

**ANALISIS SENTIMEN *CYBERBULLYING* PADA KOMENTAR
INSTAGRAM DENGAN METODE KLASIFIKASI *SUPPORT*
*VECTOR MACHINE***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Wanda Athira Luqyana

NIM: 145150200111088



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

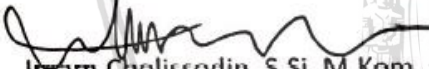

ANALISIS SENTIMEN *CYBERBULLYING* PADA KOMENTAR INSTAGRAM DENGAN
SUPPORT VECTOR MACHINE

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Wanda Athira Luqyana
NIM: 145150200111088

Skripsi ini telah diuji dan dinyatakan lulus pada
04 Mei 2018
Telah diperiksa dan disetujui oleh:

Pembimbing I	Pembimbing II
	
<u>Inram Cholissodin, S.Si, M.Kom</u>	<u>Rizal Setya Perdana, S.Kom, M.Kom</u>
NIK: 201201 850719 1 001	NIK: 201603 910118 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



H. Astoto Kurniawan, S.T, M.T, Ph.D
19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 04 Mei 2018



Wanda Athira Luqyana

NIM: 145150200111088

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan skripsi dengan judul “Analisis Sentimen *Cyberbullying* Pada Komentar Instagram Dengan Metode Klasifikasi *Support Vector Machine*” dengan baik. Skripsi ini disusun dan diselesaikan sebagai syarat dalam memperoleh gelar sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Dalam proses menyelesaikan skripsi, penulis telah mendapat banyak bantuan maupun dukungan baik secara moral maupun materiil dari berbagai pihak. Oleh karena itu penulis mengucapkan banyak terima kasih kepada:

1. Bapak Imam Cholissodin, S.Si., M.Kom., selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan, ilmu, dan masukan dalam menyelesaikan skripsi ini.
2. Bapak Rizal Setya Perdana, S.Kom., M.Kom., selaku Dosen Pembimbing II yang telah memberikan bimbingan, arahan, ilmu, dan masukan dalam menyelesaikan skripsi ini.
3. Kedua orangtua serta adik penulis yang telah memberikan doa maupun dukungan kepada penulis dalam membantu kelancaran pengerjaan skripsi penulis.
4. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Tri Astoto Kurninawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Agus Wahyu Widodo, S.T., M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya
7. Bapak M. Tanzil Furqon, S.Kom., M.Comp.Sc selaku Sekretaris Jurusan Teknik Informatika..
8. Seluruh bapak dan ibu dosen yang telah mendidik dan memberikan ilmu selama penulis menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
9. Ibu Ari Pratiwi, S.Psi., M.Psi selaku dosen psikologi Fakultas Ilmu Sosial dan Ilmu Politik Universitas Brawijaya, yang telah bersedia menjadi narasumber penulis dalam membantu kelancaran proses pengerjaan skripsi.
10. Ibu Millatuz Zakiyah S.Pd., M.A. selaku dosen Bahasa Indonesia Universitas Brawijaya, yang telah bersedia membantu penulis untuk proses kelancaran pengerjaan skripsi.
11. Teman-teman Program Studi Teknik Informatika yang selalu memberikan semangat, dukungan, serta doa selama penulis menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.

12. Teman-Teman baik penulis yaitu Vivin, Icha, dan Farah yang telah mendukung serta selalu menghibur penulis untuk tetap semangat selama masa perkuliahan serta yang telah memberikan inspirasi untuk penulisan skripsi.
13. Teman-teman kecil penulis yaitu Santika dan Via yang selalu memberikan canda dan tawa untuk selalu mendukung dan menghibur penulis selama ini.
14. Andi Mohammad R.H. yang telah menjadi kakak yang mendukung penulis dalam membantu kelancaran pengerjaan skripsi.
15. Teman-teman F-Zoo 2014 yaitu, Mada, Silvi, Ratna, Egi, Rahmat, Rizky, Chandra, dan Rayza yang selalu menemani, mengajari, dan menghibur penulis selama masa perkuliahan berlangsung hingga kini.
16. Serta seluruh pihak yang tidak dapat disebutkan oleh penulis satu per satu yang telah membantu kelancaran pengerjaan skripsi.

Semoga seluruh doa dan dukungan yang telah diberikan dibalas oleh Allah SWT. Penulis menyadari bahwa skripsi ini tidak lepas dari kekurangan baik dalam format maupun isi. Oleh karena itu diharapkan kritik maupun saran yang membangun dalam proses untuk memperbaiki diri. Penulis berharap semoga skripsi yang telah ditulis dapat memberikan manfaat bagi semua pihak.

