peringkat_qs` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2838 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-18 Create Table qs_general

4.2.18 Create Tabel “qs_asia”

```
CREATE TABLE `qs_asia` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_qs_asia` float NOT NULL,
  `peringkat_qs_asia` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2838 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-19 Create Table qs_asia

4.2.19 Create Tabel “qs_llda”

```
CREATE TABLE `qs_llda` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_qs_llda` float NOT NULL,
  `peringkat_qs_llda` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3193 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-20 Create Table qs_llda

4.2.20 Create Tabel “webometrics”

```
CREATE TABLE `webometrics` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_webometrics` float NOT NULL,
  `peringkat_webometrics` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2838 DEFAULT
CHARSET=latin1;
```

Kode Sumber 4.2-21 Create Table webometrics

4.3 Implementasi Pseudocode Proses Teks Mining (Metode L-LDA)

4.3.1 Teks Preprocessing

```

Start
  Read dataset
  List_dataset = [], List_dataset_new = []
  List_dataset_update = []
  Declare remove_punctuation_character

  If dataset contain "\n":
    Remove "\n" from dataset
    Insert dataset to list_dataset
  End if
  Insert list_dataset.split() to list_dataset_new

  For i in list_dataset_new.length:
    If list_dataset[i] contain remove_punctuation_character:
      Remove content list_dataset[i]
      Insert list_dataset[i] to list_dataset_update
    End if
  For i in list_dataset_update.length:
    If list_dataset_update.length[i] in stopwords_list:
      Remove list_dataset_update.length[i]
      Update list_dataset_update
      List_counter = []
      Counter = 0
    End if
  For i in list_dataset_update.length:
    If list_dataset_update[i] in list_counter:
      Counter[list_dataset_update[i]] += 1
    Else:
      Counter[list_dataset_update[i]] = 1
    End if
  Print list_dataset_update
End

```

Kode Sumber 4.3-1 Pseudocode Teks Preprocessing

4.3.2 Memperkaya Corpus Keyword TKT dengan Teknik Sinonym Word (WordNet) dalam Level 1, 2, 3, dan all

```

Start
  Read corpus
  Synonym = []
  List_update_nama_corpus = []
  List_update_value_corpus = []
  List_tambahan_corpus_tkt = []

  For i in corpus.length:
    Synonym[i] = wordnet.synsets(corpus[i])

    For j in synonym[i].length:

      If corpus[i] !=synonym[j]:
        Word_1 = wordnet.synsets(corpus[i])
        Word_2 = wordnet.synsets(synonym[j])
        Probabilitas = word_1.wup_similarity(word_2)

        Insert word_2 to list_update_nama_corpus
        Insert probabilitas to list_update_value_corpus

    Probabilitas.sort(ascending)
    Get probabilitas[0]
    Index_1 = Get index probabilitas[0]
    Get words in list_update_nama_corpus with index_1

    Insert words into list_tambahan_corpus_tkt

  Enf for
End

```

Kode Sumber 4.3-2 Psudocode Memperkaya Corpus Keyword TKT Dengan Sinonym Word

4.3.3 Memperkaya Corpus Keyword TKT dengan Teknik POS Tagging

```

Start
  Read list_dataset_update
  Read corpus
  list_pos_tagger = []

  For i in list_dataset_update.length:

    If list_dataset_update[i] in corpus:
      Word_postagger[i] = List_dataset_update[i]. pos_tag

      If word_postagger[i] == "Noun":
        Insert word_postagger[i-1] in list_pos_tagger
        Insert word_postagger[i+1] in list_pos_tagger

      Else if word_postagger[i] == "Verb":
        Insert word_postagger[i-1] in list_pos_tagger
        Insert word_postagger[i+1] in list_pos_tagger
      End if
    End if
  Enf for
End

```

Kode Sumber 4.3-3 Pseudocode Memperkaya Corpus Keyword TKT dengan POS Tagging

4.3.4 Update Konten Kata Corpus Keyword TKT

```

Start
  Read corpus1 until corpus9
  Read list_postagger
  List_corpus = []
  List_corpus_update1 = []
  List_tmp_value_word_similarity = []

  For i in list_postagger.length:
    If list_postagger[i] in corpus1:

```

```
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus2:
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus3:
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus4:
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus5:
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus6:
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus7:
word_similarity[i] = list_postagger.word_similarity
insert word_similarity[i] = list_tmp_value_word_
similarity

Else if list_postagger[i] in corpus8:
word_similarity[i] = list_postagger.word_similarity
```

```

        insert word_similarity[i] = list_tmp_value_word_
        similarity

    Else if list_postagger[i] in corpus9:
        word_similarity[i] = list_postagger.word_similarity
        insert word_similarity[i] = list_tmp_value_word_
        similarity
    End if

    Max_value = list_postagger[i].max()

    If max_value != 0:
        Index = i
        Word = List_tmp_value_word_similarity[i]
        Insert list_tmp_value_word_similarity[i] to
        list_corpus_update1
    End if
End for

Insert list_corpus to list_corpus_update
End

```

**Kode Sumber 4.3-4 Pseudocode Update Konten Kata Corpus Keyword
TKT**

4.3.5 Menentukan Asumsi Label Topik Awal

```

Start
  Read list_corpus_update
  Read list_dataset_update
  List_counter_word = []
  Counter1, counter2, counter3, counter4, counter5, counter6,
  counter7, counter8, counter9 = 0

  For i in list_dataset_update.length:
    If list_dataset_update[i] in list_corpus_update1:
      Counter1 += 1
    Else if list_dataset_update[i] in list_corpus_update2:
      Counter2 += 1
    Else if list_dataset_update[i] in list_corpus_update3:
      Counter3 += 1
    Else if list_dataset_update[i] in list_corpus_update4:
      Counter4 += 1
    Else if list_dataset_update[i] in list_corpus_update5:
      Counter5 += 1
    Else if list_dataset_update[i] in list_corpus_update6:
      Counter6 += 1
    Else if list_dataset_update[i] in list_corpus_update7:
      Counter7 += 1
    Else if list_dataset_update[i] in list_corpus_update8:
      Counter8 += 1
    Else if list_dataset_update[i] in list_corpus_update9:
      Counter9 += 1
    End if
  Enf for
  Insert counter1, counter2, counter3, counter4, counter5,
  counter6, counter7, counter8, counter9 to list_counter_word

  Sort_counterword = List_counter_word.sort(ascending)
  Label = get 3 value from sort_counterword
  Print label, list_dataset
End

```

Kode Sumber 4.3-5 Menentukan Asumsi Label Topik Awal

4.3.6 Pembobotan Skor Penilaian

Start

Level_tkt = []

Probs_tkt = []

Konten1 = konten_probabilitas1

Konten2 = konten_probabilitas2

Konten3 = konten_probabilitas3

Level_tkt_1, probs_tkt_1 = konten1.split(" ")

Level_tkt_2, probs_tkt_2 = konten2.split(" ")

Level_tkt_3, probs_tkt_3 = konten3.split(" ")

Insert level_tkt_1, level_tkt_2, level_tkt_3 to level_tkt

Insert probs_tkt_1, probs_tkt_2, probs_tkt_3 to probs_tkt

For i in level_tkt.length:

If level_tkt == "T1":

Bobot = 10

*Bobot_total = bobot * probs_tkt[i]*

Else if level_tkt == "T2":

Bobot = 20

*Bobot_total = bobot * probs_tkt[i]*

Else if level_tkt == "T3":

Bobot = 30

*Bobot_total = bobot * probs_tkt[i]*

Else if level_tkt == "T4":

Bobot = 40

*Bobot_total = bobot * probs_tkt[i]*

Else if level_tkt == "T5":

Bobot = 50

*Bobot_total = bobot * probs_tkt[i]*

Else if level_tkt == "T6":

Bobot = 60

*Bobot_total = bobot * probs_tkt[i]*


```
Else if level_tkt == "T7":  
    Bobot = 70  
    Bobot_total = bobot * probs_tkt[i]  
  
Else if level_tkt == "T8":  
    Bobot = 80  
    Bobot_total = bobot * probs_tkt[i]  
  
Else if level_tkt == "T9":  
    Bobot = 90  
    Bobot_total = bobot * probs_tkt[i]  
  
End if  
End for  
  
Print bobot_total  
End
```

Kode Sumber 4.3-6 Pembobotan Skor Penilaian

4.4 Implementasi Antarmuka Pengguna

4.4.1 Antarmuka Home (index)

Antarmuka ini sebagai halaman root dari sistem pemeringkatan yang muncul pertama kali ketika pengguna membuka domain sistem pada web browser.



Gambar 4.4-1 Antarmuka Home

4.4.2 Antarmuka Mencari Perguruan Tinggi Berdasarkan Potongan Nama

Pada antarmuka ini pengguna dapat mencari perguruan tinggi berdasarkan keyword potongan nama perguruan tinggi yang diinputkan. Halaman ini juga menjadi halaman home dalam sistem perangkat lunak.



Gambar 4.4-2 Antarmuka Pencarian Nama

4.4.3 Antarmuka Hasil Pencarian Perguruan Tinggi Berdasarkan Keyword Potongan Nama

Pada antarmuka ini pengguna End User dapat melihat hasil pencarian perguruan tinggi berdasarkan keyword potongan nama.

Hasil Pencarian

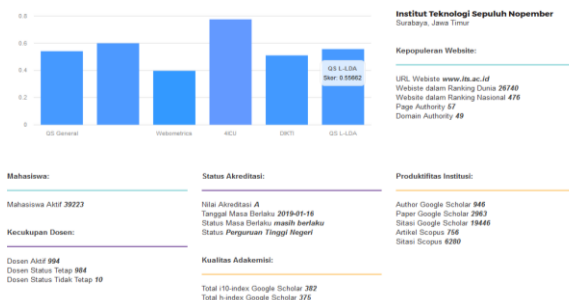
No.	Nama Perguruan Tinggi	Kota	Provinsi	QS General	QS Asia	Webometrics	AICU	Dikti	QS L-LDA	Lainnya
1	Institut Teknologi Sepuluh Nopember	Surabaya	Jawa Timur	Peringkat 8	Peringkat 7	Peringkat 8	Peringkat 12	Peringkat 3	Peringkat 11	Lihat Detail

Gambar 4.4-3 Antarmuka Hasil Pencarian

4.4.4 Antarmuka Melihat Detail Informasi Perguruan Tinggi

Pada antarmuka ini pengguna dapat melihat informasi secara detail, diantaranya grafik peringkat perguruan tinggi berdasar tipe pemeringkatan, informasi mahasiswa, kecukupan dosen, status akreditasi, produktifitas akademisi, produktifitas institusi dan kepopuleran website.

Detail Informasi



Gambar 4.4-4 Antarmuka Detail Informasi

4.4.5 Antarmuka Melihat Latar Belakang Sistem Pemingkatan

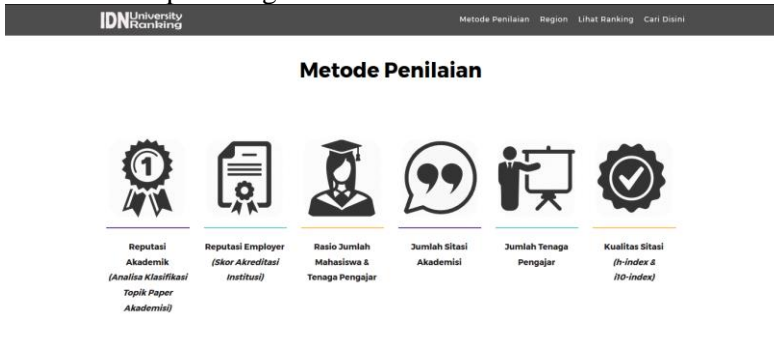
Pada antarmuka ini pengguna dapat melihat latar belakang dari sistem pemeringkatan.



Gambar 4.4-5 Antarmuka Informasi Latar Belakang

4.4.6 Antarmuka Melihat Informasi Metode Penilaian

serta informasi mengenai kriteria metode yang digunakan dalam sistem pemeringkatan ini.



Gambar 4.4-6 Antarmuka Informasi Metode Penilaian

Metode Penilaian


QS General QS Asia Webometrics 4CU Peningkatan DKTI QS L-LDA

Reputasi Akademik
(Analisa Kualifikasi Topik Paper Akademisi)

Kriteria penilaian yang berfokus pada pemodelan topik dari kumpulan dokumen paper akademisi Perguruan Tinggi yang terindeks oleh Google Scholar. Dokumen paper yang digunakan berjumlah 10 dokumen dengan kualitas terbaik, yaitu berasal dari user akademisi yang mempunyai rating sitasi tertinggi. Kualifikasi Topik dibagi menjadi 9 topik utama mengikut kelas topik dari TRK (Tingkat Keunggulan Teknologi Perguruan Tinggi). Mempunyai bobot penilaian 40% dari nilai total.

Reputasi Employer
(Skor Akreditasi Institusi)

Kriteria penilaian yang berfokus pada hasil skor akreditasi institusi Perguruan Tinggi yang diselenggarakan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Non atribut penilaian yang diambil adalah skor huruf akreditasi yang kemudian diconvert.



Gambar 4.4-7 Antarmuka Metode Penilaian QS General

SN&W Metode Penilaian Region Lihat Ranking Cari Diisi

Mempunyai bobot penilaian 20% dari nilai total.

Rasio Jumlah Mahasiswa dan Tenaga Pengajar

Kriteria penilaian yang berfokus pada hasil rasio perbandingan antara jumlah mahasiswa aktif dengan jumlah tenaga pengajar dalam Perguruan Tinggi. Data jumlah mahasiswa dan jumlah tenaga pengajar sudah disediakan oleh Kementerian Riset Teknologi dan Pendidikan Tinggi (Kemendikbud). Tujuan kriteria ini adalah untuk melihat kecukupan dan keseimbangan tenaga pengajar untuk menghimpun mahasiswa. Mempunyai bobot penilaian 45% dari nilai total.

Jumlah Sitasi Akademisi
(Database Scopus)

Kriteria penilaian yang berfokus pada kualitas akademisi Perguruan Tinggi yang dilihat dari jumlah sitasi yang terindeks oleh database paper dan artikel Scopus. Penggunaan database paper dan artikel dari Scopus dikarenakan file yang dimiliki mempunyai kualitas lebih baik daripada database lainnya.




Gambar 4.4-8 Antarmuka Metode Penilaian QS Asia

Metode Penilaian

QS General QS Asia Webometrics 4CU Peningkatan DKTI QS L-LDA

Precense
(Jumlah Domain, Subdomain, & Repository)

Kriteria penilaian yang berfokus pada seberapa banyak jumlah webhost (domain, subdomain, repository) yang dimiliki oleh Perguruan Tinggi yang terindeks oleh Google. Semakin banyak webhost yang dimiliki, maka semakin banyak kesempatan akan dikenal masyarakat melalui penyebaran dan pengisian di internet, pengembangan dan penguasaan Perguruan Tinggi menggunakan internet akan jauh lebih cepat. Selain itu jumlah webhost tersebut harus sejalan dengan konten yang diberikan, yang mendukung pengembangan dan penyebaran ilmu pengetahuan. Mempunyai bobot penilaian 40% dari nilai total.

Impact
(Jumlah Backlink Webhost)

Kriteria penilaian yang berfokus pada jumlah Eksternal Backlink webhost yang terindeks oleh




Gambar 4.4-9 Antarmuka Metode Penilaian Webometrics

4.4.7 Antarmuka Pencarian Peringkat Perguruan Tinggi Berdasarkan Region

Pada antarmuka ini pengguna dapat melihat peringkat perguruan tinggi berdasarkan region wilayah provinsi. Terdapat map wilayah Indonesia sebagai masukan pemilihan region wilayah yang ditentukan.



Gambar 4.4-10 Antarmuka Pencarian Berdasar Region

4.4.8 Antarmuka Hasil Pencarian Perguruan Tinggi Berdasar Region Wilayah Perguruan Tinggi

Pada antarmuka ini pengguna dapat melihat hasil peringkat dalam region wilayah perguruan tinggi.

Peringkat Perguruan Tinggi
(Region: Jawa Timur)

QS General QS Asia Webometrics 4ICU Pemeringkatan DKTI QS LLDA

Masukkan Nama Kota Disini...

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Universitas Brawijaya	0.561164	Malang	Jawa Timur
2	Universitas Airlangga	0.559164	Surabaya	Jawa Timur
3	Institut Teknologi Sepuluh Nopember	0.541069	Surabaya	Jawa Timur
4	Universitas Muhammadiyah Malang	0.527906	Malang	Jawa Timur
5	Universitas Kristen Petra	0.488639	Surabaya	Jawa Timur
6	Universitas Surabaya	0.472489	Surabaya	Jawa Timur
7	Universitas Muhammadiyah Sidoarjo	0.466495	Sidoarjo	Jawa Timur
8	Universitas Widya Kartika	0.462224	Surabaya	Jawa Timur
9	Universitas Merdeka Malang	0.456248	Malang	Jawa Timur
10	Universitas Trunojoyo	0.449357	Bangkalan	Jawa Timur

Gambar 4.4-11 Antarmuka Hasil Pencarian Berdasar Region

4.4.9 Antarmuka Pembobotan Pemingkatan Perguruan Tinggi

Pada antarmuka ini pengguna dapat menentukan bobot setiap kriteria perguruan tinggi sesuai dengan kebutuhan. Terdapat beberapa tipe pemingkatan yang dapat digunakan untuk menghasilkan peringkat.

IDNUniversity
Ranking
Metode Penilaian Region Lihat Ranking Cari Disini

Atur Bobot Pemingkatan

Pembobotan Pemingkatan									
Indikator	QS General	QS Asia	Webometrics	4ICU	DIKTI	QS (L-LDA)	QS (PLSA)	QS (Heinholzt)	QS (Jadaboost)
Reputasi Akademik	<input type="text" value="40"/>	<input type="text" value="30"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="26"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Reputasi Employer	<input type="text" value="10"/>	<input type="text" value="20"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="14"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Jumlah Mahasiswa Aktif	<input type="text" value="20"/>	<input type="text" value="15"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Jumlah Situs Akademisi	<input type="text" value="20"/>	<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="15"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Jumlah Tenaga Pengajar	<input type="text" value="5"/>	<input type="text" value="2.5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Kualitas Situs (Hindex & I5index)	<input type="text" value="5"/>	<input type="text" value="2.5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="3"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Jumlah Phd (B3)	<input type="text" value="0"/>	<input type="text" value="5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="20"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Jumlah Paper Akademisi	<input type="text" value="0"/>	<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="2"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Kecukupan Dosen	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="18"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Kualitas Manajemen	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="30"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Kualitas Kegiatan Kemahasiswaan	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Kualitas Kegiatan Penelitian	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="30"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Jumlah Total	100	100	100	100	100	100	100	100	100
	CR QS-General	CR QS-Asia	CR Webometrics	CR 4ICU	CR DIKTI	CR QS L-LDA	CR QS PLSA	CR QS Heinholzt	CR QS Jadaboost

Gambar 4.4-12 Antarmuka Pembobotan Pemingkatan

4.4.10 Antarmuka Hasil Pemingkatan Perguruan Tinggi dengan Berbagai Macam Tipe Pemingkatan

Pada antarmuka ini pengguna dapat melihat hasil pemingkatan perguruan tinggi dari bobot yang sudah ditentukan pada proses pembobotan. Terdapat hasil pemingkatan dari berbagai macam tipe pemingkatan, diantaranya QS General, QS Asia, Webometrics, 4ICU dan pemingkatan DIKTI.

Peringkat Perguruan Tinggi (QS General)

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi	Compare (gilit 3)
1	Institut Teknologi Bandung	0.6601	Bandung	Jawa Barat	<input type="checkbox"/>
2	Universitas Gadjah Mada	0.6522	Yogyakarta	DI Yogyakarta	<input type="checkbox"/>
3	Universitas Indonesia	0.6414	Depok	Jawa Barat	<input type="checkbox"/>
4	Institut Pertanian Bogor	0.5673	Bogor	Jawa Barat	<input type="checkbox"/>
5	Universitas Diponegoro	0.5623	Semarang	Jawa Tengah	<input type="checkbox"/>
6	Universitas Brawijaya	0.5612	Malang	Jawa Timur	<input type="checkbox"/>
7	Universitas Airlangga	0.5592	Surabaya	Jawa Timur	<input type="checkbox"/>
8	Institut Teknologi Sepuluh Nopember	0.5411	Surabaya	Jawa Timur	<input type="checkbox"/>
9	Universitas Muhammadiyah Malang	0.5279	Malang	Jawa Timur	<input type="checkbox"/>
10	Universitas Padjadjaran	0.5224	Bandung	Jawa Barat	<input type="checkbox"/>

Gambar 4.4-13 Antarmuka Hasil Pemeringkatan QS General

Peringkat Perguruan Tinggi (QS Asia)

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Institut Teknologi Bandung	0.7626	Bandung	Jawa Barat
2	Universitas Indonesia	0.7296	Depok	Jawa Barat
3	Universitas Gadjah Mada	0.7235	Yogyakarta	DI Yogyakarta
4	Institut Pertanian Bogor	0.6998	Bogor	Jawa Barat
5	Universitas Brawijaya	0.6165	Malang	Jawa Timur
6	Universitas Diponegoro	0.6053	Semarang	Jawa Tengah
7	Institut Teknologi Sepuluh Nopember	0.5995	Surabaya	Jawa Timur
8	Universitas Airlangga	0.5992	Surabaya	Jawa Timur
9	Universitas Hasanuddin	0.5671	Makassar	Sulawesi Selatan
10	Universitas Padjadjaran	0.5656	Bandung	Jawa Barat

Gambar 4.4-14 Antarmuka Hasil Pemeringkatan QS Asia

Peringkat Perguruan Tinggi (Webometrics)

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Universitas Gadjah Mada	0.8207	Yogyakarta	DI Yogyakarta
2	Institut Teknologi Bandung	0.6314	Bandung	Jawa Barat
3	Universitas Diponegoro	0.5234	Semarang	Jawa Tengah
4	Universitas Sumatera Utara	0.4745	Medan	Sumatera Utara
5	Institut Pertanian Bogor	0.4659	Bogor	Jawa Barat
6	Universitas Indonesia	0.4468	Depok	Jawa Barat
7	Universitas Negeri Yogyakarta	0.4034	Yogyakarta	DI Yogyakarta
8	Institut Teknologi Sepuluh Nopember	0.3968	Surabaya	Jawa Timur
9	Universitas Brawijaya	0.3843	Malang	Jawa Timur
10	Universitas Padjadjaran	0.3605	Bandung	Jawa Barat

Gambar 4.4-15 Antarmuka Hasil Pemeringkatan Webometrics

Peringkat Perguruan Tinggi (4ICU)

Masukkan Nama Kota Diisi...

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Universitas Sumatera Utara	0.9213	Medan	Sumatera Utara
2	Institut Teknologi Bandung	0.834	Bandung	Jawa Barat
3	Universitas Indonesia	0.8063	Dipok	Jawa Barat
4	Universitas Gadjah Mada	0.8205	Yogyakarta	DI Yogyakarta
5	Institut Pertanian Bogor	0.8105	Bogor	Jawa Barat
6	Universitas Airlangga	0.7911	Surabaya	Jawa Timur
7	Universitas Padjadjaran	0.7858	Bandung	Jawa Barat
8	Universitas Diponegoro	0.7839	Semarang	Jawa Tengah
9	Universitas Pendidikan Indonesia	0.7788	Bandung	Jawa Barat

Gambar 4.4-16 Antarmuka Hasil Pemeringkatan 4ICU

Peringkat Perguruan Tinggi (Pemeringkatan DIKTI)

Masukkan Nama Kota Diisi...

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Institut Teknologi Bandung	0.6988	Bandung	Jawa Barat
2	Institut Pertanian Bogor	0.5355	Bogor	Jawa Barat
3	Institut Teknologi Sepuluh Nopember	0.5101	Surabaya	Jawa Timur
4	Universitas Indonesia	0.4727	Dipok	Jawa Barat
5	Universitas Gadjah Mada	0.4449	Yogyakarta	DI Yogyakarta
6	Universitas Negeri Jakarta	0.4188	Jakarta	DKI Jakarta
7	Universitas Diponegoro	0.406	Semarang	Jawa Tengah
8	Universitas Hasanuddin	0.4009	Makassar	Sulawesi Selatan

Gambar 4.4-17 Antarmuka Hasil Pemeringkatan DIKTI

Peringkat Perguruan Tinggi (QS L-LDA)

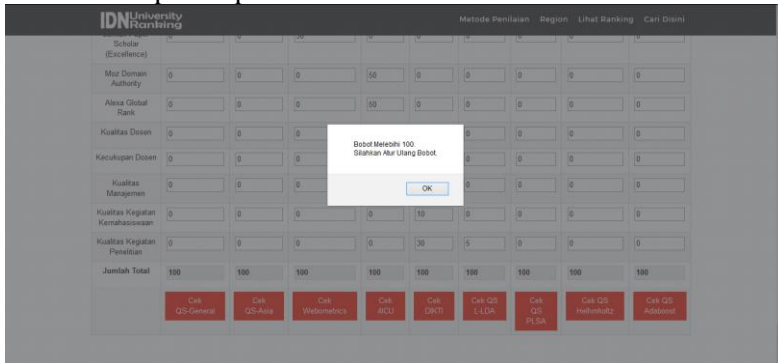
Masukkan Nama Kota Diisi...

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Universitas Indonesia	0.8085	Dipok	Jawa Barat
2	Institut Teknologi Bandung	0.7978	Bandung	Jawa Barat
3	Universitas Gadjah Mada	0.7952	Yogyakarta	DI Yogyakarta
4	Institut Pertanian Bogor	0.6802	Bogor	Jawa Barat
5	Universitas Airlangga	0.6178	Surabaya	Jawa Timur
6	Universitas Brawijaya	0.6117	Malang	Jawa Timur
7	Universitas Diponegoro	0.6014	Semarang	Jawa Tengah
8	Universitas Hasanuddin	0.5956	Makassar	Sulawesi Selatan
9	Universitas Bina Nusantara	0.5939	Jakarta	DKI Jakarta
10	Universitas Padjadjaran	0.575	Bandung	Jawa Barat
11	Institut Teknologi Sepuluh Nopember	0.5566	Surabaya	Jawa Timur
12	Universitas Udayana	0.5054	Badung	Bali

Gambar 4.4-18 Antarmuka Hasil Peringkat QS LLDA

4.4.11 Antarmuka Error Handling Pembobotan Kriteria Pemingkatan

Pada antarmuka ini pengguna dapat melihat pemberitahuan bahwa bobot yang diharuskan untuk diinputkan sebagai masukan pembobotan adalah 100, jika melebihi 100 maka akan menampilkan pemberitahuan.



Gambar 4.4-19 Antarmuka Error Handling Pembobotan

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini dibahas mengenai hasil uji coba perangkat lunak yang telah dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan dan menguji apakah fungsionalitas aplikasi telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba ini menjelaskan lingkungan yang digunakan untuk menguji implementasi dari pembuatan perangkat lunak pada Tugas Akhir. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i7-5500U CPU @2.40GHz (4CPUs)
 - b. Memori (RAM): 12 GB
 - c. Hardisk (HDD): 1 TB
 - d. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: Windows 8.1 Profesional x64 (64-bit)
 - b. Perangkat pengembang: Python v2.7.12, Anaconda 4.2.0 (64-bit), dan SublimeText 2 v2.0.

5.2 Data Uji Coba

Data uji coba digunakan untuk menguji jalannya sistem apakah sudah sesuai dengan tujuan dan lingkup pengerjaan yang ditunjukkan. Data uji coba yang digunakan dalam pengembangan perangkat lunak terdiri dari beberapa macam, diantaranya:

5.2.1 Data Dokumen Abstrak Paper Akademisi

Data uji coba yang digunakan dalam proses teks mining menggunakan metode L-LDA adalah dokumen abstrak *paper*

akademisi dari 16 perguruan tinggi yang sesuai dengan *Ground Truth* Reputasi Akademik *QS World University rankings* 2015-2016. Penggunaan 16 perguruan tinggi ini didasarkan pada kepopuleran dan kualitas akademisi perguruan tinggi sesuai dengan standart *QS World University rankings* [4]. Tidak semua perguruan tinggi dapat masuk dalam peringkat *QS World University rankings*, baik secara general maupun secara terpisah dari kriteria-kriteria penilaiannya.

Daftar 16 perguruan tinggi berdasarkan peringkat kriteria reputasi akademik *QS World University Rankings* 2015-2016 seperti pada Tabel 5.2-1.

Tabel 5.2-1 Peringkat Perguruan Tinggi

No.	Perguruan Tinggi	Peringkat
1.	Institut Teknologi Bandung	1
2.	Universitas Indonesia	2
3.	Universitas Gadjah Mada	3
4.	Universitas Muhammadiyah Semarang	4
5.	Universitas Airlangga	5
6.	Institut Pertanian Bogor	6
7.	Univeristas Diponegoro	7
8.	Universitas Brawijaya	8
9.	Institut Teknologi Sepuluh Nopember	9
10.	Universitas Padjadjaran	10
11.	Universitas Islam Indonesia	11
12.	Univeritas 17 Agustus 1945	12
13.	Universitas Bina Nusantara	13
14.	Universitas Pelita Harapan	14
15.	Universitas Sebelas Maret	15
16.	Universitas Mataram	16

Dari 16 perguruan tinggi diatas, digunakan 50 dokumen abstrak *paper* dari masing-masing perguruan tinggi dan dihasilkan 800 buah dokumen dengan total kata yang dimiliki sejumlah 139919

buah kata. Data ini yang selanjutnya akan disebut dengan dataset abstrak *paper*.

Kata-kata ini akan diproses lebih lanjut dengan tahap preprocessing, untuk menganalisa duplikasi kata, karakter non alfabet, dan banyak frekuensi kata yang tidak berkepentingan yang ada di dalam dokumen. Tahap preprocessing yang digunakan meliputi case folding, tokenisasi, remove punctuation, dan stopwords removal [22]. Hasil reduksi jumlah kata dari semua dataset abstrak *paper* akademisi seperti pada Tabel 5.2-2.

Tabel 5.2-2 Daftar Reduksi Jumlah Kata Dataset

No.	Perguruan Tinggi	Kata Asli	Kata Hasil Preprocess
1.	Institut Pertanian Bogor	10539	6293
2.	Institut Teknologi Bandung	9287	5675
3.	Institut Teknologi Sepuluh Nopember	9630	6015
4.	Universitas 17 Agustus 1945	9146	5580
5.	Univeritas Airlangga	9457	6009
6.	Universitas Bina Nusantara	7236	4432
7.	Universitas Brawijaya	8651	5469
8.	Universitas Diponegoro	8397	5327
9.	Universitas Gadjah Mada	10214	6279
10.	Universitas Indonesia	8823	5645
11.	Universitas Islam Indonesia	7897	4989
12.	Universitas Mataram	7527	4561
13.	Universitas Muhammadiyah Surakarta	10330	6117
14.	Universitas Padjadjaran	7774	5008
15.	Universitas Pelita Harapan	7204	4542
16.	Universitas Sebelas Maret	7808	5244

Dari Tabel 5.2.-2, hasil reduksi pada tahap preprocessing didapatkan rata-rata 62.42% kata bersih dari total semua jumlah kata. Kata-kata bersih ini yang kemudian akan menjadi masukan untuk tahap selanjutnya.

5.2.2 Data Corpus Taxonomy Bloom

Selain penggunaan dataset abstrak paper, dalam menghasilkan proses teks mining topic modeling metode L-LDA membutuhkan corpus data yang berisikan kumpulan kata kunci (*keyword*). Terdapat corpus Bloom Taxonomy yang sudah ada sebelumnya. Masing-masing corpus Bloom Taxonomy mempunyai jumlah *keyword* pada masing-masing level (6 level). Dalam Tugas Akhir ini mengacu pada beberapa referensi Bloom Taxonomy, diantaranya:

1. www.cte.cornell.edu
2. www.bloomtaxonomy.org
3. www.teaching.uncc.edu
4. www.web.stanford.edu

Dari masing-masing referensi Bloom Taxonomy diatas, didapatkan jumlah *keyword* seperti pada Tabel 5.2-3.

Tabel 5.2-3 Rincian Jumlah Kata Corpus Taxonomy Bloom

No.	Cte.cor nell.edu	Bloomtaxo nomy.org	(teaching.uncc.edu)			Web.stan ford.edu
			KC Metro	UMUC	GA Tech	
1.	31	21	21	9	10	8
2.	43	15	15	9	8	12
3.	47	16	15	9	9	12
4.	40	25	24	9	7	19
5.	52	34	36	7	10	14
6.	36	34	36	6	15	13

Keterangan:

1. Nomor 1, menyatakan keyword pada level Knowledge.
2. Nomor 2, menyatakan keyword pada level Comprehension.
3. Nomor 3, menyatakan keyyord pada level Application.
4. Nomor 4, menyatakan keyword pada level Analysis.
5. Nomor 5, menyatakan keyword pada level Synthesis.
6. Nomor 6, menyatakan keyword pada level Evaluation.

Dalam Tugas Akhir ini menggabungkan semua corpus Bloom Taxonomy dari ke-4 referensi yang sudah disebutkan. Memungkinkan ada *keyword* yang sama, dan kemudian akan direduksi dengan cara menghapus *keyword* yang sama. Tujuan dari menggabungkan adalah untuk memperkaya corpus Bloom Taxonomy sebagai dasar untuk membuat corpus Tingkat Kesiapterapan Teknologi (TKT).

Hasil penggabungan 4 referensi corpus Bloom Taxonomy dengan mereduksi *keyword* yang sama menjadi corpus Bloom Taxonomy final seperti pada Tabel 5.2.-4.

Tabel 5.2-4 Penggabungan Referensi Corpus Taxonomy Bloom

No.	Level	Jumlah Kata
1.	Knowledge	35
2.	Comprehension	29
3.	Application	36
4.	Analysis	51
5.	Synthesis	51
6.	Evaluation	46

5.2.3 Data Corpus Tingkat Kesiapterapan Teknologi (TKT)

Data corpus TKT didapatkan dari beberapa cara yaitu dengan breakdown 6 level corpus Bloom Taxonomy, dan pengembangan menggunakan beberapa cara seperti teknik POS Tagging dan Sinonim Word. Tujuannya adalah untuk memperkaya *keyword* pada corpus TKT.

5.2.3.1 Breakdown 6 Level Corpus Bloom Taxonomy

Dalam corpus Bloom Taxonomy terdapat *keyword* yang terdiri dari lebih dari satu kata, sehingga tidak dapat digunakan untuk pencocokan pada proses selanjutnya, maka dari itu *keyword* yang terdiri dari lebih dari satu kata akan dipecah menjadi satu-satu. Sehingga jumlah kata pada 6 level Corpus

Bloom Taxonomy juga mengalami proses reduksi dan penambahan.

Dari hasil corpus Blom Taxonomy yang sudah mengandung kata yang berbeda di masing-masing level akan diurutkan berdasarkan tingkat makna konten kata. Kata yang mempunyai makna dasar berada pada urutan paling bawah, dan yang mempunyai makna implementasi/pengembangan berada pada urutan paling atas. Urutan tingkat makna kata ini akan dipecah menjadi 9 bagian sebagai implementasi terhadap corpus TKT yang mempunyai 9 level. Hasil breakdown menjadi 9 level corpus TKT seperti pada Tabel 5.2-5.

Tabel 5.2-5 Hasil Breakdown Menjadi 9 Corpus TKT

No.	Level TKT	Jumlah Kata
1.	TKT 1	31
2.	TKT 2	24
3.	TKT 3	32
4.	TKT 4	15
5.	TKT 5	31
6.	TKT 6	24
7.	TKT 7	26
8.	TKT 8	30
9.	TKT 9	34

5.2.3.2 Pengembangan Menggunakan POS Tagging

Hasil corpus TKT menggunakan cara breakdown dari corpus Bloom Taxonomy memperoleh kata-kata dasar, dan jumlahnya masih cenderung sedikit bila dibandingkan dengan kata (*term*) pada dataset yang mencapai 139919 buah kata. Hal ini menjadi salah satu kunci hasil akurasi kecocokan kata. Dengan penambahan menggunakan POS Tagging diharapkan mampu menambah pengetahuan *keyword* dalam corpus TKT. Dilakukan beberapa percobaan untuk memperkaya corpus TKT dengan POS Tagging, diantaranya:

1. Akuisisi kata 1 (-1, +1): akuisisi kata naik satu kata dan turun satu kata.
2. Akuisisi kata 2 (-2, +2): akuisisi kata naik dua kata dan turun dua kata.
3. Akuisisi kata 3 (-3, +3): akuisisi kata naik tiga kata dan turun tiga kata.
4. Akuisisi kata 4 (-4, +4): akuisisi kata naik empat kata dan turun empat kata.
5. Akuisisi kata 8 (-5, +5): akuisisi kata naik delapan kata dan turun delapan kata.
6. Akuisisi kata 10 (-10, +10): akuisisi kata naik sepuluh kata dan turun sepuluh kata.

Hasil penambahan kata corpus TKT menggunakan POS Tagging dengan percobaan akuisisi seperti pada Tabel 5.2-6.

Tabel 5.2-6 Hasil Penambahan Kata Corpus TKT Menggunakan POS Tangging

Level	Jumlah Kata (default)	Jumlah Kata Akuisisi untuk TKT ke-					
		1	2	3	4	8	10
TKT 1	31	379	610	789	917	1313	1481
TKT 2	24	233	328	420	506	735	820
TKT 3	32	321	489	611	741	1032	1123
TKT 4	15	275	436	561	778	1010	1139
TKT 5	31	212	332	425	504	741	796
TKT 6	24	228	324	407	463	587	618
TKT 7	26	156	224	295	347	527	584
TKT 8	30	90	122	159	193	288	323
TKT 9	34	169	259	300	345	517	581

Keyword dalam masing-masing level corpus TKT hasil dari breakdown corpus Bloom Taxonomy dan teknik POS Tagging ini akan menjadi corpus TKT final dalam teks mining. Kecerdasan

dokumen semakin bagus jika corpus yang dimiliki semakin banyak keywordnya.

5.2.3.3 Pengembangan Menggunakan Sinonim Word WordNet

Salah satu cara untuk memperkaya corpus adalah dengan mencari sinonim kata dari setiap kata yang ada pada 9 level corpus TKT. Cara ini digunakan sebagai cara pengganti dari penggunaan teknik POS Tagging, karena berdasar hasil analisa akhir menghasilkan hasil yang kurang memuaskan. Penggunaan sinonim word ini cukup efektif karena lebih menghemat waktu kerja dan juga mendapatkan banyak kata yang bisa diserap dalam proses pencocokan kata daripada menggunakan teknik POS Tagging. Kata dari hasil penambahan sinonim ini akan ditambahkan pada level corpus TKT kata dasarnya. Sehingga menjadikan setiap level TKT menjadi kaya akan kata.

Dilakukan beberapa percobaan untuk memperkaya corpus menggunakan sinonim word, diantaranya:

1. Sinonim level 1: mencari satu sinonim teratas dari setiap kata dari ke-9 level TKT yang dilihat dari probabilitas kesamaan paling tinggi.
2. Sinonim level 2: mencari dua sinonim teratas yang dilihat dari dua probabilitas kesamaan paling tinggi.
3. Sinonim level 3: mencari tiga sinonim teratas yang dilihat dari tiga probabilitas kesamaan paling tinggi.
4. Sinonim level all: mengambil semua sinonim dari setiap kata dari ke-9 level TKT.

Hasil penambahan kata corpus TKT menggunakan sinonim word seperti pada Tabel 5.2-7.

Tabel 5.2-7 Hasil Penambahan Kata Corpus TKT Menggunakan Sinonim Word

Level	Jumlah Kata (default)	Total kata corpus TKT dengan Level sinonim word			
		1	2	3	all
TKT 1	31	50	57	62	132

TKT 2	24	43	54	59	75
TKT 3	32	41	44	47	70
TKT 4	15	25	29	30	50
TKT 5	31	55	68	73	120
TKT 6	24	28	29	29	31
TKT 7	26	48	58	63	122
TKT 8	30	55	66	72	109
TKT 9	34	60	74	77	100

Dari percobaan memperkaya corpus TKT diatas baik menggunakan teknik POS Tagging dan sinonim word, satu per satu diuji coba untuk melakukan pencocokan kata sebagai asumsi label awal dokumen. Disimpulkan bahwa hasil akhir yang paling bagus dari segi distance peringkat perguruan tinggi akhir adalah memperkaya corpus menggunakan sinonim word level all.

5.2.4 Rekap Jumlah Mahasiswa dan Tenaga Pengajar

Data jumlah mahasiswa dan tenaga pengajar merupakan salah satu data utama untuk kriteria pemeringkatan. Data ini didapatkan dari database Pangkalan Data Pendidikan Tinggi (PDDIKTI). Pengguna umum dapat mengakses halaman pada alamat <http://www.forlap.dikti.go.id>. Data yang disediakan dapat diunduh menjadi file excel.