Malang, 04 Mei 2018

Penulis

wandathira@gmail.com

ABSTRAK

Instagram merupakan media sosial yang paling populer pada zaman sekarang. Pengguna yang dimulai dari anak-anak, remaja hingga orang dewasa turut mendongkrak popularitas Instagram. Namun, media sosial ini tidak lepas dari bahaya cyberbullying yang sering dilakukan oleh pengguna khususnya pada kolom komentar. Dengan data statistik yang telah didapatkan, bahwa 42% remaja berusia 12-20 tahun telah menjadi korban cyberbullying. Bahaya cyberbullying tentunya meresahkan banyak orang dikarenakan dampak yang ditimbulkan, maka dari itu dapat dilakukan suatu analisis sentimen pada kolom komentar Instagram yang berupaya untuk mengetahui sentimen dari setiap komentar. Analisis sentimen merupakan suatu cabang ilmu dari text mining yang digunakan untuk mengekstrak, memahami, dan mengolah data teks. Untuk mengetahui setiap sentimen pada komentar digunakan fitur Term Frequency-Inverse Document Frequency (TF-IDF) dan metode klasifikasi Support Vector Machine (SVM). Dokumen yang berisi 400 data yang diambil secara luring (offline) dengan total fitur 1799. Dokumen komentar dibagi menjadi 70% data latih dan 30% data uji. Berdasarkan pengujian yang dilakukan didapatkan parameter terbaik pada metode SVM yaitu dengan nilai degree kernel polynomial sebesar 2, nilai learning rate sebesar 0,0001, dan jumlah iterasi maksimum yang digunakan adalah 200 kali. Dari pengujian tersebut didapatkan hasil akurasi tertinggi sebesar 90% pada komposisi data latih 50% dan komposisi data uji 50%.

Kata kunci: Instagram, *cyberbullying*, *analisis sentimen*, *Support Vector Machine*, *svm*

ABSTRACT

Instagram is the most popular social media in these recent days. The users who start from kids, teenagers to adults, have the role in boosting the popularity of Instagram. However, this social media could not be seperated from the dangers of cyberbullying which is done often by the users, especially in the comment column. The dangers of cyberbullying are certainly worried many people because of the impact it has. Therefore, a sentiment analysis in Instagram comment column can be done in order to find out the sentiments in each comment. Sentiment analysis is a branch of text mining science which is used to extract, understand, and cultivate the data. This research used Term Frequency-Inverse Document Frequency (TF-IDF) and Support Vector Machine (SVM) classification method to examine the sentiments in each comment. Data consisted of 400 data which taken offline have a total 1799 features. The comment document is divided into 70% of training data and 30% of test data. Based on the tests performed, the best parameters obtained in the SVM method are the degree of polynomial kernel 2, the average of learning rate of 0.0001, and the maximum number of iterations which is 200 times. From these result, it obtained that the highest accuracy is 90%, 50% in the training data composition and 50% composition of test data.

Keywords: Instagram, cyberbullying, sentimen analysis, Support Vector Machine, svm.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
DAFTAR PERSAMAAN.....	xvi
DAFTAR KODE PROGRAM	xvii
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	19
1.1 Latar Belakang.....	19
1.2 Rumusan Masalah.....	21
1.3 Tujuan	21
1.4 Manfaat.....	21
1.5 Batasan Masalah.....	21
1.6 Sistematika Pembahasan.....	22
BAB 2 LANDASAN KEPUSTAKAAN	23
2.1 Kajian Pustaka	23
2.2 Analisis Sentimen	25
2.3 <i>Text Mining</i>	26
2.3.1 <i>Pre-processing</i>	26
2.3.2 Pembobotan TF-IDF	28
2.4 Media Sosial	29
2.4.1 Instagram	30
2.5 <i>Cyberbullying</i>	30
2.6 <i>Lexicon Based Features</i>	32
2.6.1 Normalisasi <i>Min-Max</i>	32
2.6.2 Skor Sentimen	33
2.7 Klasifikasi.....	33