5.2.4.1 Data Jumlah Mahasiswa

Data jumlah mahasiswa yang digunakan adalah mengacu pada data perguruan tinggi yang terdaftar pada PDDIKTI. Perguruan tinggi yang terdaftar meliputi Institut, Universitas, Sekolah Tinggi, Sekolah Tinggi Ilmu Kesehatan, Akademi, dan Politeknik.

Dalam Tugas Akhir ini dibatasi perguruan tinggi dalam lingkup Universitas dan Institut saja karena ranah ilmunya cukup merata. Terdapat 2 kelas yaitu perguruan tinggi negeri dan

perguruan tinggi swasta yang dibagi menjadi 14 sub wilayah kopertis. Informasi yang akan digunakan untuk masukan sistem adalah dari beberapa atribut pada rekapan data, yaitu masing-masing perguruan tinggi akan diambil jumlah mahasiswa pada 2 tahun terakhir dalam periode rekap. Misal untuk rekap tahun 2017, maka data yang digunakan adalah jumlah mahasiswa semester 1 dan 2 tahun 2016, dan semester 1 dan 2 tahun 2017. Rincian jumlah perguruan tinggi dalam rekap seperti pada Tabel 5.2-8.

Tabel 5.2-8 Daftar Jumlah Perguruan Tinggi Berdasarkan Jenis Wilayah Administrasi

No.	Jenis Perguruan Tinggi	Jumlah Perguruan Tinggi
1.	PTN	122
2.	Kopertis wilayah 1	266
3.	Kopertis wilayah 2	217
4.	Kopertis wilayah 3	333
5.	Kopertis wilayah 4	478
6.	Kopertis wilayah 5	105
7.	Kopertis wilayah 6	249
8.	Kopertis wilayah 7	330
9.	Kopertis wilayah 8	165
10.	Kopertis wilayah 9	364
11.	Kopertis wilayah 10	248
12.	Kopertis wilayah 11	168
13.	Kopertis wilayah 12	44
14.	Kopertis wilayah 13	109
15.	Kopertis wilayah 14	60

5.2.4.2 Data Jumlah Tenaga Pengajar

Data jumlah tenaga pengajar mempunyai beberapa kriteria khusus yang disediakan oleh PDDIKTI, diantaranya rekap tenaga pengajar berdasarkan jabatan akademik, berdasarkan status kepegawaian, dan berdasarkan pendidikan tertinggi. Daftar perguruan tinggi yang digunakan sama seperti Tabel 5.2.4.1 yang hanya dipilih yang berbadan Institut atau Universitas saja.

Informasi yang akan digunakan untuk masukan sistem dari beberapa atribut pada rekapan data, diantaranya:

1. Berdasarkan status pendidikan tertinggi
 - a) Jenjang diploma (D1, D2, D3, D4)
 - b) Jenjang sarjana (S1, S2, S3)
 - c) Jenjang spesialis (SP1, SP2)
 - d) Jenjang profesi
 - e) Jenang non-formal dan informal
2. Berdasarkan jabatan
 - a) Jabatan akademik lektor
 - b) Jabatan akademik lektor kepala
 - c) Jabatan akademik profesor
3. Berdasarkan status kepegawaian
 - a) Dosen dengan status tetap
 - b) Dosen dengan status tidak tetap

5.2.5 Data Online Science and Technologu Index (SINTA)

Data online dari *Science and Technology Index* (SINTA) merupakan data yang berhubungan dengan jumlah penelitian akademisi perguruan tinggi. SINTA dapat diakses pada alamat www.sinta1.ristekdikti.go.id. Data yang disediakan berupa informasi yang ditampilkan dalam konten website. Oleh karena itu dalam kriteria ini proses pengambilan data menggunakan teknik *grabbing* dari halaman website.

Pemilihan data online dari SINTA bertujuan untuk mengumpulkan informasi tentang penelitian akademisi yang mempunyai standart tinggi. Data yang akan diolah harus masuk dalam skala database Scopus sebagai pusat data penelitian terbaik saat ini. Data yang dihimpun oleh SINTA sesuai dengan kriteria keberadaan di Scopus [18].

Menggunakan teknik *grabbing* dari website SINTA untuk mengoptimasi pengolahan data yang berbasis pada Scopus, karena Scopus tidak memberikan akses secara mudah untuk dapat menggunakan databasenya. Sehingga penggunaan pihak ketiga

sebagai penyedia data menjadi alternatif dalam penentuan data kriteria ini.

Berikut beberapa konten yang dapat diambil dari website SINTA dan digunakan dalam kriteria penilaian:

1. Authors, merupakan jumlah penulis penelitian yang ter-index oleh Google Scholar dan Scopus.
2. Google Citation, merupakan jumlah sitasi dari semua author dalam perguruan tinggi yang ter-index oleh Google Scholar.
3. Google Documents, merupakan jumlah dokumen penelitian (*paper*) yang ter-index Google Scholar.
4. Google Scholar Indexed, merupakan jumlah h-index dan i10-index dari setiap akademisi yang ter-index oleh Google Scholar.
5. Scopus Indexed, merupakan jumlah h-index dan i10-index dari setiap akademisi yang ter-index oleh Scopus.
6. Journal Articles, merupakan jumlah paper penelitian yang masuk dalam jurnal.
7. Articles in Scopus, merupakan artikel penelitian dari akademisi yang masuk dalam database Scopus.
8. Non Articles in Scopus, merupakan non artikel yang ter-index oleh database Scopus.

Sebagai catatan bahwa data yang disediakan oleh SINTA untuk Tugas Akhir ini kurang valid, karena database SINTA yang digunakan tidak diperbaharui secara berkala. Data SINTA yang digunakan dalam Tugas Akhir ini menggunakan data pada periode 2016-2017. Sehingga jumlah data pada masing-masing variable yang disediakan SINTA perlu dilakukan validasi lagi mengikuti periode update data SINTA setiap tahunnya.

Alasan menggunakan data dari SINTA karena data yang disediakan bersifat *free* dan sesuai dengan variable indikator penilaian *Ground Truth* yaitu *QS World University Rankings* sebagai indikator jumlah sitasi. Pada saat dimulainya penelitian Tugas Akhir ini data yang tersedia adalah data pada periode tahun 2016-2017. Jika data yang digunakan diperbaharui setiap

periodenya, maka hasil akhir sistem pemeringkatan Perguruan Tinggi ini akan berubah sesuai kondisi data yang *real time* dengan bobot indikator yang cukup besar yaitu 20%.

5.2.6 Data Online Analisa Webhost

Data analisa webhost merupakan data yang didapatkan dari teknik *grabbing* pada informasi *webhost*. Informasi yang didapatkan berfokus pada traffict website perguruan tinggi.

Penggunaan analisa *webhost* ini merupakan salah satu kriteria penilaian pada tipe pemeringkatan lain yaitu 4ICU dan Webometrics, yang sebagian besar kriteria masih bergantung pada informasi dan jumlah traffict website. Dalam Tugas Akhir ini, kriteria yang bergantung pada traffict website akan digunakan sebagai kriteria tambahan yang dapat digunakan oleh pengguna perangkat lunak jika ingin mengembangkan masukan kriteria penilaian.

Pengambilan data online ini memanfaatkan beberapa API tambahan untuk memaksimalkan hasil dan memudahkan analisa. Memanfaatkan API sangat dianjurkan untuk mengurangi running time dan detail dalam penyediaan informasi. API yang digunakan antara lain Alexa Rank API, dan MOZ Analytic API.

5.2.6.1 Alexa Rank

Alexa Rank API berfokus pada pengalihan informasi traffict rank dari *webhost* (domain, subdomain). Informasi yang didapatkan menggunakan Alexa Rank API antara lain:

1. World Rank, merupakan peringkat *webhost* dalam peringkat dunia. Peringkat ini sangat mempengaruhi kepopuleran sebuah website. Semakin tinggi peringkat, maka akan semakin kecil nominal rank yang didapatkan.
2. Country Rank, merupakan peringkat *webhost* dalam peringkat negara. Peringkat ini diperoleh dari analisa lokasi webhost berasal.

3. Country Name, merupakan informasi lokasi negara dimana *webhost* berasal.

5.2.6.2 MOZ Analytic

MOZ Analytic API berfokus pada penggalian informasi kepopuleran *webhost* yang dilihat dari jumlah backlink yang dimiliki. Informasi yang didapatkan menggunakan MOZ Analytic API antara lain:

1. Domain Authority (DA), merupakan tingkat traffict pada domain utama maupun subdomain website perguruan tinggi.
2. Page Authority (PA), merupakan tingkat traffict pada page tertentu saja dari website perguruan tinggi.
3. External Backlink, jumlah traffict akses pengguna terhadap link yang terdapat pada sebuah page website lain.
4. Domain Name, merupakan informasi domain name milik *webhost*.

5.2.6.3 Data Jumlah Webhost dari Google

Data jumlah webhost yang digunakan adalah jumlah domain utama, subdomain, dan repository dari website perguruan tinggi yang ter-*index* oleh Google. Pengambilan data ini menggunakan teknik *grabbing* dengan memanfaatkan API dari bahasa pemrograman PHP yaitu simpleHTMLDOM. Dimana sistem akan mengambil konten website pencarian Google, kemudian mengambil konten *results page* yang didapatkan, hasil *results page* ini yang akan dijumlah dan disimpan sebagai jumlah *webhost*. Keseluruhan data online analisa *webhost*, data rekap jumlah mahasiswa dan tenaga pengajar akan disimpan dalam database yang sudah dijelaskan pada Bab 3.

5.3 Skenario Pengujian

Dalam uji coba yang dilakukan dalam tugas akhir ini memiliki beberapa tahapan yang dijelaskan pada subbab ini.

5.3.1 Skenario Pengambilan Data Online Menggunakan Teknik *Grabbing*

Skenario pengambilan data online menggunakan jaringan internet untuk tersambung pada server database maupun website yang akan diambil datanya.

5.3.1.1 Skor Nilai Akreditasi

Metode	: Teknik <i>Grabbing</i>
Sasaran	: Penyedia informasi akreditasi perguruan tinggi <ul style="list-style-type: none"> - https://banpt.or.id/direktori/institusi/pencarian_institusi - http://daftarkampus.com/daftar-akreditasi-perguruan-tinggi-negeri-dan-swasta-lengkap/
Masukan	: Halaman website pada poin sasaran
Keluaran	: Hasil informasi dalam beberapa atribut <ul style="list-style-type: none"> - nama_universitas - peringkat_huruf - masa_berlaku - status_berlaku - skor_masa_berlaku - tmp_nama_universitas
Jumlah Data	: 921 data
Tabel Database	: <code>tampung_akreditasi</code>
Tipe Tabel	: Single table
Tipe Pemrograman	: Bahasa PHP dan Library
Status Kebutuhan	: Terpenuhi

5.3.1.2 Skor Jumlah Sitasi

Metode	: Teknik <i>Grabbing</i>
Sasaran Dikti	: Penyedia informasi dari database SINTA Ristek Dikti
	- http://sinta1.ristekdikti.go.id/
Masukan	: Halaman website pada poin sasaran
Keluaran	: Hasil informasi dalam beberapa atribut
	- nama_universitas
	- jumlah_author_scholar
	- jumlah_paper_scholar
	- jumlah_sitasi_scholar
	- jumlah_jurnal_artikel
	- jumlah_book_chapter
	- jumlah_paper_conference
	- jumlah_artikel_scopus
	- jumlah_non_artikel_scopus
	- jumlah_sitasi_scopus
	- sinta_score
Jumlah Data	: 3471 data
Tabel Database	:
	- tampung_sitasi_sinta
	- tampung_sitasi_sinta_2
	- tampung_log_sitasi_sinta
	- tampung_sitasi_sinta_3
	- jumlah_sitasi_sinta
Tipe Tabel	: Multi table
Tipe Pemrograman	: Bahasa PHP dan Library
Status Kebutuhan	: Terpenuhi

5.3.1.3 Skor Kualitas Sitasi

Metode	: Teknik <i>Grabbing</i>
Sasaran	: Penyedia informasi dari database Google Scholar dan SINTA Ristek Dikti

	-	https://scholar.google.co.id/
	-	http://sinta1.ristekdikti.go.id/
Masukan	:	Halaman website pada poin sasaran
Keluaran	:	Hasil informasi dalam beberapa atribut <ul style="list-style-type: none"> - nama_universitas - jumlah_sitasi_scholar - jumlah_i10_index_scholar - jumlah_h_index_scholar
Jumlah Data	:	3471 data
Tabel Database	:	<ul style="list-style-type: none"> - tampung_author_sinta - tampung_author_sinta_2 - tampung_log_author_sinta - jumlah_author_sinta - jumlah_author_sinta_2
Tipe Tabel	:	Multi table
Tipe Pemrograman	:	Bahasa PHP dan Library
Status Kebutuhan	:	Terpenuhi

5.3.1.4 Skor Jumlah Paper

Metode	:	Teknik <i>Grabbing</i>
Sasaran	:	Penyedia informasi dari database SINTA Ristek Dikti
	-	http://sinta1.ristekdikti.go.id/
Masukan	:	Halaman website pada poin sasaran
Keluaran	:	Hasil informasi dalam beberapa atribut <ul style="list-style-type: none"> - nama_universitas - jumlah_paper_scholar
Jumlah Data	:	3471 data
Tabel Database	:	<ul style="list-style-type: none"> - tampung_sitasi_sinta - tampung_sitasi_sinta_2 - tampung_log_sitasi_sinta - tampung_sitasi_sinta_3

- jumlah_sitasi_sinta
 Tipe Tabel : Multi table
 Tipe Pemrograman : Bahasa PHP dan Library
 Status Kebutuhan : Terpenuhi

5.3.1.5 Skor Jumlah Artikel Jurnal

Metode : Teknik *Grabbing*
 Sasaran : Penyedia informasi dari database SINTA Ristek Dikti
 Dikti
 - <http://sinta1.ristekdikti.go.id/>
 Masukan : Halaman website pada poin sasaran
 Keluaran : Hasil informasi dalam beberapa atribut
 - nama_universitas
 - jumlah_jurnal_artikel
 Jumlah Data : 3471 data
 Tabel Database :
 - tampung_sitasi_sinta
 - tampung_sitasi_sinta_2
 - tapung_log_sitasi_sinta
 - tampung_sitasi_sinta_3
 - jumlah_sitasi_sinta
 Tipe Tabel : Multi table
 Tipe Pemrograman : Bahasa PHP dan Library
 Status Kebutuhan : Terpenuhi

5.3.1.6 Skor Jumlah Paper Conference

Metode : Teknik *Grabbing*
 Sasaran : Penyedia informasi dari database SINTA Ristek Dikti
 Dikti
 - <http://sinta1.ristekdikti.go.id/>
 Masukan : Halaman website pada poin sasaran
 Keluaran : Hasil informasi dalam beberapa atribut

- nama_universitas
- jumlah_paper_conference

Jumlah Data : 3471 data

Tabel Database :

- tampung_sitasi_sinta
- tampung_sitasi_sinta_2
- tapung_log_sitasi_sinta
- tampung_sitasi_sinta_3
- jumlah_sitasi_sinta

Tipe Tabel : Multi table

Tipe Pemrograman : Bahasa PHP dan Library

Status Kebutuhan : Terpenuhi

5.3.1.7 Skor Precense

Metode : Teknik *Grabbing*

Sasaran : Website mesin pencari Google (index Google)

- <http://www.google.com/search>

Masukan : Domain website perguruan tinggi yang diaplikasikan menggunakan library dan indexing website pada poin sasaran

Keluaran : Hasil informasi dalam beberapa atribut

- url_web
- domain_search
- hasil_search
- tanggal_akses

Jumlah Data : 442 data

Tabel Database : precense

Tipe Tabel : Single table

Tipe Pemrograman : Bahasa PHP dan Library

Status Kebutuhan : Terpenuhi

5.3.1.8 Skor Impact

Metode : Teknik *Grabbing*
 Sasaran : Penyedia informasi analisa SEO menggunakan MOZ Analytics

- <http://lsapi.seomoz.com/linkscope/url-metrics/>

Masukan : Domain website perguruan tinggi yang diaplikasikan menggunakan API dan Token user MOZ Analytics

Keluaran : Hasil informasi dalam beberapa atribut

- url_web
- domain_web
- page_authority
- domain_authority
- backlink_eksternal
- tanggal_akses

Jumlah Data : 447 data

Tabel Database : impact

Tipe Tabel : Single table

Tipe Pemrograman : Bahasa PHP dan Library

Status Kebutuhan : Terpenuhi

5.3.2 Skenario Pengambilan Data pada Tabel Database

Skenario pengambilan data pada database digunakan pada data yang bersifat pengolahan lokal (tanpa campur tangan koneksi internet) yang didapatkan dari pengolahan data online yang sudah disimpan dalam database maupun dari komputasi metode teks mining.

5.3.2.1 Skor Reputasi Akademik

Metode : Analisa Database

Masukan : Hasil skor dari metode teks mining L-LDA terhadap dokumen abstrak *paper* setiap perguruan tinggi

Keluaran : Hasil informasi dalam beberapa atribut

- nama_universitas
- skor_universitas
- bobot_total
- jumlah_dokumen
- tanggal_akses

 Jumlah Data : 122 data
 Tabel Database : reputasi_akademik
 Tipe Tabel : Single table
 Tipe Pemrograman : Bahasa Python dan Library
 Status Kebutuhan : Terpenuhi

5.3.2.2 Skor Jumlah Mahasiswa

Metode : Analisa Database
 Masukan : Dokumen rekap jumlah mahasiswa dari database PDDIKTI
 Keluaran : Hasil informasi dalam beberapa atribut

- kode_universitas
- nama_universitas
- jumlah_tahun1_semester1
- jumlah_tahun1_semester2
- jumlah_tahun2_semester1
- jumlah_tahun2_semester2
- status_universitas

 Jumlah Data : 3247 data
 Tabel Database : jumlah_mahasiswa
 Tipe Tabel : Single table
 Tipe Pemrograman : Bahasa Python dan Library
 Status Kebutuhan : Terpenuhi

5.3.2.3 Skor Jumlah Tenaga Pengajar

Metode : Analisa Database

Masukan : Dokumen rekap jumlah tenaga pengajar dari database PDDIKTI

Keluaran : Hasil informasi dalam beberapa atribut

- kode_universitas
- nama_universitas
- asisten_ahli_10000
- asisten_ahli_15000
- lektor_20000
- lektor_30000
- lektor_kepala_40000
- lektor_kepala_55000
- lektor_kepala_70000
- profesor_85000
- profesor_105000
- tanpa_jabatan
- dosen_tetap
- dosen_tidak_tetap
- status_universitas
- pendidikan_terakhir

Jumlah Data : 3251 data

Tabel Database :

- jumlah_tenaga_pengajar_jabatan_akademik
- jumlah_tenaga_pengajar_status_dosen
- jumlah_tenaga_pengajar_tetap
- jumlah_tenaga_pengajar_tidak_tetap

Tipe Tabel : Multi table

Tipe Pemrograman : Bahasa Python dan Library

Status Kebutuhan : Terpenuhi

5.3.2.4 Skor Jumlah PhD

Metode : Analisa Database

Masukan : Dokumen rekap jumlah tenaga pengajar dari database PDDIKTI

Keluaran : Hasil informasi dalam beberapa atribut

- kode_universitas
- nama_universitas
- pendidikan_terakhir_S3

 Jumlah Data : 3251 data

 Tabel Database :

- jumlah_tenaga_pengajar_tetap
- jumlah_tenaga_pengajar_tidak_tetap

 Tipe Tabel : Multi table

 Tipe Pemrograman : Bahasa Python dan Library

 Status Kebutuhan : Terpenuhi

5.3.2.5 Skor Transparency

Metode : Analisa Database

 Masukan : Hasil query data jumlah sitasi Google Scholar pada tabel tabel_jumlah_sitasi_sinta

 Keluaran : Hasil informasi dalam beberapa atribut

- nama_universitas
- jumlah_sitasi_scholar

 Jumlah Data : 3471 data

 Tabel Database : jumlah_sitasi_sinta

 Tipe Tabel : Single table

 Tipe Pemrograman : Bahasa PHP dan Library

 Status Kebutuhan : Terpenuhi

5.3.2.6 Skor Excellence

Metode : Analisa Database

 Masukan : Hasil query data jumlah paper Google Scholar pada tabel tabel_jumlah_sitasi_sinta

 Keluaran : Hasil informasi dalam beberapa atribut

- nama_universitas
- jumlah_paper_scholar

Jumlah Data : 3471 data
 Tabel Database : jumlah_sitasi_sinta
 Tipe Tabel : Single table
 Tipe Pemrograman : Bahasa PHP dan Library
 Status Kebutuhan : Terpenuhi

5.3.3 Skenario Pengujian Dataset Menggunakan Teks Mining Metode L-LDA

5.3.3.1 Tahap Preprocessing

Dataset dokumen abstrak *paper* yang digunakan berasal dari 16 perguruan tinggi teratas yang masuk dalam *Ground Truth QS World University Rankings 2016-2017*. Rincian jumlah kata asli maupun kata hasil reduksi dalam tahap preprocessing seperti pada Tabel 5.3-1.

Tabel 5.3-1 Hasil Reduksi Kata Tahap Preprocessing

No.	Nama Perguruan Tinggi	Jumlah Kata Asli	Jumlah Kata Hasil Preprocessing	Reduksi Kata
1.	Institut Pertanian Bogor	10539	6293	59,712%
2.	Institut Teknologi Bandung	9287	5675	61,107%
3.	Institut Teknologi Sepuluh Nopember	9630	6015	62,461%
4.	Universitas 17 Agustus 1945	9146	5580	61,01%
5.	Universitas Airlangga	9457	6009	63,54%
6.	Universitas	7236	4432	61,249%

	Bina Nusantara			
7.	Universitas Brawijaya	8651	5469	63,218%
8.	Universitas Diponegoro	8397	5327	63,439%
9.	Universitas Gadjah Mada	10214	6279	61,474%
10.	Universitas Indonesia	8823	5645	63,981%
11.	Universitas Islam Indonesia	7897	4989	63,176%
12.	Universitas Mataram	7527	4561	60,595%
13.	Universitas Muhammadiyah Surakarta	10330	6117	59,216%
14.	Universitas Padjadjaran	7774	5008	64,42%
15.	Universitas Pelita Harapan	7204	4542	63,048%
16.	Universitas Sebelas Maret	7807	5244	67,17%
Rata-rata				62,426%

Berdasarkan Tabel 5.3-1 didapatkan rata-rata hasil reduksi kata dari semua dataset sebesar 62,426%, yang menunjukkan tahap preprocessing mampu menghapus karakter kata yang tidak penting sebesar 37,574%.

Jumlah duplikasi kata yang tinggi dalam sebuah dokumen menjadi tingkat kepentingan sebuah kata menjadi kurang bagus. Tabel 5.3-2 menampilkan jumlah tingkat duplikasi kata dalam dokumen abstrak *paper* akademisi di 16 perguruan tinggi.

Tabel 5.3-2 Tingkat Duplikasi Kata

No	Nama Perguruan Tinggi	Jumlah Kata Hasil Preprocessing	Jumlah Kata (duplikasi)	Jumlah Kata (single)	Duplikasi Kata (%)
1.	Institut Pertanian Bogor	6293	4469	1824	71,051%
2.	Institut Teknologi Bandung	5675	3929	1746	69,233%
3.	Institut Teknologi Sepuluh Nopember	6015	3879	2136	64,489%
4.	Universitas 17 Agustus 1945	5580	3576	2004	64,086%
5.	Universitas Airlangga	6009	4234	1775	70,461%
6.	Universitas Bina Nusantara	4432	3004	1428	67,78%
7.	Universitas Brawijaya	5469	3581	1888	65,478%
8.	Universitas Diponegoro	5327	3519	1808	66,06%
9.	Universitas Gadjah Mada	6279	3979	2300	63,37%
10.	Universitas Indonesia	5645	3641	2004	64,5%
11.	Universitas Islam Indonesia	4989	3036	1953	60,854%
12.	Universitas Mataram	4561	3253	1308	71,322%
13.	Universitas Muhammadiyah Surakarta	6117	4123	1994	67,402%

14.	Universitas Padjadjaran	5008	3203	1805	63,958%
15.	Universitas Pelita Harapan	4542	2994	1548	65,918%
16.	Universitas Sebelas Maret	5244	3806	1538	71,22%
Rata-rata					66,697%

Berdasarkan Tabel 5.3-2 didapatkan rata-rata duplikasi kata dalam dokumen hasil preprocessing sebesar 66,697% yang menunjukkan duplikasi kata sangat besar.

5.3.3.2 Tahap Memperkaya Corpus dengan POS Tagging

Corpus TKT yang didapatkan dari proses breakdown Corpus Bloom Taxonomy akan dikembangkan lagi agar menghasilkan *keyword* yang lebih banyak. Pengembangan ini menggunakan metode POS Tagging. Rincian jumlah kata penambahan dengan menggunakan POS Tagging terhadap default pada corpus TKT seperti pada Tabel 5.3-3.

Tabel 5.3-3 Rincian Jumlah Penambahan Kata dengan POS Tagging Setiap Perguruan Tinggi

No.	Nama Perguruan Tinggi	POS Tagging Akuisisi Ke-					
		1	2	3	4	8	10
1.	Institut Pertanian Bogor	201	329	457	585	1097	1353
2.	Institut Teknologi Bandung	149	241	333	425	793	977
3.	Institut Teknologi Sepuluh Nopember	215	349	483	617	1086	1220

4.	Universitas 17 Agustus 1945	226	386	510	652	1220	1291
5.	Universitas Airlangga	224	364	504	644	1134	1274
6.	Universitas Bina Nusantara	225	369	513	657	1233	1521
7.	Universitas Brawijaya	189	309	429	549	1029	1269
8.	Universitas Diponegoro	177	287	397	507	947	1167
9.	Universitas Gadjah Mada	261	343	425	507	835	999
10.	Universitas Indonesia	227	367	507	647	1207	1487
11.	Universitas Islam Indonesia	226	366	506	646	1206	1486
12.	Universitas Mataram	126	206	286	366	686	846
13.	Universitas Muhammadi yah Surakarta	212	346	480	614	1150	1418
14.	Universitas Padjadjaran	234	380	526	672	1256	1548
15.	Universitas Pelita Harapan	180	292	404	516	964	1188
16.	Universitas Sebelas Maret	159	259	359	459	659	809
Total Keseluruhan		3241	5209	7141	9091	16554	19917
Rata-rata Penambahan		3,7%	6,0%	8,2%	10,5%	19,1%	23,1%

Berdasarkan Tabel 5.3-3 didapatkan rata-rata penambahan kata baru menggunakan teknik POS Tagging dengan variasi akuisisi yang berbeda-beda. Dapat dilihat rata-rata tertinggi adalah 23,1% untuk variasi akuisisi 10.

Semakin besar akuisisi maka presentase penambahan juga semakin tinggi, akan tetapi perlu dilihat juga tingkat kata-kata duplikasi yang dihasilkan. Oleh karena itu tidak bisa menjadi jaminan jika prosentase jumlah penambahan kata tinggi akan menghasilkan corpus yang besar dan benar, perlu adanya proses lebih lanjut untuk menganalisa dan mereduksi kata duplikasi. Jumlah breakdown kata dari hasil POS Tagging per dataset paper pada corpus TKT seperti pada Tabel 5.3-4.

Tabel 5.3-4 Rincian Jumlah Breakdown Hasil POS Tagging pada Corpus TKT

No.	Level	POS Tagging Akuisisi Ke-					
		1	2	3	4	8	10
1.	TKT 1	520	927	1326	1698	3235	3951
2.	TKT 2	358	552	766	971	1722	2091
3.	TKT 3	507	816	1124	1442	2629	1260
4.	TKT 4	462	738	1014	1284	2333	2818
5.	TKT 5	321	528	728	940	1731	2081
6.	TKT 6	421	637	876	1098	1905	1168
7.	TKT 7	236	368	515	650	1211	1422
8.	TKT 8	158	216	275	343	542	628
9.	TKT 9	258	409	517	665	1246	1498
Total Keseluruhan		3241	5209	7141	9091	16554	19917

Hasil breakdown pada Tabel 5.3-4 dihasilkan jumlah kata untuk corpus final TKT, corpus ini akan menjadi dasar untuk proses metode teks mining L-LDA pada percobaan menggunakan teknik POS Tagging. Rincian corpus final TKT seperti pada Tabel 5.3-5.

Tabel 5.3-5 Corpus TKT Final Berdasarkan Percobaan POS Tagging

No.	Level	Jumlah kata default	Corpus final dengan POS Tagging Akuisisi Ke-					
			1	2	3	4	8	10
1.	TKT 1	31	379	610	789	917	1313	1481
2.	TKT 2	24	233	328	420	506	735	820
3.	TKT 3	32	321	489	611	741	1032	1123
4.	TKT 4	15	275	436	561	668	1010	1139
5.	TKT 5	31	212	332	425	504	741	786
6.	TKT 6	24	228	324	407	463	587	618
7.	TKT 7	26	156	224	295	347	527	584
8.	TKT 8	30	90	122	159	193	288	323
9.	TKT 9	34	169	259	300	345	517	581
Total Keseluruhan		256	2063	3124	3967	4684	6750	7465

Jumlah kata pada masing-masing level corpus TKT final berdasarkan penambahan POS Tagging dijabarkan pada poin-poin dibawah ini:

1. Akuisisi POS Tagging 1 (-1, +1)

Tabel 5.3-6 Analisa Kata Hasil Akuisisi POS Tagging 1 (-1, +1)

No.	Nama	Level TKT									Not in TKT
		1	2	3	4	5	6	7	8	9	
1.	IPB	114	67	93	81	66	78	46	15	44	1220
2.	ITB	106	72	93	70	62	58	41	11	42	1191
3.	ITS	120	63	107	83	78	84	53	13	55	1480
4.	UNTAG	101	72	89	82	69	78	39	16	54	1404
5.	UNAIR	109	65	95	75	56	87	41	13	59	1175
6.	BINUS	118	58	92	84	56	74	37	14	42	853
7.	UB	117	72	111	92	67	82	35	9	47	1256
8.	UNDIP	106	59	102	68	60	75	37	12	40	1249
9.	UGM	139	75	125	105	87	85	47	18	55	1564
10.	UI	123	60	121	96	65	85	51	18	49	1336
11.	UIN	109	75	94	84	59	76	42	12	58	1344
12.	UNRAM	93	46	79	58	55	59	30	16	34	838

13.	UMS	112	64	99	94	69	90	48	18	49	1351
14.	UNPAD	123	75	97	69	57	80	52	11	43	1198
15.	UPH	98	64	99	65	67	73	35	15	38	994
16.	UNS	86	59	95	78	74	65	38	7	32	1004

Berdasarkan POS Tagging akuisisi 1 didapatkan rata-rata keberadaan kata dari dataset dokumen abstrak paper sebesar 33,30%.

2. Akuisisi POS Tagging 2 (-2, +2)

Tabel 5.3-7 Analisa Kata Hasil Akuisisi POS Tagging 2 (-2, +2)

No.	Nama	Level TKT									Not in TKT
		1	2	3	4	5	6	7	8	9	
1.	IPB	180	78	134	117	97	100	62	15	65	976
2.	ITB	173	90	140	100	85	78	57	20	56	947
3.	ITS	200	82	158	143	117	118	68	22	74	1154
4.	UNTAG	164	98	131	121	90	107	54	20	79	1140
5.	UNAIR	178	87	132	112	84	114	51	22	86	909
6.	BINUS	167	82	139	113	85	104	48	14	64	612
7.	UB	175	101	154	121	104	110	50	15	72	986
8.	UNDIP	176	79	148	104	93	99	51	21	60	977
9.	UGM	210	94	179	151	120	116	64	21	76	1269
10.	UI	196	86	170	113	102	114	71	22	65	1065
11.	UII	175	94	143	118	86	102	57	21	80	1077
12.	UNRAM	143	58	111	87	86	74	46	16	51	636
13.	UMS	184	89	152	135	101	122	68	25	69	1049
14.	UNPAD	200	93	136	99	88	106	68	16	62	937
15.	UPH	156	81	137	91	94	95	53	18	59	764
16.	UNS	138	77	138	118	103	86	49	12	54	763

Berdasarkan POS Tagging akuisisi 2 didapatkan rata-rata keberadaan kata dari dataset dokumen abstrak paper sebesar 47,84%.

3. Akuisisi POS Tagging 3 (-3, +3)

Tabel 5.3-8 Analisa Kata Hasil Akuisisi POS Tagging 3 (-3, +3)

No.	Nama	Level TKT									Not in TKT
		1	2	3	4	5	6	7	8	9	
1.	IPB	219	97	158	145	111	117	70	16	78	813
2.	ITB	219	100	157	123	101	94	67	24	61	800
3.	ITS	248	102	178	174	143	145	78	24	81	963
4.	UNTAG	208	114	155	150	109	131	66	24	80	967
5.	UNAIR	221	106	154	134	93	135	72	24	94	742
6.	BINUS	198	98	162	135	99	130	55	17	73	461
7.	UB	220	118	180	158	123	137	59	17	77	799
8.	UNDIP	215	94	171	131	107	115	66	21	69	819
9.	UGM	262	115	193	187	141	141	74	21	80	1086
10.	UI	246	98	197	182	117	139	80	27	73	845
11.	UII	219	122	162	150	106	118	64	23	87	902
12.	UNRAM	168	70	127	104	95	85	56	25	53	525
13.	UMS	238	110	170	161	119	150	79	30	87	850
14.	UNPAD	238	117	172	120	104	128	83	20	66	757
15.	UPH	202	104	157	110	105	115	65	23	65	602
16.	UNS	179	86	156	135	123	101	57	15	62	624

Berdasarkan POS Tagging akuisisi 3 didapatkan rata-rata keberadaan kata dari dataset dokumen abstrak paper sebesar 57,17%.

4. Akuisisi POS Tagging 4 (-4, +4)

Tabel 5.3-9 Analisa Kata Hasil Akuisisi POS Tagging 4 (-4, +4)

No.	Nama	Level TKT									Not in TKT
		1	2	3	4	5	6	7	8	9	
1.	IPB	246	107	180	163	126	131	74	19	85	693
2.	ITB	249	119	170	129	113	106	73	27	66	694
3.	ITS	282	113	203	193	155	155	82	27	90	836
4.	UNTAG	222	129	178	167	122	145	71	28	93	849
5.	UNAIR	232	121	169	157	104	147	68	33	102	642

6.	BINUS	219	108	179	151	107	143	60	18	79	364
7.	UB	239	133	203	171	145	152	66	23	86	670
8.	UNDIP	243	111	193	153	131	129	64	23	71	690
9.	UGM	294	128	208	215	166	157	82	30	84	936
10.	UI	269	119	227	195	139	155	82	29	82	707
11.	UII	242	137	185	162	118	134	76	29	89	781
12.	UNRAM	186	76	149	116	113	93	61	26	51	437
13.	UMS	262	131	204	180	138	162	88	33	87	709
14.	UNPAD	269	129	198	144	112	144	86	21	71	631
15.	UPH	219	114	177	118	127	126	69	25	68	505
16.	UNS	197	99	176	148	138	107	60	18	70	525

Berdasarkan POS Tagging akuisisi 4 didapatkan rata-rata keberadaan kata dari dataset dokumen abstrak paper sebesar 63,77%.

5. Akuisisi POS Tagging 8 (-8, +8)

Tabel 5.3-10 Analisa Kata Hasil Akuisisi POS Tagging 8 (-8, +8)

No.	Nama	Level TKT									Not in TKT
		1	2	3	4	5	6	7	8	9	
1.	IPB	300	132	217	212	155	154	93	24	105	432
2.	ITB	324	143	200	153	134	120	98	20	72	482
3.	ITS	358	146	234	231	200	166	104	34	105	558
4.	UNTAG	276	159	205	197	143	167	100	38	108	611
5.	UNAIR	285	144	214	200	122	164	83	32	109	422
6.	BINUS	269	127	216	182	123	151	69	20	88	183
7.	UB	302	160	233	210	171	172	84	20	96	440
8.	UNDIP	287	131	210	191	158	142	93	29	92	475
9.	UGM	365	163	254	259	193	177	101	32	95	661
10.	UI	339	150	268	244	166	175	114	32	96	420
11.	UII	311	182	226	203	143	156	92	32	111	497
12.	UNRAM	228	98	169	142	137	103	77	29	61	264
13.	UMS	328	159	240	223	164	186	108	37	105	444
14.	UNPAD	348	167	235	177	135	170	106	25	85	357
15.	UPH	278	143	203	152	128	143	86	26	78	311
16.	UNS	241	116	202	178	168	119	68	19	74	353

Berdasarkan POS Tagging akuisisi 8 didapatkan rata-rata keberadaan kata dari dataset dokumen abstrak paper sebesar 76,59%.

6. Akuisisi POS Tagging 10 (-10, +10)

Tabel 5.3-11 Analisa Kata Hasil Akuisisi POS Tagging 10 (-10, +10)

No.	Nama	Level TKT									Not in TKT
		1	2	3	4	5	6	7	8	9	
1.	IPB	318	135	234	247	161	156	95	27	107	344
2.	ITB	355	150	215	157	131	122	106	21	78	411
3.	ITS	375	151	242	241	198	169	113	40	105	502
4.	UNTAG	288	169	214	208	145	170	101	39	106	564
5.	UNAIR	303	161	211	202	130	164	83	32	112	377
6.	BINUS	279	132	221	181	126	156	73	20	88	152
7.	UB	320	159	243	226	181	175	83	26	101	374
8.	UNDIP	298	138	222	200	170	147	97	26	85	425
9.	UGM	388	169	260	274	203	184	104	29	105	584
10.	UI	361	158	276	260	169	180	121	31	104	344
11.	UII	334	187	233	215	149	157	97	38	119	424
12.	UNRAM	235	104	181	159	137	103	79	29	60	221
13.	UMS	345	166	256	244	170	188	109	40	101	375
14.	UNPAD	357	173	255	191	143	175	110	29	91	281
15.	UPH	296	154	208	157	134	146	87	28	88	250
16.	UNS	260	126	208	183	175	120	74	15	81	296

Berdasarkan POS Tagging akuisisi 10 didapatkan rata-rata keberadaan kata dari dataset dokumen abstrak paper sebesar 79,98%.

Dalam Tugas Akhir ini disimpulkan menggunakan akuisisi kata pada POS Tagger 1 untuk mengurangi waktu komputasi. Berdasarkan hasil analisa terdapat kesamaan pola antar pengambilan kata menggunakan akuisisi 1 hingga 10, yang berbeda adalah jumlah kata saja. Akan tetapi pola jumlah kata per corpus TKT tetap sama.

5.3.3.3 Memperkaya Corpus TKT dengan Sinonim Word

Selain menggunakan percobaan memperkaya corpus TKT dengan teknik POS Tagging, Tugas Akhir ini juga menggunakan pengembangan dari segi sinonim word. Pembahasan lebih dalam telah dijelaskan dalam bab 5.2.3.3. Penggunaan sinonim word sebagai salah satu alternative pengembangan corpus TKT dari segi corpus, bukan dari segi dokumen.