2.7.1 <i>Support Vector Machine</i>	34
2.8 Evaluasi	38
BAB 3 METODOLOGI	39
3.1 Studi Pustaka.....	39
3.2 Perancangan Sistem.....	40
3.3 Pengumpulan Data	40
3.4 Wawancara	40
3.5 Implementasi	40
3.6 Pengujian dan Analisis	41
3.7 Kesimpulan dan Saran	41
BAB 4 ANALISIS DAN PERANCANGAN SISTEM	42
4.1 Deskripsi Permasalahan	42
4.2 Deskripsi Umum Sistem	43
4.3 <i>Pre-processing</i>	44
4.3.2 <i>Case Folding</i>	46
4.3.3 <i>Data cleaning</i>	47
4.3.4 Normalisasi Bahasa	48
4.3.5 <i>Stopword Removal</i>	49
4.3.6 <i>Stemming</i>	51
4.3.7 Tokenisasi	52
4.4 Pembobotan TF-IDF	53
4.4.2 Tentukan Fitur Kata.....	55
4.4.3 Hitung nilai TF.....	56
4.4.4 Hitung nilai W_{tf}	57
4.4.5 Hitung Nilai DF dan IDF	59
4.4.6 Hitung Nilai TF-IDF.....	60
4.5 <i>Lexicon Based Features</i>	62
4.5.1 Normalisasi <i>Min-Max</i>	62
4.5.2 Skor Sentimen	64
4.6 <i>Support Vector Machine</i>	65
4.6.2 Perhitungan Kernel	67
4.6.3 Perhitungan Matriks Hessian	68

4.6.4	Perhitungan <i>Sequential Training SVM</i>	70
4.6.5	Perhitungan Bias	73
4.6.6	Perhitungan <i>Testing</i>	76
4.7	Manualisasi Metode Support Vector Machine.....	79
4.7.1	Manualisasi <i>Case Folding</i>	80
4.7.2	Manualisasi <i>Data Cleaning</i>	83
4.7.3	Manualisasi Normalisasi Bahasa	85
4.7.4	Manualisasi <i>Stopword Removal</i>	87
4.7.5	Manualisasi <i>Stemming</i>	90
4.7.6	Manualisasi Tokenisasi.....	91
4.7.7	Manualisasi Perhitungan <i>tf</i> , $W_{tf,d}$, <i>df</i> , dan <i>idf</i>	93
4.7.8	Manualisasi Perhitungan TF-IDF	96
4.7.9	Manualisasi Perhitungan <i>Lexicon Based Features</i> dengan Normalisasi <i>Min-Max</i>	97
4.7.10	Manualisasi Perhitungan Klasifikasi <i>Support Vector Machine</i> ..	98
4.7.11	Manualisasi Perhitungan Evaluasi.....	104
4.8	Perancangan Pengujian	105
4.8.1	Perancangan Pengujian Terhadap Parameter <i>Support Vector Machine</i>	105
4.8.2	Perancangan Pengujian Implementasi <i>Lexicon Based Features</i>	106
4.8.3	Penarikan Kesimpulan.....	108
BAB 5	IMPLEMENTASI	109
5.1	Implementasi Sistem	109
5.2	<i>Pre-processing</i>	109
5.2.1	<i>Case Folding</i>	109
5.2.2	<i>Data Cleansing</i>	110
5.2.3	Normalisasi Bahasa	110
5.2.4	<i>Stopword Removal</i>	111
5.2.5	<i>Stemming</i>	111
5.2.6	Tokenisasi	112
5.3	Perhitungan Kata	112
5.3.1	Pencarian Fitur Kata	112
5.3.2	Perhitungan Nilai TF dan $W_{tf,d}$	113



5.3.3	Perhitungan Nilai DF dan IDF	114
5.3.4	Perhitungan Nilai TF-IDF	114
5.4	Pembobotan Lexicon Based Features.....	115
5.5	Klasifikasi dengan <i>Support Vector Machine</i>	117
5.5.1	Pembentukan Matriks dan Transposisi Matriks	117
5.5.2	Perhitungan Kernel	118
5.5.3	Perhitungan Matriks Hessian	119
5.5.4	Perhitungan Sequential Learning.....	120
5.5.5	Perhitungan <i>Bias</i>	121
5.5.6	Perhitungan <i>Testing</i>	122
5.6	Evaluasi	122
5.6.1	Perhitungan Confusion Matrix.....	123
5.6.2	Perhitungan Akurasi, Precision, Recall, dan F-Measure	123
5.6.3	Perhitungan Waktu Komputasi.....	124
BAB 6	PENGUJIAN DAN ANALISIS.....	125
6.1	Pengujian Pengaruh Parameter <i>Support Vector Machine</i>	125
6.1.1	Skenario Pengujian Pengaruh Parameter <i>Support Vector Machine</i>	125
6.1.2	Analisis Hasil Pengujian Pengaruh Parameter <i>Support Vector Machine</i>	127
6.2	Pengujian Pengaruh Implementasi <i>Lexicon Based Features</i>	130
6.2.1	Skenario Pengujian Pengaruh Implementasi <i>Lexicon Based Features</i>	130
6.2.2	Analisis Hasil Pengujian Pengaruh Penerapan <i>Lexicon Based Features</i>	131
BAB 7	PENUTUP	134
7.1	Kesimpulan.....	134
7.2	Saran	135
DAFTAR PUSTAKA	136