5.3.3.4 Tahap Menentukan Asumsi Label Topik Awal

Menentukan asumsi label topik awal menggunakan *Term Frequency* (TF) menjadi tahap yang paling berperan dalam metode teks mining dalam Tugas Akhir. Beberapa percobaan untuk Menentukan asumsi label topik awal seperti berikut:

1. Percobaan 1

- Dataset : Dokumen hasil preprocessing perguruan tinggi
- Corpus : Corpus TKT final dengan POS Tagging akuisisi 1
- Proses similarity : Term Frequency
- Jumlah label : 3 label teratas

Tabel 5.3-12 Label Dokumen Hasil Percobaan Teknik POS Tagging Akuisisi 1

No.	Nama	Asumsi Label Awal
1.	IPB	T6, T7, T5
2.	ITB	T6, T7, T5
3.	ITS	T6, T7, T5
4.	UNTAG	T6, T7, T5
5.	UNAIR	T6, T7, T5
6.	BINUS	T6, T7, T5
7.	UB	T6, T7, T5
8.	UNDIP	T6, T7, T5
9.	UGM	T6, T7, T5
10.	UI	T6, T7, T5

11.	UII	T6, T7, T5
12.	UNRAM	T6, T7, T5
13.	UMS	T6, T7, T5
14.	UNPAD	T6, T7, T5
15.	UPH	T6, T7, T5
16.	UNS	T6, T7, T5

2. Percobaan 2

- Dataset : Dokumen hasil preprocessing perguruan tinggi
- Corpus : Corpus TKT final tanpa menggunakan POS Tagging
- Proses similarity : Term Frequency
- Jumlah label : 3 label teratas

Tabel 5.3-13 Label Dokumen Hasil Percobaan Tanpa Menggunakan Teknik POS Tagging

No.	Nama	Asumsi Label Awal
1.	IPB	T7, T8, T9
2.	ITB	T7, T8, T9
3.	ITS	T5, T8, T9
4.	UNTAG	T7, T5, T8
5.	UNAIR	T7, T8, T9
6.	BINUS	T7, T5, T9
7.	UB	T6, T8, T9
8.	UNDIP	T6, T7, T9
9.	UGM	T7, T8, T9
10.	UI	T7, T8, T9
11.	UII	T6, T7, T9
12.	UNRAM	T7, T5, T9
13.	UMS	T7, T8, T9
14.	UNPAD	T5, T8, T9
15.	UPH	T7, T5, T9
16.	UNS	T7, T5, T8

3. Percobaan 3

- Dataset : Dokumen hasil preprocessing perguruan tinggi
- Corpus : Corpus TKT final dengan sinonim word level 1
- Proses similarity : Term Frequency
- Jumlah label : 3 label teratas

Tabel 5.3-14 Label Dokumen Hasil Percobaan Sinonim Word Level 1

No.	Nama	Asumsi Label Awal
1.	IPB	T5, T7, T9
2.	ITB	T5, T7, T9
3.	ITS	T5, T8, T9
4.	UNTAG	T5, T8, T9
5.	UNAIR	T5, T8, T9
6.	BINUS	T5, T7, T9
7.	UB	T7, T8, T9
8.	UNDIP	T5, T8, T9
9.	UGM	T7, T8, T9
10.	UI	T5, T7, T9
11.	UII	T5, T7, T9
12.	UNRAM	T3, T5, T7
13.	UMS	T5, T7, T9
14.	UNPAD	T5, T7, T9
15.	UPH	T5, T7, T9
16.	UNS	T5, T8, T9

4. Percobaan 4

- Dataset : Dokumen hasil preprocessing perguruan tinggi
- Corpus : Corpus TKT final dengan sinonim word level 2
- Proses similarity : Term Frequency
- Jumlah label : 3 label teratas

Tabel 5.3-15 Label Dokumen Hasil Percobaan Sinonim Word Level 2

No.	Nama	Asumsi Label Awal
1.	IPB	T5, T7, T9
2.	ITB	T5, T7, T9
3.	ITS	T5, T8, T9
4.	UNTAG	T5, T8, T9
5.	UNAIR	T5, T7, T9
6.	BINUS	T5, T7, T9
7.	UB	T5, T8, T9
8.	UNDIP	T5, T8, T9
9.	UGM	T5, T8, T9
10.	UI	T5, T8, T9
11.	UIN	T5, T7, T9
12.	UNRAM	T3, T5, T9
13.	UMS	T5, T7, T9
14.	UNPAD	T2, T5, T9
15.	UPH	T2, T5, T7
16.	UNS	T5, T8, T9

5. Percobaan 5

- Dataset : Dokumen hasil preprocessing perguruan tinggi
- Corpus : Corpus TKT final dengan sinonim word level 3
- Proses similarity : Term Frequency
- Jumlah label : 3 label teratas

Tabel 5.3-16 Label Dokumen Hasil Percobaan Sinonim Word Level 3

No.	Nama	Asumsi Label Awal
1.	IPB	T7, T8, T9
2.	ITB	T5, T7, T9
3.	ITS	T5, T8, T9
4.	UNTAG	T5, T8, T9
5.	UNAIR	T5, T7, T9

6.	BINUS	T5, T7, T9
7.	UB	T5, T8, T9
8.	UNDIP	T7, T8, T9
9.	UGM	T5, T8, T9
10.	UI	T5, T8, T9
11.	UII	T5, T7, T9
12.	UNRAM	T3, T7, T9
13.	UMS	T5, T7, T9
14.	UNPAD	T5, T8, T9
15.	UPH	T5, T7, T9
16.	UNS	T5, T7, T9

6. Percobaan 6

- Dataset : Dokumen hasil preprocessing perguruan tinggi
- Corpus : Corpus TKT final dengan sinonim word level all (semua sinonim digunakan)
- Proses similarity : Term Frequency
- Jumlah label : 2 label teratas

Tabel 5.3-17 Label Dokumen Hasil Percobaan Sinonim Word Level All

No.	Nama	Asumsi Label Awal
1.	IPB	T3, T5, T7
2.	ITB	T3, T5, T7
3.	ITS	T3, T5, T9
4.	UNTAG	T1, T2, T5
5.	UNAIR	T1, T5, T7
6.	BINUS	T5, T7, T9
7.	UB	T1, T2, T7
8.	UNDIP	T5, T7, T9
9.	UGM	T3, T5, T7
10.	UI	T1, T5, T7
11.	UII	T1, T3, T5
12.	UNRAM	T3, T5, T7

13.	UMS	T1, T5, T7
14.	UNPAD	T1, T5, T7
15.	UPH	T1, T2, T7
16.	UNS	T1, T5, T7

5.3.3.5 Tahap Generatif Model L-LDA

Berdasarkan percobaan pada tahap preprocessing, memperkaya corpus TKT, dan penentuan asumsi label topik awal dilakukan beberapa percobaan pada tahap L-LDA yang nantinya akan dibandingkan hasil akhir yang paling bagus didapatkan dari langkah apa saja. Berikut adalah hasil generative L-LDA dari beberapa percobaan:

1. Percobaan 1

Berdasarkan percobaan 1 dalam tahap memperkaya corpus didapatkan hasil probabilitas topik terhadap dokumen seperti pada Tabel 5.3-18.

Tabel 5.3-18 Probabilitas Label dengan Dokumen dalam Percobaan 1

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0,32	0,35	0,33	0	0
2.	ITB	0	0	0	0	0,32	0,34	0,34	0	0
3.	ITS	0	0	0	0	0,35	0,32	0,33	0	0
4.	UNTAG	0	0	0	0	0,31	0,41	0,28	0	0
5.	UNAIR	0	0	0	0	0,30	0,35	0,35	0	0
6.	BINUS	0	0	0	0	0,33	0,34	0,34	0	0
7.	UB	0	0	0	0	0,35	0,33	0,32	0	0
8.	UNDIP	0	0	0	0	0,34	0,35	0,31	0	0
9.	UGM	0	0	0	0	0,32	0,31	0,37	0	0
10.	UI	0	0	0	0	0,34	0,30	0,36	0	0
11.	UII	0	0	0	0	0,36	0,31	0,33	0	0
12.	UNRAM	0	0	0	0	0,38	0,28	0,34	0	0
13.	UMS	0	0	0	0	0,33	0,35	0,32	0	0
14.	UNPAD	0	0	0	0	0,33	0,34	0,33	0	0

15.	UPH	0	0	0	0	0,34	0,32	0,34	0	0
16.	UNS	0	0	0	0	0,37	0,30	0,33	0	0

2. Percobaan 2

Berdasarkan percobaan 2 dalam tahap memperkaya corpus didapatkan hasil probabilitas topik terhadap dokumen seperti pada Tabel 5.3-19.

Tabel 5.3-19 Probabilitas Label dengan Dokumen dalam Percobaan 2

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0	0	0,33	0,33	0,34
2.	ITB	0	0	0	0	0	0	0,31	0,34	0,35
3.	ITS	0	0	0	0	0	0	0,34	0,32	0,34
4.	UNTAG	0	0	0	0	0	0	0,35	0,32	0,33
5.	UNAIR	0	0	0	0	0	0	0,35	0,32	0,33
6.	BINUS	0	0	0	0	0,33	0	0,30	0	0,37
7.	UB	0	0	0	0	0	0,32	0	0,33	0,34
8.	UNDIP	0	0	0	0	0	0,30	0,36	0	0,34
9.	UGM	0	0	0	0	0	0	0,33	0,33	0,34
10.	UI	0	0	0	0	0	0	0,33	0,33	0,34
11.	UII	0	0	0	0	0	0,33	0,32	0	0,35
12.	UNRAM	0	0	0	0	0,31	0	0,37	0	0,32
13.	UMS	0	0	0	0	0	0	0,36	0,30	0,33
14.	UNPAD	0	0	0	0	0,35	0	0	0,32	0,33
15.	UPH	0	0	0	0	0,33	0	0,31	0	0,36
16.	UNS	0	0	0	0	0,34	0	0,30	0	0,36

3. Percobaan 3

Berdasarkan percobaan 3 dalam tahap memperkaya corpus didapatkan hasil probabilitas topik terhadap dokumen seperti pada Tabel 5.3-20.

Tabel 5.3-20 Probabilitas Label dengan Dokumen dalam Percobaan 3

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9

1.	IPB	0	0	0	0	0,32	0	0,33	0	0,35
2.	ITB	0	0	0	0	0,32	0	0,33	0	0,35
3.	ITS	0	0	0	0	0,34	0	0	0,35	0,31
4.	UNTAG	0	0	0	0	0,34	0	0	0,33	0,33
5.	UNAIR	0	0	0	0	0,31	0	0	0,39	0,30
6.	BINUS	0	0	0	0	0,33	0	0,35	0	0,32
7.	UB	0	0	0	0	0	0	0,35	0,30	0,35
8.	UNDIP	0	0	0	0	0,32	0	0	0,37	0,31
9.	UGM	0	0	0	0	0	0	0,33	0,34	0,33
10.	UI	0	0	0	0	0,30	0	0,33	0	0,37
11.	UII	0	0	0	0	0,31	0	0,36	0	0,33
12.	UNRAM	0	0	0,32	0	0,30	0	0,38	0	0
13.	UMS	0	0	0	0	0,34	0	0,34	0	0,32
14.	UNPAD	0	0	0	0	0,32	0	0,34	0	0,34
15.	UPH	0	0	0	0	0,32	0	0,36	0	0,32
16.	UNS	0	0	0	0	0,37	0	0	0,29	0,34

4. Percobaan 4

Berdasarkan percobaan 4 dalam tahap memperkaya corpus didapatkan hasil probabilitas topik terhadap dokumen seperti pada Tabel 5.3-21.

Tabel 5.3-21 Probabilitas Label dengan Dokumen dalam Percobaan 4

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0,33	0	0,34	0	0,33
2.	ITB	0	0	0	0	0,33	0	0,32	0	0,35
3.	ITS	0	0	0	0	0,34	0	0	0,35	0,31
4.	UNTAG	0	0	0	0	0,34	0	0	0,32	0,34
5.	UNAIR	0	0	0	0	0,34	0	0,30	0	0,36
6.	BINUS	0	0	0	0	0,36	0	0,35	0	0,29
7.	UB	0	0	0	0	0,35	0	0	0,33	0,32
8.	UNDIP	0	0	0	0	0,36	0	0	0,32	0,32
9.	UGM	0	0	0	0	0,31	0	0	0,37	0,32
10.	UI	0	0	0	0	0,34	0	0	0,32	0,34
11.	UII	0	0	0	0	0,36	0	0,33	0	0,31
12.	UNRAM	0	0	0,31	0	0,39	0	0	0	0,30

13.	UMS	0	0	0	0	0,35	0	0,32	0	0,33
14.	UNPAD	0	0,31	0	0	0,36	0	0	0	0,34
15.	UPH	0	0,37	0	0	0,30	0	0,37	0	0
16.	UNS	0	0	0	0	0,36	0	0	0,33	0,31

5. Percobaan 5

Berdasarkan percobaan 5 dalam tahap memperkaya corpus didapatkan hasil probabilitas topik terhadap dokumen seperti pada Tabel 5.3-22.

Tabel 5.3-22 Probabilitas Label dengan Dokumen dalam Percobaan 5

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0	0	0,34	0,34	0,32
2.	ITB	0	0	0	0	0,32	0	0,32	0	0,36
3.	ITS	0	0	0	0	0,36	0	0	0,30	0,34
4.	UNTAG	0	0	0	0	0,36	0	0	0,32	0,32
5.	UNAIR	0	0	0	0	0,33	0	0,30	0	0,37
6.	BINUS	0	0	0	0	0,32	0	0,36	0	0,32
7.	UB	0	0	0	0	0,34	0	0	0,32	0,34
8.	UNDIP	0	0	0	0	0	0	0,32	0,35	0,33
9.	UGM	0	0	0	0	0,32	0	0	0,37	0,31
10.	UI	0	0	0	0	0,32	0	0	0,33	0,35
11.	UII	0	0	0	0	0,34	0	0	0,34	0,33
12.	UNRAM	0	0	0,35	0	0	0	0,34	0	0,31
13.	UMS	0	0	0	0	0,31	0	0,36	0	0,33
14.	UNPAD	0	0	0	0	0,37	0	0	0,31	0,32
15.	UPH	0	0	0	0	0,35	0	0,30	0	0,35
16.	UNS	0	0	0	0	0,37	0	0,30	0	0,33

6. Percobaan 6

Berdasarkan percobaan 6 dalam tahap memperkaya corpus didapatkan 3 asumsi label awal, dari label ini dilakukan percobaan lain yaitu dengan mengambil 2 label dari level paling tinggi. Sehingga pada percobaan 6 ini, probabilitas

topik terhadap dokumen hanya berasal dari 2 label saja seperti pada Tabel 5.3-23.

Tabel 5.3-23 Probabilitas Label dengan Dokumen dalam Percobaan 6

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0,50	0	0,50	0	0
2.	ITB	0	0	0	0	0,48	0	0,52	0	0
3.	ITS	0	0	0	0	0,53	0	0	0	0,47
4.	UNTAG	0	0,50	0	0	0,50	0	0	0	0
5.	UNAIR	0	0	0	0	0,47	0	0,53	0	0
6.	BINUS	0	0	0	0	0	0	0,52	0	0,48
7.	UB	0	0,49	0	0	0	0	0,51	0	0
8.	UNDIP	0	0	0	0	0	0	0,51	0	0,49
9.	UGM	0	0	0	0	0,47	0	0,53	0	0
10.	UI	0	0	0	0	0,47	0	0,53	0	0
11.	UII	0	0	0,49	0	0,51	0	0	0	0
12.	UNRAM	0	0	0	0	0,53	0	0,47	0	0
13.	UMS	0	0	0	0	0,50	0	0,50	0	0
14.	UNPAD	0	0	0	0	0,49	0	0,51	0	0
15.	UPH	0	0,48	0	0	0	0	0,52	0	0
16.	UNS	0	0	0	0	0,47	0	0,53	0	0

5.3.3.6 Tahap Pembobotan Final Pemeringkatan

Hasil keluaran dari tahap Generatif Model L-LDA akan dihitung menggunakan bobot yang sudah ditentukan. Dengan rincian setiap percobaan sebagai berikut.

1. Percobaan 1

Tabel 5.3-24 Topik Akhir Dokumen Menggunakan Percobaan Teknik POS Tagging Akuisisi 1

No.	Nama	Label Topik AKhir
1.	IPB	T6
2.	ITB	T6
3.	ITS	T5

4.	UNTAG	T6
5.	UNAIR	T7
6.	BINUS	T6
7.	UB	T5
8.	UNDIP	T6
9.	UGM	T7
10.	UI	T7
11.	UII	T5
12.	UNRAM	T5
13.	UMS	T6
14.	UNPAD	T7
15.	UPH	T5
16.	UNS	T5

2. Percobaan 2

Tabel 5.3-25 Topik Akhir Dokumen Menggunakan Percobaan Tanpa Teknik POS Tagging

No.	Nama	Label Topik AKhir
1.	IPB	T9
2.	ITB	T7
3.	ITS	T5
4.	UNTAG	T5
5.	UNAIR	T9
6.	BINUS	T7
7.	UB	T6
8.	UNDIP	T9
9.	UGM	T8
10.	UI	T8
11.	UII	T6
12.	UNRAM	T9
13.	UMS	T8
14.	UNPAD	T9
15.	UPH	T7
16.	UNS	T5

3. Percobaan 3

Tabel 5.3-26 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level 1

No.	Nama	Label Topik AKhir
1.	IPB	T9
2.	ITB	T9
3.	ITS	T8
4.	UNTAG	T8
5.	UNAIR	T8
6.	BINUS	T7
7.	UB	T9
8.	UNDIP	T8
9.	UGM	T8
10.	UI	T9
11.	UIN	T7
12.	UNRAM	T7
13.	UMS	T7
14.	UNPAD	T9
15.	UPH	T7
16.	UNS	T8

4. Percobaan 4

Tabel 5.3-27 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level 2

No.	Nama	Label Topik AKhir
1.	IPB	T7
2.	ITB	T7
3.	ITS	T8
4.	UNTAG	T5
5.	UNAIR	T9
6.	BINUS	T5
7.	UB	T5
8.	UNDIP	T5

9.	UGM	T8
10.	UI	T5
11.	UII	T9
12.	UNRAM	T5
13.	UMS	T5
14.	UNPAD	T9
15.	UPH	T2
16.	UNS	T5

5. Percobaan 5

Tabel 5.3-28 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level 3

No.	Nama	Label Topik AKhir
1.	IPB	T8
2.	ITB	T9
3.	ITS	T5
4.	UNTAG	T5
5.	UNAIR	T5
6.	BINUS	T7
7.	UB	T9
8.	UNDIP	T9
9.	UGM	T8
10.	UI	T9
11.	UII	T5
12.	UNRAM	T3
13.	UMS	T7
14.	UNPAD	T5
15.	UPH	T5
16.	UNS	T5

6. Percobaan 6

Tabel 5.3-29 Topik Akhir Dokumen Menggunakan Percobaan Sinonim Word Level All

No.	Nama	Label Topik AKhir
1.	IPB	T7
2.	ITB	T7
3.	ITS	T5
4.	UNTAG	T2
5.	UNAIR	T7
6.	BINUS	T7
7.	UB	T7
8.	UNDIP	T7
9.	UGM	T7
10.	UI	T7
11.	UII	T5
12.	UNRAM	T5
13.	UMS	T7
14.	UNPAD	T7
15.	UPH	T7
16.	UNS	T7

5.3.3.7 Menghitung Distance Hasil Sistem dengan Ground Truth QS World University Rankings 2016-2017

Distance dalam hal ini adalah beda peringkat hasil sistem dengan peringkat yang ada pada *Ground Truth*. Semakin kecil distance maka semakin bagus peringkat sistem yang dihasilkan. Berikut adalah analisa distance dari 6 percobaan yang sudah dilakukan diatas.

1. Percobaan 1

Tabel 5.3-30 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 1

Nama	Peringkat Ground Truth	Peringkat Sistem	Beda	Flag Toleransi (<=3)
------	------------------------	------------------	------	----------------------

IPB	5	3	2	1
ITB	1	1	0	1
ITS	7	9	2	1
UB	9	7	2	1
UGM	3	2	1	1
UI	2	4	2	1
UMS	8	6	2	1
UNAIR	4	5	1	1
UNDIP	6	8	2	1
Total Distance			14	9

2. Percobaan 2

Tabel 5.3-31 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 2

Nama	Peringkat Ground Truth	Peringkat Sistem	Beda	Flag Toleransi (≤ 3)
IPB	5	5	0	1
ITB	1	4	3	1
ITS	7	8	1	1
UB	9	9	0	1
UGM	3	2	1	1
UI	2	3	1	1
UMS	8	6	2	1
UNAIR	4	1	3	1
UNDIP	6	7	1	1
Total Distance			12	9

3. Percobaan 3

Tabel 5.3-32 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 3

Nama	Peringkat Ground Truth	Peringkat Sistem	Beda	Flag Toleransi (≤ 3)
IPB	5	8	3	1

ITB	1	7	6	0
ITS	7	5	2	1
UB	9	1	8	0
UGM	3	2	1	1
UI	2	6	4	0
UMS	8	9	1	1
UNAIR	4	3	1	1
UNDIP	6	4	2	1
Total Distance			28	6

4. Percobaan 4

Tabel 5.3-33 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 4

Nama	Peringkat Ground Truth	Peringkat Sistem	Beda	Flag Toleransi (≤ 3)
IPB	5	8	3	1
ITB	1	7	6	0
ITS	7	3	4	0
UB	9	4	5	0
UGM	3	1	2	1
UI	2	2	0	1
UMS	8	9	1	1
UNAIR	4	6	2	1
UNDIP	6	5	1	1
Total Distance			24	6

5. Percobaan 5

Tabel 5.3-34 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 5

Nama	Peringkat Ground Truth	Peringkat Sistem	Beda	Flag Toleransi (≤ 3)
IPB	5	2	3	1
ITB	1	8	7	0

ITS	7	6	1	1
UB	9	5	4	0
UGM	3	4	1	1
UI	2	3	1	1
UMS	8	9	1	1
UNAIR	4	7	3	1
UNDIP	6	1	5	1
Total Distance			26	7

6. Percobaan 6

Tabel 5.3-35 Perbandingan Peringkat Ground Truth QS World University Rankings 2016-2017 dengan Hasil Percobaan 6

Nama	Peringkat Ground Truth	Peringkat Sistem	Beda	Flag Toleransi (≤ 3)
IPB	5	7	2	1
ITB	1	6	5	0
ITS	7	2	5	0
UB	9	9	0	1
UGM	3	5	2	1
UI	2	3	1	1
UMS	8	8	0	1
UNAIR	4	4	0	1
UNDIP	6	1	5	0
Total Distance			20	6

5.3.3.8 Perbandingan Hasil Analisa Masing-masing Percobaan dengan Ground Truth QS World University Rankings 2015-2016

Tabel 5.3-36 Hasil Analisa Semua Percobaan dengan Ground Truth QS World University Rankings 2015-2016

Lingkup	Indikator Analisa	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5	Percobaan 6
Indikator Reputasi Akademik	Distance (beda)	32	22	62	60	50	76
	Toleransi	13/16	16/16	8/16	9/16	10/16	4/16
	Prosentase toleransi	81,25%	100%	50%	56,25%	62,5%	25%
	Similaritay Pearson Correlation	82,9%	93,8%	38,5%	47,6%	62%	27,3%
	Analisa Uji Table T (kepercayaan 95%)	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima
	Analisa Uji Table T (kepercayaan 60%)	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 ditolak
Sistem Keseluruhan	Distance (beda)	16	18	22	22	20	28
	Toleransi	10/11	10/11	9/11	9/11	9/11	7/11
	Prosentase toleransi	90,91%	90,91%	81,82%	81,82%	81,82%	63,63%
	Similaritay Pearson Correlation	79,09%	77,27%	65,45%	69,1%	73,63%	55,45%

5.3.3.9 Perbandingan Hasil Analisa Masing-masing Percobaan dengan Ground Truth QS World University Rankings 2016-2017

Tabel 5.3-37 Hasil Analisa Semua Percobaan dengan Ground Truth QS World University Rankings 2016-2017

Lingkup	Indikator Analisa	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5	Percobaan 6
Indikator Reputasi Akademik	Distance (beda)	14	12	28	24	26	20
	Toleransi	9/9	9/9	6/9	6/9	7/9	6/9
	Prosentase toleransi	100%	100%	66,67%	66,67%	77,78%	66,67%
	Similaritay Pearson Correlation	78,3%	78,3%	13,3%	20%	6,6%	30%
	Analisa Uji Table T (kepercayaan 95%)	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima
	Analisa Uji Table T (kepercayaan 60%)	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 ditolak	h0 diterima
Sistem Keseluruhan	Distance (beda)	12	12	14	12	12	12
	Toleransi	8/9	8/9	7/9	8/9	8/9	9/9
	Prosentase toleransi	88,89%	88,89%	77,78%	88,89%	88,89%	100%
	Similaritay Pearson Correlation	78,33%	78,33%	60%	78,33%	78,33%	78,33%

5.3.3.10 Perbandingan Hasil Analisa Masing-masing Percobaan dengan Ground Truth QS World University Rankings 2017-2018

Tabel 5.3-38 Hasil Analisa Semua Percobaan dengan Ground Truth QS World University Rankings 2017-2018

Lingkup	Indikator Analisa	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5	Percobaan 6
Indikator Reputasi Akademik	Distance (beda)	12	10	26	24	26	20
	Toleransi	9/9	9/9	7/9	6/9	7/9	6/9
	Prosentase toleransi	100%	100%	77,78%	66,67%	77,78%	66,67%
	Similaritay Pearson Correlation	81,6%	80%	6,6%	21,6%	5%	26,67%
	Analisa Uji Table T (kepercayaan 95%)	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 diterima
	Analisa Uji Table T (kepercayaan 60%)	h0 diterima	h0 diterima	h0 diterima	h0 diterima	h0 ditolak	h0 diterima
Sistem Keseluruhan	Distance (beda)	12	12	14	12	12	12
	Toleransi	8/9	8/9	7/9	8/9	8/9	9/9
	Prosentase toleransi	88,89%	88,89%	77,78%	88,89%	88,89%	100%
	Similaritay Pearson Correlation	78,33%	78,33%	68,32%	78,33%	78,33%	78,33%

5.3.4 Skenario Pengujian Sitem Berdasarkan Hasil Pembobotan Akhir Sesuai *Ground Truth QS Worlds University Rankings*

5.3.4.1 Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan QS Metode L-LDA

- Aktor : pengguna aplikasi website
- Langkah :
 1. Pengguna membuka aplikasi website perangkat lunak.
 2. Menuju halaman “Lihat Ranking” dengan scroll page website kebawah.
 3. Mengisi bobot sesuai tipe pemeringkatan L-LDA, bobot bisa menggunakan konfigurasi default maupun dapat dirubah sesuai kebutuhan seperti Gambar 5.3.4.1.1.
 4. Pilih tombol “Cek QS L-LDA” seperti Gambar 5.3.4.1.2.
 5. Sistem kan menampilkan hasil pemeringkatan sesuai tipe QS L-LDA seperti Gambar 5.3-1.

Pembobotan Pemeringkatan									
Indikator	QS General	QS Asia	Webometrics	4ICU	DIKTI	QS (L-LDA)	QS (PLSA)	QS (Helmholtz)	QS (Adaboost)
Reputasi Akademik	40	30	0	0	0	26	0	0	0
Reputasi Employer	10	20	0	0	0	14	0	0	0
Jumlah Mahasiswa Aktif	20	15	0	0	0	5	0	0	0
Jumlah Sitasi Akademisi	20	10	0	0	0	15	0	0	0
Jumlah Tenaga Pengajar	5	2.5	0	0	0	5	0	0	0
Kualitas Sitasi (H-index & I10-index)	5	2.5	0	0	0	3	0	0	0
Jumlah Phd (S3)	0	5	0	0	0	20	0	0	0

Gambar 5.3-1 Antarmuka Pembobotan Peringkat

Jumlah Total	100	100	100	100	100	100	100	100	100
	Cek QS-General	Cek QS-Asia	Cek Webometrics	Cek 4CU	Cek DIKT1	Cek QS L-LDA	Cek QS PLSA	Cek QS Heilmholtz	Cek QS Adaboost

Gambar 5.3-2 Antarmuka Tombol Cek Pemingkatan

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Universitas Indonesia	0.9085	Depok	Jawa Barat
2	Institut Teknologi Bandung	0.7978	Bandung	Jawa Barat
3	Universitas Gadjah Mada	0.7952	Yogyakarta	DI Yogyakarta
4	Institut Pertanian Bogor	0.6802	Bogor	Jawa Barat
5	Universitas Airlangga	0.6178	Surabaya	Jawa Timur
6	Universitas Brawijaya	0.6117	Malang	Jawa Timur
7	Universitas Diponegoro	0.6014	Semarang	Jawa Tengah
8	Universitas Hasanuddin	0.5956	Makassar	Sulawesi Selatan
9	Universitas Bina Nusantara	0.5939	Jakarta	DKI Jakarta
10	Universitas Padjadjaran	0.575	Bandung	Jawa Barat
11	Institut Teknologi Sepuluh Nopember	0.5666	Surabaya	Jawa Timur
12	Universitas Udayana	0.5054	Badung	Bali

Gambar 5.3-3 Antarmuka Hasil Pemingkatan

Berdasarkan skenario pengujian kinerja sistem terhadap *Ground Truth QS World University Ranking* didapatkan nilai similaritas peringkat sebesar sebesar 78.33%.

5.4 Evaluasi

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.4.1. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

Tabel 5.4-1 Rangkuman Hasil Pengujian

ID	Nama	Hasil Uji Coba
UJ-UC-001	Memasukkan presentase bobot kriteria penilaian pemingkatan	Berhasil

UJ-UC-002	Meihat peringkat perguruan tinggi berdasar tipe pemeringkatan	Berhasil
UJ-UC-003	Mencari perguruan tinggi berdasarkan potongan nama	Berhasil
UJ-UC-004	Mencari perguruan tinggi berdasarkan region wilayah provinsi	Berhasil
UJ-UC-005	Melihat detail informasi utama perguruan tinggi	Berhasil

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Dari percobaan yang dilakukan dalam penggunaan corpus *keyword* dalam membantu metode L-LDA untuk menghasilkan pencocokan asumsi label topik, dihasilkan percobaan terbaik pada percobaan 6 dengan rincian kinerja sistem sebagai berikut, gap distance peringkat sebesar 20, dan peringkat dalam toleransi sebesar 66,67%.
2. Kinerja sistem secara keseluruhan dapat menyamai *Ground Truth QS World University Rankings 2016-2017*, dengan menggunakan percobaan 6 yaitu menggunakan Sinonim Word (WordNet) level all dengan rincian sebagai berikut, gap distance peringkat sebesar 12, peringkat dalam toleransi sebesar 100%, dan similarity peringkat sebesar 78,33%.
3. Penggunaan teks mining menggunakan metode L-LDA dapat menggantikan penentuan kriteria reputasi akademik yang sebelumnya bersifat manual.
4. Metode L-LDA dapat dikembangkan lagi menggunakan teknik POS Tagging dan Sinonim Word (WordNet) untuk memperkaya corpus TKT yang digunakan sebagai acuan kelas topik dokumen.
5. Memperkaya corpus dapat menghasilkan hasil yang baik jika proses breakdown dan pengembangannya tepat. Metode analisa manual lebih akurat dalam menghasilkan pencocokan term sebagai asumsi label topik awal dokumen

6. Penggunaan data berbasis SINTA dalam Tugas Akhir ini kurang valid, karena data yang digunakan tidak diperbaharui secara berkala oleh pihak SINTA.
7. Kombinasi kriteria pemeringkatan yang bersifat kualitatif dan kuantitatif dapat menghasilkan hasil pemeringkatan yang mendekati Ground Truth sistem yaitu QS University Rankings 2016-2017.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Melakukan penambahan kriteria penilaian pemeringkatan dapat dikondisikan dengan kebutuhan agar menghasilkan keragaman kriteria penilaian dan tidak bertumpu pada satu kriteria utama saja.
2. Menambahkan dan/atau memperbaiki cara dalam memperkaya corpus dengan POS Tagging atau Sinonim Word (WordNet) dapat meningkatkan koleksi keyword dalam corpus.
3. Penggunaan otomatisasi dalam proses menentukan asumsi label awal harus diperhatikan. Lebih sesuai menggunakan analisa manual daripada melakukan otomatisasi menggunakan pencocokan *Term Frequency* (TF).
4. Mengurangi *running time* untuk proses dengan data yang sangat besar terutama pada bagaian grabbing data online dan query data dari database.
5. Pengembangan perangkat lunak kedepannya disarankan menggunakan data untuk kriteria Jumlah Sitasi dan kualitas Sitasi maupun data yang bersifat kuantitatif terhadap akademisi menggunakan data dari Scopus, karena data yang disediakan telah lengkap dan diperbarui secara berkala.

DAFTAR PUSTAKA

- [1] K.S. Reddy, En Xie, Qingqing Tang, "Higher education, high-impact research, and world university rankings: A Case of India and comparison with China," *Pacific Science Review B: Humanities and Social Sciences* 2, pp. 1-21, 2016.
- [2] Tatiana Sidorenko, Tatiana Gorbatova, "Efficiency of Russian education through the scale of," *Procedia - Social and Behavioral Sciences* 166, pp. 464-467, 2015.
- [3] Mu-Hsuan Huang, "Opening The Black Box of QS World University Rankings," *Research Evaluation* 21, pp. 71-78, 2012.
- [4] QS World University Rankings, "QS Top University," 1 January 2016. [Online]. Available: <http://www.topuniversities.com/university-rankings/asian-university-rankings/2016>. [Diakses 10 December 2016].
- [5] Kementerian, Riset Teknologi dan Pendidikan Tinggi, "Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia Nomor 42 Tahun 2016 Tentang Pengukuran dan Penetapan Tingkat Kesiapterapan Teknologi," Kementerian Riset, Teknologi dan Pendidikan Tinggi, Jakarta, 2016.
- [6] Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," 2009.
- [7] Wikipedia, "Wikipedia College and University Rankings," 14 December 2016. [Online]. Available: https://en.wikipedia.org/wiki/College_and_university_rankings. [Diakses 18 December 2016].
- [8] Adina - Petruta Pavel, "Global University Rankings - A Comparative Analysis," *Procedia Economics and Finance* 26, pp. 54-63, 2015.
- [9] QS World University Rankings, "QS Global Academic Survey 2014," QS World University Rankings, 2014.
- [10] Vidya Rajiv Yeravdekar, Gauri Tiwari, "Global Rankings of Higher Education Institutions and India's Effective Non-Precense: Why Have World-Class Universities Eluded The Indian Higher Education System? And, How Worthwhile is The Indian Government's Captivation to Launch World Class Universities," *Procedia - Social and Behavioral Sciences* 157, pp. 63-83, 2014.

- [11] QS World University Rankings, "Annual QS Global Employer Survey," QS World University Rankings, 2014.
- [12] L. R. Raymond J. Mooney, "Content-based book recommending using learning for text categorization," dalam *DL '00 Proceedings of the fifth ACM conference on Digital libraries*, San Antonio, Texas, USA, 2000.
- [13] Wikipedia, "Wikipedia Scopus," 4 August 2016. [Online]. Available: <https://en.wikipedia.org/wiki/Scopus>. [Diakses 10 December 2016].
- [14] Elsevier, "Elsevier Developers," 1 January 2016. [Online]. Available: <http://dev.elsevier.com/>. [Diakses 10 December 2016].
- [15] Anne-Wil K. Harzing, Ron van der Wal, "Google Scholar as A New Source for Citation Analysis," *Ethics in Science and Environmental Politics*, pp. 61-73, 2008.
- [16] Wikipedia, "Wikipedia Google Scholar," 27 June 2017. [Online]. Available: https://en.wikipedia.org/wiki/Google_Scholar. [Diakses 17 December 2016].
- [17] Wikipedia, "Wikipedia H-Index," 28 July 2017. [Online]. Available: <https://en.wikipedia.org/wiki/H-index>. [Diakses 28 July 2017].
- [18] Kementerian Riset, Teknologi dan Pendidikan Tinggi, "Profil, Fitur, Registrasi Sinta Pusat Index, Sitasi dan Kepakaran Terbesar di Indonesia," Kementerian Riset, Teknologi dan Pendidikan Tinggi, Jakarta, 2017.
- [19] Kementerian Riset, teknologi dan Pendidikan Tinggi, "Sinta Science and Technology Index," Kementerian Riset Teknologi dan Pendidikan Tinggi, 28 July 2017. [Online]. Available: <http://sintal.ristekdikti.go.id/>. [Diakses 28 July 2017].
- [20] Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia, "Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia," 1 January 2016. [Online]. Available: <http://www.forlap.dikti.go.id/>. [Diakses 10 December 2016].
- [21] Arjun Thakur, A. L. Sangal, Harminder Bindra, "Quantitative Measurement and Comparison of Effects of Various SearchEngine Optimization Parameters on Alexa Traffic Rank," *International Journal of Computer Applications*, pp. 975-8887, 2011.
- [22] Wikipedia, "Wikipedia Python Programming Language," 28 July

2017. [Online]. Available: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)). [Diakses 28 July 2017].
- [23] Steven Bird, "NLTK: The Natural Language Toolkit," *Proceedings of the COLINGACL on Interactive presentation*, pp. 69-72, 2006.
- [24] Tingting Wei, Yonghe Lu, Houyou Chang, Qiang Zhou, Xianyu Bao, "A Semantic Approach for Text Clustering using WordNet and Lexical Chains," *Expert Systems with Applications* 42, vol. 4, no. 1, pp. 2264-2275, 2015.
- [25] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research* 12, pp. 2825-2830, 2011.
- [26] TextBlob, "TextBlob: Simplified Text Processing," 28 June 2017. [Online]. Available: <https://textblob.readthedocs.io/en/dev/>. [Diakses 28 June 2017].
- [27] Wikipedia, "Wikipedia PHP," 28 July 2017. [Online]. Available: <https://en.wikipedia.org/wiki/PHP>. [Diakses 28 July 2017].
- [28] Wikipedia, "Wikipedia MySQL," 28 July 2017. [Online]. Available: <https://en.wikipedia.org/wiki/MySQL>. [Diakses 28 July 2017].
- [29] Daniel Jurafsky, James H. Martin, "Part-of-Speech Tagging," dalam *Speech and Language Processing*, 2016.
- [30] Center of Teaching and Learning Stanford University, "Bloom's Taxonomy of Educational Objectives," Stanford University, California, 1981.
- [31] David M. Blei, Andrew Y. Ng, Michael I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research* 20, pp. 993-1022, 2003.
- [32] Zulhanif, "Pemodelan Topik dengan Latent Dirichlet Allocation," 2016.
- [33] William M. Darling, "A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling," 1 December 2011.
- [34] Andreyan Rizky Baskara, Riyanarto Sarno, Adhatus Solichah, "Discovering Traceability between Business Process and Software Component using Latent Dirichlet Allocation," 2016.

- [35] David M. Blei, Jon D. McAuliffe, “Supervised Topic Model,” *Advances in Neural Information Processing Systems*, 2007.
- [36] Simon Lacoste-Julien, Fei Sha, Michael I. Jordan, “DiscLDA Discriminative Learning for Dimensionality Reduction and Classification,” *Advances in Neural Information Processing Systems 21*, 2008.
- [37] Nikhil Johri, Daniel Ramage, Daniel A. McFarland, Daniel Jurafsky, “A Study of Academic Collaboration in Computational Linguistics with Latent Mixtures of Authors,” *LaTeCH '11 Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pp. 124-132, 2011.
- [38] Juan Ramos, “Using TF-IDF to Determine Word Relevance in Document Queries,” 2003.
- [39] Badan Akreditasi Nasional Perguruan Tinggi, “Pedoman Penyusunan Borang Akreditasi Institusi Perguruan Tinggi,” Badan Akreditasi Nasional Perguruan Tinggi, Jakarta, 2011.