DAFTAR GAMBAR

Gambar 2.1 Pemisah <i>Hyperplane</i> Terbaik.....	34
Gambar 2.2 <i>Overfitting</i> dan <i>Underfitting</i>	36
Gambar 3.1 Blok Diagram Metodologi Penelitian	39
Gambar 4.1 Deskripsi Umum Sistem	43
Gambar 4.2 Alur Proses <i>Pre-processing</i>	45
Gambar 4.3 Alur Proses <i>Case Folding</i>	46
Gambar 4.4 Alur Proses <i>Data cleaning</i>	47
Gambar 4.5 Alur Proses Normalisasi Bahasa	49
Gambar 4.6 Alur Proses <i>Stopword Removal</i>	50
Gambar 4.7 Alur Proses <i>Stemming</i>	51
Gambar 4.8 Alur Proses Tokenisasi.....	53
Gambar 4.9 Alur Proses Pembobotan TF-IDF	54
Gambar 4.10 Alur Proses Menentukan Fitur Kata	55
Gambar 4.11 Alur Proses Hitung Nilai TF.....	57
Gambar 4.12 Alur Proses Hitung Nilai W_{tf}	58
Gambar 4.13 Alur Proses Hitung Nilai DF dan IDF	60
Gambar 4.14 Alur Proses Hitung Nilai TF-IDF	61
Gambar 4.15 Alur Proses <i>Lexicon Based Features</i> dengan normalisasi <i>min-max</i>	63
Gambar 4.16 Alur Proses <i>Lexicon Based Features</i> dengan skor sentimen	65
Gambar 4.17 Alur Proses <i>Support Vector Machine</i>	66
Gambar 4.18 Alur Proses Perhitungan Kernel	68
Gambar 4.19 Alur Proses Perhitungan Matriks Hessian	69
Gambar 4.20 Alur Proses Perhitungan <i>Sequential Learning SVM</i>	72
Gambar 4.21 Alur Proses Perhitungan <i>Bias</i>	75
Gambar 4.22 Alur Proses Perhitungan Testing	78
Gambar 6.1 Grafik hasil pengujian pengaruh nilai <i>degree</i>	127
Gambar 6.2 Grafik hasil pengujian iterasi maksimum	128
Gambar 6.3 Grafik hasil pengujian konstanta <i>learning rate</i>	129
Gambar 6.4 Grafik hasil pengujian pengaruh iterasi maksimum terhadap waktu komputasi sistem	129

Gambar 6.5 Grafik hasil pengujian perbandingan pengaruh penerapan *lexicon based features* dan tanpa *lexicon based features* 133



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	24
Tabel 2.2 Contoh kata positif	31
Tabel 2.3 Contoh kata negatif	32
Tabel 2.4 <i>Confusion Matrix</i>	38
Tabel 4.1 Data latih	79
Tabel 4.2 Data uji	80
Tabel 4.3 Data latih <i>proses case folding</i>	81
Tabel 4.4 Data uji <i>proses case folding</i>	82
Tabel 4.5 Data latih <i>proses data cleaning</i>	83
Tabel 4.6 Data uji <i>proses data cleaning</i>	84
Tabel 4.7 Data Latih Proses Normalisasi Bahasa	85
Tabel 4.8 Data Uji Proses Normalisasi Bahasa	87
Tabel 4.9 Data latih <i>proses stopword removal</i>	88
Tabel 4.10 Data uji <i>proses stopword removal</i>	89
Tabel 4.11 Data Latih Proses <i>Stemming</i>	90
Tabel 4.12 Data Uji Proses <i>Stemming</i>	91
Tabel 4.13 Data latih <i>proses tokenisasi</i>	92
Tabel 4.14 Data uji <i>proses tokenisasi</i>	92
Tabel 4.15 Perhitungan nilai <i>tf</i> dan <i>df</i> pada data latih dan data uji	93
Tabel 4.16 Perhitungan nilai <i>Wtd</i> dan <i>idf</i> pada data latih dan data uji	94
Tabel 4.17 Perhitungan TF-IDF pada data latih dan data uji	96
Tabel 4.18 Pembobotan <i>Lexicon</i> pada data latih dan data uji	97
Tabel 4.19 Normalisasi pembobotan <i>Lexicon</i>	98
Tabel 4.20 Fitur <i>term</i> hasil perhitungan pembobotan TF-IDF dan pembobotan <i>Lexicon</i> yang diproses dengan SVM	99
Tabel 4.21 Hasil perhitungan kernel pada data latih	100
Tabel 4.22 Hasil perhitungan matriks Hessian	101
Tabel 4.23 Hasil perhitungan Error rate pada data latih	101
Tabel 4.24 Hasil perhitungan delta alfa pada data latih	102
Tabel 4.25 Hasil perhitungan <i>ai</i> pada data latih	102
Tabel 4.26 Hasil E3 iterasi maksimum pada data latih	103