LAMPIRAN

1. Kode Sumber Text Mining Metode L-LDA

1.1. Kode Sumber Text Preprocessing

```
import sys
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize,
sent_tokenize, RegexpTokenizer
import re, string, nltk, os, math, random, time
from textblob import TextBlob

#fungsi sorting str+number (campuran) tipe karakter
def atoi(text):
    return int(text) if text.isdigit() else text

def natural_keys(text):
    return [atoi(c) for c in re.split('(\\d+)', text)]

def perkaya_corpus_tkt(tmp_result,
list_update_tkt_all):
    openfile_tkt1 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT1.txt", "r")
    content_tkt1 = openfile_tkt1.read()
    content_tkt1 = content_tkt1.split()
    openfile_tkt1.close()
    # list_update_tkt_all.append(content_tkt1)

openfile_tkt2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT2.txt", "r")
    content_tkt2 = openfile_tkt2.read()
    content_tkt2 = content_tkt2.split()
    openfile_tkt2.close()

openfile_tkt3 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT3.txt", "r")
    content_tkt3 = openfile_tkt3.read()
    content_tkt3 = content_tkt3.split()
    openfile_tkt3.close()
```

```
openfile_tkt4 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT4.txt", "r")
    content_tkt4 = openfile_tkt4.read()
    content_tkt4 = content_tkt4.split()
    openfile_tkt4.close()

openfile_tkt5 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT5.txt", "r")
    content_tkt5 = openfile_tkt5.read()
    content_tkt5 = content_tkt5.split()
    openfile_tkt5.close()

openfile_tkt6 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT6.txt", "r")
    content_tkt6 = openfile_tkt6.read()
    content_tkt6 = content_tkt6.split()
    openfile_tkt6.close()

openfile_tkt7 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT7.txt", "r")
    content_tkt7 = openfile_tkt7.read()
    content_tkt7 = content_tkt7.split()
    openfile_tkt7.close()

openfile_tkt8 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT8.txt", "r")
    content_tkt8 = openfile_tkt8.read()
    content_tkt8 = content_tkt8.split()
    openfile_tkt8.close()

openfile_tkt9 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT9.txt", "r")
    content_tkt9 = openfile_tkt9.read()
    content_tkt9 = content_tkt9.split()
    openfile_tkt9.close()
```

```

tmp_result = tmp_result.split()
tmp_word = ""
listWord = []
lw = {}
tmp_result_2 = ""

for word in tmp_result:
    if word in lw:
        lw[word] += 1;
    else:
        lw[word] = 1;
        tmp_result_2 += word + " "

listWord.append(lw)
tmp_result_2_final = tmp_result_2.split()

index=0
for word in tmp_result_2_final:

    if word in content_tkt1:
        text_postagger = nltk.word_tokenize(word)
        pos = nltk.pos_tag(text_postagger)
        list_update_tkt_all.append(word)

        for i in range(0, len(pos)):
            jos = pos[i][1]
            if jos=='NN':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='NNS':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='NNP':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

```

```

elif jos=='NNPS':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='VBD':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='VBG':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='VBN':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='VBP':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

elif jos=='VBZ':
list_update_tkt_all.append(tmp_result_2_final[index-
1])
list_update_tkt_all.append(tmp_result_2_final[index+1]
)

if word in content_tkt2:
    text_postagger = nltk.word_tokenize(word)
    pos = nltk.pos_tag(text_postagger)
    list_update_tkt_all.append(word)

    ...

index+=1

```

```
def set_label(tmp_result):
    sys.stderr.write("JUMLAH KATA DALAM DOKUMEN:
"+str(len(tmp_result))+"\n")

    openfile_tkt1 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT1.txt",
"r")
    content_tkt1 = openfile_tkt1.read()
    openfile_tkt1.close()

    openfile_tkt2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT2.txt",
"r")
    content_tkt2 = openfile_tkt2.read()
    openfile_tkt2.close()

    openfile_tkt3 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT3.txt",
"r")
    content_tkt3 = openfile_tkt3.read()
    openfile_tkt3.close()

    openfile_tkt4 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT4.txt",
"r")
    content_tkt4 = openfile_tkt4.read()
    openfile_tkt4.close()

    openfile_tkt5 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT5.txt",
"r")
    content_tkt5 = openfile_tkt5.read()
    openfile_tkt5.close()

    openfile_tkt6 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT6.txt",
"r")
    content_tkt6 = openfile_tkt6.read()
    openfile_tkt6.close()
```

```

openfile_tkt7 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT7.txt",
"r")
    content_tkt7 = openfile_tkt7.read()
    openfile_tkt7.close()

    openfile_tkt8 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT8.txt",
"r")
    content_tkt8 = openfile_tkt8.read()
    openfile_tkt8.close()

    openfile_tkt9 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_Update/update_TKT9.txt",
"r")
    content_tkt9 = openfile_tkt9.read()
    openfile_tkt9.close()

    openfile_tkt_all =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_All/tampung_TKT.txt", "r")
    content_tkt_all = openfile_tkt_all.read()
    openfile_tkt_all.close()

counter1=counter2=counter3=counter4=counter5=counter6=
counter7=counter8=counter9=counter_all=0

    tmp_result = tmp_result.split()
    tmp_word = ""

#hitung frekuensi kemunculan kata
listWord = []
lw = {}
tmp_result_2 = ""
for word in tmp_result:
    if word in lw:
        lw[word] += 1;
    else:
        lw[word] = 1;
        tmp_result_2 += word + " "
listWord.append(lw)

```



```

tmp_result_2_final = tmp_result_2.split()

for word in tmp_result_2_final:
    if word in content_tkt1:
        counter1 += 1
    if word in content_tkt2:
        counter2 += 1
    if word in content_tkt3:
        counter3 += 1
    if word in content_tkt4:
        counter4 += 1
    if word in content_tkt5:
        counter5 += 1
    if word in content_tkt6:
        counter6 += 1
    if word in content_tkt7:
        counter7 += 1
    if word in content_tkt8:
        counter8 += 1
    if word in content_tkt9:
        counter9 += 1

list_label = [counter1, counter2, counter3, counter4,
              counter5, counter6, counter7, counter8, counter9]
list_nama_label =
["T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"]

tmp_lnl = {}
a=0
for lnl in list_nama_label:
    tmp_lnl[a] = lnl
    a+=1

tmp_lb = {}
b=0
p = ""
simpan = []

for lb in list_label:
    if lb>0:
        tmp_lb[b] = lb
        simpan.append(str(tmp_lb[b])+"-
"+str(tmp_lnl[b]))
        p += tmp_lnl[b] + ","
        b+=1

```

```

a = sorted(simpan, key=natural_keys, reverse=True)
sys.stderr.write("SORTING HASIL TF (pencocokan
berdasarkan term): "+str(a)+"\n")

ambil_4_label_teratas = a[:4]
ambil_4_label_teratas = '
'.join(ambil_4_label_teratas)

b = ambil_4_label_teratas.split()

c = {}
d=0
e=0
f = {}
label_final = ""
for j in b:
    d+=1
    c[d] = j[-2:]

for k in b:
    label_final += str(c[d-e])+"",
    e+=1

label_final = label_final[:-1]
label_final_akhir = "["+label_final+"]"
label_final_akhir = re.sub('-', ' ', label_final_akhir)

return label_final_akhir

```

```

def inggris(dir3, nama_universitas, dir2_input, lala2,
listTF, tipe_bahasa, list_update_tkt1,
list_update_tkt2, list_update_tkt3, list_update_tkt4,
list_update_tkt5, list_update_tkt6, list_update_tkt7,
list_update_tkt8, list_update_tkt9,
list_update_tkt_all):
    tipe_bahasa = " - TIPE INGGRIS"
    nama_file = "Dokumen_"

    # 1 tokenisasi & remove punctuation
    file_input = dir2_input.lower().split()

    punctuation = re.compile(r'^\w.'])
    word_list_punctuation = [punctuation.sub(" ",
word) for word in file_input]
    tmp_word_list_punctuation = [x.strip(' ') for x in
word_list_punctuation]

    punctuation2 = re.compile(r'[-.?!,":;()|0-9]')
    word_list_punctuation2 = [punctuation2.sub("",
word) for word in tmp_word_list_punctuation]
    tmp_word_list_punctuation2 = [x.strip(' ') for x
in word_list_punctuation2]

    # 2 stopwords & stemming
    snowball_stemmer = SnowballStemmer("english")
    stop_words = {}
    stop_words = set(stopwords.words("english"))

    words_list = []
    for i in tmp_word_list_punctuation2:
        if i not in stop_words:
            words_list.append(i)

    listResult = []
    for word in words_list:
        listResult.append(word)

tf = {}
for word in listResult:
    if word in tf:
        tf[word] += 1;
    else:
        tf[word] = 1;

```

```

listTF.append(tf)

    tmp_result = ""

    for word in listResult:
        if len(word) > 2:
            tmp_result += word + " "

    jumlah_kata_hasil_preprocessing =
tmp_result.split()
#     sys.stderr.write("JUMLAH KATA HASIL
PREPROCESSING:
"+str(len(jumlah_kata_hasil_preprocessing))+ "\n")
    dir3.write("JUMLAH KATA HASIL PREPROCESSING:
"+str(len(jumlah_kata_hasil_preprocessing))+ "\n\n")

    path_full_doc =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/re
sult_preprocessing_dan_labeling_bloom/"+nama_universit
as+"/amat_final_full_result_preprocessing_"
    nama_file_full_doc_result =
path_full_doc+lala2+".txt"
    final_full = open(nama_file_full_doc_result, 'w')

    final_full.write(tmp_result)
    final_full.close()

    perkaya_corpus_tkt(tmp_result,
list_update_tkt_all)
    path_update_tkt_all =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatraining_All/tampung_TKT_"+nama_universit
as+".txt"
    final_tkt_all = open(path_update_tkt_all, 'w')

    for i in range(0, len(list_update_tkt_all)):

    final_tkt_all.write(list_update_tkt_all[i]+ "\n")
    final_tkt_all.close()

sys.stderr.write("SUKSES PREPROCESS!!!\n\n")

```

```

def compute():
    nama_universitas = "Universitas_Sebelas_Maret"
    path =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_universitas/"+nama_universitas+"/"
    dirs = os.listdir(path)
    lala = path[71:]
    lala2 = lala[:-1]
    print lala2

path_full_doc =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/re
sult_preprocessing_dan_labeling_bloom/"+nama_universit
as+"/full_result_preprocessing_"
    ekstensi_full_doc = ".txt"
    nama_file_full_doc_result =
path_full_doc+lala2+ekstensi_full_doc
    final = open(nama_file_full_doc_result, 'w')

nFiles = 0
    listTF = []
    tipe_bahasa = ""
    list_update_tkt1 = []
    list_update_tkt2 = []
    list_update_tkt3 = []
    list_update_tkt4 = []
    list_update_tkt5 = []
    list_update_tkt6 = []
    list_update_tkt7 = []
    list_update_tkt8 = []
    list_update_tkt9 = []
    list_update_tkt_all = []

for eachFile in dirs:
    nFiles += 1
    with open(path+eachFile, 'r+') as myFile:
        inputan_file = myFile.read().replace('\n',
' ')
        final.write(inputan_file+" ")
    final.close()

```

```

dir2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_preprocessing_dan_labeling_bloom/"+nama_univ
ersitas+"/full_result_preprocessing_"+lala2+".txt",
"r")
    dir2_input = dir2.read()

dir3 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_preprocessing_dan_labeling_bloom/"+nama_univ
ersitas+"/analisa_hasil_preprocessing_"+lala2+".txt",
"w")
    jumlah_kata = dir2_input.split()
    sys.stderr.write("JUMLAH SEMUA KATA:
"+str(len(jumlah_kata))+"\n")
    dir3.write("JUMLAH SEMUA KATA:
"+str(len(jumlah_kata))+"\n\n")

    inggris(dir3, nama_universitas, dir2_input, lala2,
listTF, tipe_bahasa, list_update_tkt1,
list_update_tkt2, list_update_tkt3, list_update_tkt4,
list_update_tkt5, list_update_tkt6, list_update_tkt7,
list_update_tkt8, list_update_tkt9,
list_update_tkt_all)

    dir3.close()

#MAIN FUNCTION#
if __name__ == "__main__":
    start_time = time.time()

    compute()

    time_akhir = time.time() - start_time
    sys.stderr.write("--- "+str(time_akhir)+" detik --
-")

```

Kode Sumber 6.2-1 Kode Sumber Text Preprocessing

1.2. Kode Sumber Memberkaya Corpus TKT dengan Synonym Word

```

from nltk.corpus import wordnet
import random, time, sys, os, re

start_time = time.time()

#fungsi sorting str+number (campuran) tipe karakter
def atoi(text):
    return int(text) if text.isdigit() else text
def natural_keys(text):
    return [atoi(c) for c in re.split('(\d+)', text)]

openfile_tkt1 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT1.txt", "r")
content_tkt1 = openfile_tkt1.read()
list1 = content_tkt1.split()

openfile_tkt2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT2.txt", "r")
content_tkt2 = openfile_tkt2.read()
list2 = content_tkt2.split()

openfile_tkt3 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT3.txt", "r")
content_tkt3 = openfile_tkt3.read()
list3 = content_tkt3.split()

openfile_tkt4 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT4.txt", "r")
content_tkt4 = openfile_tkt4.read()
list4 = content_tkt4.split()

openfile_tkt5 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT5.txt", "r")
content_tkt5 = openfile_tkt5.read()
list5 = content_tkt5.split()

```

```

openfile_tkt6 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT6.txt", "r")
content_tkt6 = openfile_tkt6.read()
list6 = content_tkt6.split()

openfile_tkt7 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT7.txt", "r")
content_tkt7 = openfile_tkt7.read()
list7 = content_tkt7.split()

openfile_tkt8 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT8.txt", "r")
content_tkt8 = openfile_tkt8.read()
list8 = content_tkt8.split()

openfile_tkt9 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/TKT9.txt", "r")
content_tkt9 = openfile_tkt9.read()
list9 = content_tkt9.split()

openfile_tkt_all =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain/ALL.txt", "r")
content_tkt_all = openfile_tkt_all.read()
list_all = content_tkt_all.split()

path =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_All/"
dirs = os.listdir(path)

path_simpan_all =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_All/simpan_all.txt"
final = open(path_simpan_all, 'w')

for eachFile in dirs:
    with open(path+eachFile, 'r+') as myFile:
        inputan_file = myFile.read()
        final.write(str(inputan_file))
final.close()

```



```
dir2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Datatrain_All/simpan_all.txt", "r")
dir2_input = dir2.read()
jos = dir2_input.split()
listX = jos

list_update_tkt_all = []
list_update_tkt1 = []
list_update_tkt2 = []
list_update_tkt3 = []
list_update_tkt4 = []
list_update_tkt5 = []
list_update_tkt6 = []
list_update_tkt7 = []
list_update_tkt8 = []
list_update_tkt9 = []

jump = []
jump2 = None

jonas = []
jonas2 = []
jonas3 = []

joni = []
tampung = []
tampung2 = []
tampung3 = []

jonas_blue = {}
jonas_red = {}
jonas_green = {}

list_A = {}
list_val_A = {}
list_pair_A = {}

index=0
for i in range(0, len(listX)):
    tmp_list_all = []
    maks_tmp_list_all = {}
    tmp_wordx = listX[index]
```

```

for word_all in list_all:
    wordFromListX = wordnet.synsets(tmp_wordx)
    wordFromListAll = wordnet.synsets(word_all)

    if wordFromListX and wordFromListAll:
        pp =
wordFromListX[0].wup_similarity(wordFromListAll
[0])
        tmp_list_all.append(pp)

    if pp != jump2:
        if tmp_wordx in list_A:
            if tmp_wordx==list_A[tmp_wordx]
and pp>=list_val_A[tmp_wordx]:
                del list_A[tmp_wordx]
                del list_val_A[tmp_wordx]
                del list_pair_A[tmp_wordx]

                list_A[tmp_wordx] = tmp_wordx
                list_val_A[tmp_wordx] = pp
                list_pair_A[tmp_wordx] = word_all
                print "JON", tmp_wordx, pp, word_all

            else:
                list_A[tmp_wordx] = tmp_wordx
                list_val_A[tmp_wordx] = pp
                list_pair_A[tmp_wordx] = word_all

index+=1

print list_A
print list_val_A
print list_pair_A
print len(list_pair_A)

for key, value in list_pair_A.iteritems():
    print key
    print value

if value in list9:
    print "ADA 9"
    term = key
    list_update_tkt9.append(term)
    # break;

```

```
elif value in list8:
    print "ADA 8"
    term = key
    list_update_tkt8.append(term)
    # break;
elif value in list7:
    print "ADA 7"
    term = key
    list_update_tkt7.append(term)
    # break;
elif value in list6:
    print "ADA 6"
    term = key
    list_update_tkt6.append(term)
    # break;
elif value in list5:
    print "ADA 5"
    term = key
    list_update_tkt5.append(term)
    # break;
elif value in list4:
    print "ADA 4"
    term = key
    list_update_tkt4.append(term)
    # break;

elif value in list3:
    print "ADA 3"
    term = key
    list_update_tkt3.append(term)
    # break;
elif value in list2:
    print "ADA 2"
    term = key
    list_update_tkt2.append(term)
    # break;
elif value in list1:
    print "ADA 1"
    term = key
    list_update_tkt1.append(term)
    # break;
```

```
#print hasil to txt file
path_update_tkt1 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT1.txt"
final_tkt1 = open(path_update_tkt1, 'w')

for i in range(0, len(list_update_tkt1)):
    final_tkt1.write(list_update_tkt1[i]+"\\n")

path_update_tkt2 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT2.txt"
final_tkt2 = open(path_update_tkt2, 'w')

for i in range(0, len(list_update_tkt2)):
    final_tkt2.write(list_update_tkt2[i]+"\\n")

path_update_tkt3 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT3.txt"
final_tkt3 = open(path_update_tkt3, 'w')

for i in range(0, len(list_update_tkt3)):
    final_tkt3.write(list_update_tkt3[i]+"\\n")

path_update_tkt4 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT4.txt"
final_tkt4 = open(path_update_tkt4, 'w')

for i in range(0, len(list_update_tkt4)):
    final_tkt4.write(list_update_tkt4[i]+"\\n")

path_update_tkt5 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT5.txt"
final_tkt5 = open(path_update_tkt5, 'w')

for i in range(0, len(list_update_tkt5)):
    final_tkt5.write(list_update_tkt5[i]+"\\n")
```

```

path_update_tkt6 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT6.txt"
final_tkt6 = open(path_update_tkt6, 'w')

for i in range(0, len(list_update_tkt6)):
    final_tkt6.write(list_update_tkt6[i]+"\\n")

path_update_tkt7 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT7.txt"
final_tkt7 = open(path_update_tkt7, 'w')

for i in range(0, len(list_update_tkt7)):
    final_tkt7.write(list_update_tkt7[i]+"\\n")

path_update_tkt8 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT8.txt"
final_tkt8 = open(path_update_tkt8, 'w')

for i in range(0, len(list_update_tkt8)):
    final_tkt8.write(list_update_tkt8[i]+"\\n")

path_update_tkt9 =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/do
kumen_tkt/A_Datatrain_Update/lolo_update_TKT9.txt"
final_tkt9 = open(path_update_tkt9, 'w')

for i in range(0, len(list_update_tkt9)):
    final_tkt9.write(list_update_tkt9[i]+"\\n")

final_tkt1.close()
final_tkt2.close()
final_tkt3.close()
final_tkt4.close()
final_tkt5.close()
final_tkt6.close()
final_tkt7.close()
final_tkt8.close()
final_tkt9.close()

```

Kode Sumber 6.2-2 Kode Sumber Memberkaya Corpus TKT

1.3. Kode Sumber Pencocokan Term Asumsi Label Awal

```
#!/usr/bin/python
# -*- coding: latin-1 -*-

import sys
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize,
sent_tokenize, RegexpTokenizer
import re, string, nltk, os, math, random, time
from textblob import TextBlob

#fungsi sorting str+number (campuran) tipe karakter
def atoi(text):
    return int(text) if text.isdigit() else text

def natural_keys(text):
    return [atoi(c) for c in re.split('(\\d+)', text)]

def term_frequency(tmp_result, isi_dokumen, dir3, dir4):
    openfile_tkt1 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
    TE/dokumen_tkt/A_Level_3/amat_final_update_TKT1.txt",
    "r")
        content_tkt1 = openfile_tkt1.read()
        content_tkt1 = content_tkt1.split()
        openfile_tkt1.close()

    openfile_tkt2 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
    TE/dokumen_tkt/A_Level_3/amat_final_update_TKT2.txt",
    "r")
        content_tkt2 = openfile_tkt2.read()
        content_tkt2 = content_tkt2.split()
        openfile_tkt2.close()

    openfile_tkt3 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
    TE/dokumen_tkt/A_Level_3/amat_final_update_TKT3.txt",
    "r")
        content_tkt3 = openfile_tkt3.read()
        content_tkt3 = content_tkt3.split()
        openfile_tkt3.close()
```

```
        openfile_tkt4 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Level_3/amat_final_update_TKT4.txt",
"r")
        content_tkt4 = openfile_tkt4.read()
        content_tkt4 = content_tkt4.split()
        openfile_tkt4.close()
openfile_tkt5 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Level_3/amat_final_update_TKT5.txt",
"r")
        content_tkt5 = openfile_tkt5.read()
        content_tkt5 = content_tkt5.split()
        openfile_tkt5.close()

openfile_tkt6 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Level_3/amat_final_update_TKT6.txt",
"r")
        content_tkt6 = openfile_tkt6.read()
        content_tkt6 = content_tkt6.split()
        openfile_tkt6.close()
openfile_tkt7 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Level_3/amat_final_update_TKT7.txt",
"r")
        content_tkt7 = openfile_tkt7.read()
        content_tkt7 = content_tkt7.split()
        openfile_tkt7.close()
openfile_tkt8 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Level_3/amat_final_update_TKT8.txt",
"r")
        content_tkt8 = openfile_tkt8.read()
        content_tkt8 = content_tkt8.split()
        openfile_tkt8.close()

        openfile_tkt9 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/dokumen_tkt/A_Level_3/amat_final_update_TKT9.txt",
"r")
        content_tkt9 = openfile_tkt9.read()
        content_tkt9 = content_tkt9.split()
        openfile_tkt9.close()
```

```

counter1=counter2=counter3=counter4=counter5=counter6=
counter7=counter8=counter9=counter_all=0
    counter_not_in=0

    tmp_result = tmp_result.split()
    tmp_word = ""
dir4.write("JUMLAH KATA DALAM DOKUMEN:
"+str(len(tmp_result))+ "\n\n")

listWord = []
    lw = {}
    tmp_result_2 = ""

    for word in tmp_result:
        if word in lw:
            lw[word] += 1;
        else:
            lw[word] = 1;
            tmp_result_2 += word + " "
    listWord.append(lw)

dir4.write("JUMLAH KATA SINGLE :
"+str(len(lw))+ "\n\n")

    tmp_result_counter1 = ""
    tmp_result_counter2 = ""
    tmp_result_counter3 = ""
    tmp_result_counter4 = ""
    tmp_result_counter5 = ""
    tmp_result_counter6 = ""
    tmp_result_counter7 = ""
    tmp_result_counter8 = ""
    tmp_result_counter9 = ""

    tmp_result_counter_not_in = ""
    tmp_result_counter_in = ""

tmp_result_2_final = tmp_result_2.split()

    for word in tmp_result_2_final:
        if word in content_tkt1:
            counter1 += 1
            tmp_result_counter1 += word + " "

```



```
elif word in content_tkt2:
    counter2 += 1
    tmp_result_counter2 += word + " "

elif word in content_tkt3:
    counter3 += 1
    tmp_result_counter3 += word + " "

elif word in content_tkt4:
    counter4 += 1
    tmp_result_counter4 += word + " "

elif word in content_tkt5:
    counter5 += 1
    tmp_result_counter5 += word + " "

elif word in content_tkt6:
    counter6 += 1
    tmp_result_counter6 += word + " "

elif word in content_tkt7:
    counter7 += 1
    tmp_result_counter7 += word + " "

elif word in content_tkt8:
    counter8 += 1
    tmp_result_counter8 += word + " "

elif word in content_tkt9:
    counter9 += 1
    tmp_result_counter9 += word + " "

elif word not in content_tkt1 and word not in
content_tkt2 and word not in content_tkt3 and word not
in content_tkt4 and word not in content_tkt5 and word
not in content_tkt6 and word not in content_tkt7 and
word not in content_tkt8 and word not in content_tkt9:
    counter_not_in +=1
    tmp_result_counter_not_in += word + " "

else:
    tmp_result_counter_in += word + " "
```

```

print "TKT 1:\n", tmp_result_counter1
print "TKT 2:\n", tmp_result_counter2
print "TKT 3:\n", tmp_result_counter3
print "TKT 4:\n", tmp_result_counter4
print "TKT 5:\n", tmp_result_counter5
print "TKT 6:\n", tmp_result_counter6
print "TKT 7:\n", tmp_result_counter7
print "TKT 8:\n", tmp_result_counter8
print "TKT 9:\n", tmp_result_counter9

list_label = [counter1, counter2, counter3, counter4,
counter5, counter6, counter7, counter8, counter9]
list_nama_label =
["T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"]

dir4.write("JUMLAH KATA NOT IN DOKUMEN:
"+str(counter_not_in)+"\n\n")

dir4.write("FREKUENSI
TERM:\n"+str(list_label)+"\n"+str(list_nama_label)+"\n
\n")

tmp_lnl = {}
a=0
for lnl in list_nama_label:
    tmp_lnl[a] = lnl
    a+=1

tmp_lb = {}
b=0
p = ""
simpan = []

for lb in list_label:
    if lb>0:
        tmp_lb[b] = lb
        simpan.append(str(tmp_lb[b])+"-
"+str(tmp_lnl[b]))
        p += tmp_lnl[b] + ", "

        b+=1

a = sorted(simpan, key=natural_keys, reverse=True)

```

```

sys.stderr.write("SORTING HASIL TF (pencocokan
berdasarkan term):\n"+str(a)+"\n")
    dir4.write("SORTING HASIL TF (pencocokan
berdasarkan term):\n"+str(a)+"\n\n")

ambil_4_label_teratas = a[:3]
    ambil_4_label_teratas = '
'.join(ambil_4_label_teratas)

    b = ambil_4_label_teratas.split()

    c = {}
    d=0
    e=0
    f = {}
    label_final = ""
    for j in b:
        d+=1
        c[d] = j[-2:]

    for k in b:
        label_final += str(c[d-e])+", "
        e+=1

    label_final = label_final[:-1]
    label_final_akhir = "["+label_final+"]"
    label_final_akhir = re.sub('-', ' ',
label_final_akhir)
    sys.stderr.write("SORTING HASIL TF AMBIL 4
TERATAS:\n"+str(label_final_akhir)+"\n")

    dir3.write(str(label_final_akhir)+"
"+str(isi_dokumen))

def compute():
    nama_universitas = "Universitas_Sebelas_Maret"
    dir2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_preprocessing_dan_labeling_bloom/"+nama_univ
ersitas+"/amat_final_full_result_preprocessing_"+nama_
universitas+".txt", "r")

    dir2_input = dir2.read()
    isi_dokumen = dir2_input

```

```

dir2.close()

    dir3 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_term_frequency_dan_labeling_bloom/"+nama_uni
versitas+"/label_dokumen_based_on_term_frequency_"+nam
a_universitas+".txt", "w")
    dir4 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_term_frequency_dan_labeling_bloom/"+nama_uni
versitas+"/analisa_hasil_term_frequency_"+nama_univers
itas+".txt", "w")

    term_frequency(dir2_input, isi_dokumen, dir3,
dir4)

    dir3.close()
    dir4.close()

#MAIN FUNCTION#
if __name__ == "__main__":
    start_time = time.time()

    compute()

    time_akhir = time.time() - start_time
    sys.stderr.write("--- "+str(time_akhir)+" detik --
-")

```

Kode Sumber 6.2-3 Kode Sumber Pencocokan Term Asumsi Label Awal

1.4. Kode Sumber Metode L-LDA

```

#LLDA: Labeled Latent Dirichlet Allocation#

from optparse import OptionParser
import sys, re, numpy
import time
import csv

def load_corpus(filename):
    nama_file = str(filename)

    corpus = []
    labels = []
    labelmap = dict()

    open_file_corpus = open(filename, 'r')

    line = ""

    for line in open_file_corpus:
        get_label_per_line = re.match(r'\[(.+?)\](.+)',
line)

        if get_label_per_line:
            label = get_label_per_line.group(1).split(',')
            for i in label:
                labelmap[i] = 1

            line = get_label_per_line.group(2)

        else:
            label = None

        doc = re.findall(r'\w+(?:\'\w+)?',
line.lower())

        if len(doc)>0:
            corpus.append(doc)
            labels.append(label)

    open_file_corpus.close()

    return labelmap.keys(), corpus, labels,
nama_file

```

```

class LLDA:
    def __init__(self, K, alpha, beta):
        self.alpha = alpha
        self.beta = beta

    def complement_label(self, label):
        if not label:
            return numpy.ones(len(self.labelmap))

        vec_complemet_label =
numpy.zeros(len(self.labelmap))

        for i in label:
            vec_complemet_label[self.labelmap[i]] = 1.0
        return vec_complemet_label

    def term_to_id(self, word):
        if word not in self.vocas_id:
            voca_id = len(self.vocas)
            self.vocas_id[word] = voca_id
            self.vocas.append(word)
        else:
            voca_id = self.vocas_id[word]

        return voca_id

    def set_corpus(self, labelset, corpus, labels):
        #labelset.insert(0, "common")
        self.labelmap = dict(zip(labelset,
range(len(labelset))))
        self.K = len(self.labelmap)

        self.vocas = []
        self.vocas_id = dict()

self.labels =
numpy.array([self.complement_label(label) for label in
labels])

self.docs = [[self.term_to_id(word) for word in doc]
for doc in corpus]
        M = len(corpus)
        V = len(self.vocas)

```

```

self.zmn = []
self.nmz = numpy.zeros((M, self.K), dtype=int)
self.nzt = numpy.zeros((self.K, V), dtype=int)
self.nz = numpy.zeros(self.K, dtype=int)

i=1
for m, doc, label in zip(range(M), self.docs,
self.labels):
    nm = len(doc)
    zn = [numpy.random.multinomial(1,
label/label.sum()).argmax() for x in range(nm)
self.zmn.append(zn)

    for t, z in zip(doc, zn):
        self.nmz[m, z] += 1
        self.nzt[z, t] += 1
        self.nz[z] += 1
    i+=1

def gibbs_sampling_inference(self):
    V = len(self.vocas)

    for m, doc, label in zip(range(len(self.docs)),
self.docs, self.labels):
        len_doc = len(doc)
        for n in range(len_doc):
            t = doc[n]
            z = self.zmn[m][n]
            self.nmz[m, z] -= 1
            self.nzt[z, t] -= 1
            self.nz[z] -=1

        denom_alpha = self.nmz[m].sum() + self.K * self.alpha
            denom_beta =
self.nzt.sum(axis=1) + V * self.beta

            probabilitas_z = label *
(self.nzt[:, t] + self.beta) / denom_beta *
(self.nmz[m] + self.alpha) / denom_alpha
            new_z =
numpy.random.multinomial(1, probabilitas_z /
probabilitas_z.sum()).argmax()

```

```

self.zmn[m][n] = new_z
self.nmz[m, new_z] += 1
self.nzt[new_z, t] += 1
self.nz[new_z] += 1

def phi(self):
    V = len(self.vocas)
    nilai_phi = (self.nzt + self.beta) /
    (self.nz[:, numpy.newaxis] + V * self.beta)
    return nilai_phi

def theta(self):
    n_alpha = self.nmz + self.labels * self.alpha
    nilai_theta = n_alpha / n_alpha.sum(axis=1)[:,
numpy.newaxis]
    return nilai_theta

def perplexity(self, docs=None):
    if docs == None:
        docs = self.docs

    nilai_phi = self.phi()
    nilai_thetas = self.theta()
    N = 0
    log_per = 0

    for doc, theta in zip(docs, nilai_thetas):
        for word in doc:
            log_per -=
numpy.log(numpy.inner(nilai_phi[:, word], theta))
            N += len(doc)
            result_perplexity = numpy.exp(log_per/N)
            print result_perplexity
            return result_perplexity

```



```

def main():
    parser = OptionParser()
    parser.add_option("-f", dest="filename",
        help="corpus filename") #get nama file corpusnya
    parser.add_option("--alpha", dest="alpha",
        type="float", help="parameter alpha (parameter
        penyemibang jumlah topik T)", default=25.0)
    parser.add_option("--beta", dest="beta",
        type="float", help="parameter beta (parameter
        penyemibang jumlah kata W)", default=0.01)
    parser.add_option("-k", dest="K", type="int",
        help="jumlah topik utama (sesuai jumlah TKT)",
        default=9)
    parser.add_option("-i", dest="iteration",
        type="int", help="jumlah iterasi gibbs sampling",
        default=100)

    (options, args) = parser.parse_args()
    if not options.filename:
        parser.error("Masukkan nama file corpus
        anda (diawali dengan -f)")

    labelset, corpus, labels, nama_file =
    load_corpus(options.filename)

    lala = nama_file[115:]
    lala2 = lala[:-4]
    lala_final = lala2.replace("_", " ")
    print lala_final

    llda = LLDA(options.K, options.alpha, options.beta)
    llda.set_corpus(labelset, corpus, labels)

    for i in range(options.iteration):
        llda.gibbs_sampling_inference()
    print '\n'

    simpan_file =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
    TE/result_llda_with_tf_idf_sklearn/result_llda_"+lala_
    final+".txt", "w")

```

```

simpan_file_hasil_theta =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_llda_with_tf_idf_sklearn/result_theta_max_"+
lala_final+".txt", "w")
    simpan_file_hasil_theta_all =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDA
TE/result_llda_with_tf_idf_sklearn/result_theta_all_"+
lala_final+".txt", "w")

    phi = llda.phi()
    theta = llda.theta()

    simpan_file.write(str(labelset))
    simpan_file.write("\n")

header_phi = '\n\n=====
PROBABILITAS PHI ====='
    simpan_file.write(header_phi+"\n")

for v, vocabulary in enumerate(llda.vocas):
tmp_hasil_label = ', '.join([vocabulary] +
[str(hasil_x) for hasil_x in llda.nzt[:, v]])

tmp_hasil_phi = ", ".join([vocabulary] + [str(hasil_x)
for hasil_x in phi[:, v]])

    simpan_file.write(tmp_hasil_label)
    simpan_file.write("\n")
    simpan_file.write(tmp_hasil_phi)
    simpan_file.write("\n\n")

simpan_file.write("\n\n"+str(labelset))

    header_theta =
'\n\n===== PROBABILITAS THETA
===== '
    print header_theta
    simpan_file.write(header_theta+"\n")

d=0
tmp_posisi_labelmap_key_akhir = llda.K - 1
tmp_posisi_labelmap_key_awal = 0

```

```

for p, label_per_document in enumerate(llda.labels):
    j=0
    flag=0
    simpan = []
    simpan2 = {}

    p+=1
    print "LABEL DOKUMEN", p, ":",
label_per_document
    simpan_file.write("LABEL DOKUMEN "+str(p)+"
"+str(label_per_document)+"\n")

    for hasil_y in theta[d, :]:
        if tmp_posisi_labelmap_key_awal >
tmp_posisi_labelmap_key_akhir:

            tmp_posisi_labelmap_key_awal = 0
            simpan.append(str(hasil_y))
            simpan2[j] = hasil_y
        print "Posisi Label ke -", j, ":", simpan2[j]

        simpan_file_hasil_theta_all.write(str(lab
elset[j])+ " "+str(simpan2[j])+"\n")
        j+=1
        tmp_posisi_labelmap_key_awal+=1

        print "--- NILAI N ---:", j, "\n"
        d+=1

        print "HASIL ARRAY PROBABILITAS\n",
simpan, "\n"
        simpan_file.write("HASIL TANPA SORTING
PROBABILITAS: "+str(simpan)+"\n")

        a = sorted(simpan, key=float, reverse=True)
        print "HASIL SORTING PROBABILITAS\n", a, "\n"
        simpan_file.write("HASIL SORTING PROBABILITAS:
"+str(a)+"\n")

        seq = a[:1]
        final_seq = ''.join(seq)
        print "Probabilitas tertinggi adalah:",
final_seq

```

```

simpan_file.write("PROBABILITAS TERTINGGI:
"+str(final_seq)+"\n")
    tanda=0

    for num in range(0, j):
        if final_seq==simpan[num]:
            tanda += 1
            if tanda<=1:
                print "INI KETEMU"
                flag=num
                print "INI FLAG", flag

            seq_labelset = labelset[num:]
            seq_labelset_2 = seq_labelset[:1]
            final_seq_labelset =
''.join(seq_labelset_2)
                print "TOPIK FINAL:", final_seq_labelset

        simpan_file.write("TOPIK FINAL DOKUMEN
"+str(p)+" : "+str(final_seq_labelset)+"\n\n")

simpan_file_hasil_theta.write(str(final_seq_labelset)+
"\n")

        print "TANDA: ADA", tanda, "ARRAY TERATAS
DENGAN NILAI YANG SAMA"
        print '\n'

    print "TOTAL DOKUMEN:", d

    simpan_file.close()
    simpan_file_hasil_theta.close()
    print("SUKSES SIMPAN!")

if __name__ == "__main__":
    start_time = time.time()

    main()

    time_akhir = time.time() - start_time
    sys.stderr.write("--- "+str(time_akhir)+" detik
---")

```

Kode Sumber 6.2-4 Kode Sumber Metode L-LDA

1.5. Kode Sumber Pembobotan Skor Reputasi Akademik

```

import time, sys, MySQLdb, time

def send_to_database(skor_universitas, bobot_total,
jumlah_dokumen, nama_universitas):
    host = "localhost";
    user = "root";
    password = "";
    dbname = "database_qs_2";

    db = MySQLdb.connect(host,user,password,dbname)
    cursor = db.cursor()

    nama_univ = nama_universitas
    nilai_bobot_total = skor_universitas
    tanggal = time.strftime('%Y-%m-%d %H:%M:%S')

try:
    cursor.execute("SELECT * FROM
TABEL_REPUTASI_AKADEMIK \
                    WHERE NAMA_UNIVERSITAS='%s'" %
(nama_univ))
    results = cursor.fetchall()

    if results:
        cursor.execute("DELETE FROM
TABEL_REPUTASI_AKADEMIK WHERE NAMA_UNIVERSITAS='%s'" %
(nama_univ))
        cursor.execute("INSERT INTO
TABEL_REPUTASI_AKADEMIK(NAMA_UNIVERSITAS,
SKOR_UNIVERSITAS, BOBOT_TOTAL, JUMLAH_DOKUMEN,
TANGGAL_AKSES) VALUES (%s, %s, %s, %s, %s)",
(nama_univ, nilai_bobot_total, bobot_total,
jumlah_dokumen, tanggal))
    else:
        cursor.execute("INSERT INTO
TABEL_REPUTASI_AKADEMIK(NAMA_UNIVERSITAS,
SKOR_UNIVERSITAS, BOBOT_TOTAL, JUMLAH_DOKUMEN,
TANGGAL_AKSES) VALUES (%s, %s, %s, %s, %s)",
(nama_univ, nilai_bobot_total, bobot_total,
jumlah_dokumen, tanggal))

```

```

db.commit()
except:
    db.rollback()
db.close()
print "SUKSES INSERT!"

def main():
    nama_file = "result_theta_all_Universitas
Sebelas Maret.txt"
    fname =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_UPDATE/re
sult_llda_with_tf_idf_sklearn/"+nama_file

    lala = fname[100:]
    nama_universitas = lala[:-4]
    print nama_universitas

    a=""
    bobot=0
    bobot_total=0
    float(bobot_total)

    with open(fname) as f:
        content = f.readlines()

    content = [x.strip() for x in content]
    simpan_label = {}
    simpan_probs = {}
    jumlah_label=0

    for i in range(0, len(content)):
        simpan_label[i] = content[i][:2]
        simpan_probs[i] = content[i][3:]

    if(simpan_label[i]=="T1"):
        bobot = 10 * float(simpan_probs[i])
        bobot_total += bobot

    elif(simpan_label[i]=="T2"):
        bobot = 20 * float(simpan_probs[i])
        bobot_total += bobot
    elif(simpan_label[i]=="T3"):
        bobot = 30 * float(simpan_probs[i])
        bobot_total += bobot

```

```

elif(simpan_label[i]=="T4"):
    bobot = 40 * float(simpan_probs[i])
    bobot_total += bobot

elif(simpan_label[i]=="T5"):
    bobot = 50 * float(simpan_probs[i])
    bobot_total += bobot

elif(simpan_label[i]=="T6"):
    bobot = 60 * float(simpan_probs[i])
    bobot_total += bobot

elif(simpan_label[i]=="T7"):
    bobot = 70 * float(simpan_probs[i])
    bobot_total += bobot

elif(simpan_label[i]=="T8"):
    bobot = 80 * float(simpan_probs[i])
    bobot_total += bobot

elif(simpan_label[i]=="T9"):
    bobot = 90 * float(simpan_probs[i])
    bobot_total += bobot

jumlah_label+=1

jumlah_dokumen=9
skor_universitas =
float(bobot_total)/jumlah_dokumen

sys.stderr.write("JUMLAH LABEL TERDETEKSI:
"+str(jumlah_label)+"\n")
sys.stderr.write("BOBOT TOTAL:
"+str(skor_universitas)+"\n")

if __name__ == "__main__":
    start_time = time.time()

    main()

    time_akhir = time.time() - start_time
    sys.stderr.write("--- "+str(time_akhir)+" detik
---")

```

Kode Sumber 6.2-5 Kode Sumber Pembobotan Skor Reputasi Akademik

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Eko Putro Fitrianto, lahir di Mojokerto, pada tanggal 13 Maret 1995. Penulis menempuh pendidikan mulai dari SD Negeri Sumberdadi 1, Mantup, Lamongan (2001-2007), SMP Negeri 1 Gedeg, Mojokerto (2007-2010), SMK Negeri 1 Kota Mojokerto (2010-2013) dan saat ini sedang menempuh pendidikan Sarjana di Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya angkatan tahun 2013.

Selama belajar di kampus Teknik Informatika, penulis berkesempatan terlibat aktif dalam organisasi kemahasiswaan serta kepanitiaan, antara lain staff magang Departemen Pengembangan Profesi di Himpunan Mahasiswa Teknik Computer-Informatika ITS (2013-2014), staff Departemen Pengembangan Profesi di Himpunan Mahasiswa Teknik Computer-Informatika ITS (2014-2015), staff Biro National Seminar of Technology (NST) pada acara Schematics 2014, wakil 1 badan pengurus harian Biro National Seminar of technology (NST) pada acara Schematics 2015.

Penulis memiliki bidang minat Manajemen Informasi (MI) dengan fokus studi pada bidang data analis, pemrograman website, pemrograman mobile dan teks minning. Komunikasi dengan penulis dapat melalui email: putro994@gmail.com.