Tabel 4.27 Hasil δa_3 iterasi maksimum pada data latih 103

Tabel 4.28 Hasil a_3 iterasi maksimum pada data latih 103

Tabel 4.29 Hasil perhitungan nilai bias 103

Tabel 4.30 Hasil perhitungan klasifikasi pada data uji 104

Tabel 4.31 Hasil *Confusion Matrix* 104

Tabel 4.32 Perancangan Pengujian Nilai *Degree* 105

Tabel 4.33 Perancangan Pengujian Konstanta Learning Rate γ 106

Tabel 4.34 Perancangan pengujian *Lexicon Based Features* 107

Tabel 6.1 Hasil pengujian pengaruh nilai *degree* 126

Tabel 6.2 Hasil pengujian konstanta *learning rate* 126

Tabel 6.3 Hasil pengujian pengaruh implementasi *léxicon based features*..... 130



DAFTAR PERSAMAAN

Persamaan (2.1)	28
Persamaan (2.2).....	29
Persamaan (2.3).....	29
Persamaan (2.4).....	33
Persamaan (2.5).....	35
Persamaan (2.6).....	35
Persamaan (2.7).....	35
Persamaan (2.8)	35
Persamaan (2.9).....	35
Persamaan (2.10).....	35
Persamaan (2.11).....	35
Persamaan (2.12).....	35
Persamaan (2.13).....	36
Persamaan (2.14).....	37
Persamaan (2.15)	37
Persamaan (2.16)	37
Persamaan (2.17).....	37
Persamaan (2.18).....	38
Persamaan (2.19).....	38
Persamaan (2.20).....	38
Persamaan (2.21).....	38
Persamaan (2.22).....	38

DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi <i>Case Folding</i>	109
Kode Program 5.2 Implementasi <i>Data cleaning</i>	110
Kode Program 5.3 Implementasi Normalisasi Bahasa	110
Kode Program 5.4 Implementasi <i>Stopword Removal</i>	111
Kode Program 5.5 Implementasi <i>Stemming</i>	111
Kode Program 5.6 Implementasi Tokenisasi	112
Kode Program 5.7 Implementasi Pencarian Fitur Kata	113
Kode Program 5.8 Implementasi perhitungan nilai tf dan wtf	113
Kode Program 5.9 Implementasi perhitungan nilai df dan idf	114
Kode Program 5.10 Implementasi perhitungan TF-IDF	115
Kode Program 5.11 Implementasi perhitungan pembobotan <i>Lexicon Based Features</i>	116
Kode Program 5.12 Implementasi pembentukan matriks dan transposisi	118
Kode Program 5.13 Implementasi perhitungan kernel	119
Kode Program 5.14 Implementasi perhitungan matriks hessian	119
Kode Program 5.15 Implementasi perhitungan <i>sequential learning</i>	120
Kode Program 5.16 Implementasi perhitungan <i>bias</i>	121
Kode Program 5.17 Implementasi perhitungan <i>testing</i>	122
Kode Program 5.18 Implementasi <i>confusion matrix</i>	123
Kode Program 5.19 Implementasi perhitungan akurasi	124
Kode Program 5.20 Implementasi perhitungan waktu komputasi	124

DAFTAR LAMPIRAN

Lampiran A Hasil Wawancara Pakar Psikologi	140
Lampiran B Hasil Wawancara Pakar Bahasa Indonesia	143
Lampiran C Kamus Normalisasi	145
Lampiran D Kamus <i>Lexicon</i>	155
Lampiran E <i>Stoplist</i>	159
Lampiran F Perhitungan nilai <i>tf</i> dan <i>df</i> pada data latih dan data uji	165
Lampiran G Perhitungan nilai $W_{tf,d}$ dan <i>idf</i> pada data latih dan data uji	169
Lampiran H Perhitungan TF-IDF pada data latih dan data uji	173
Lampiran I Fitur term gabungan pembobotan TF-IDF dan <i>Lexicon</i>	177
Lampiran J Dataset Komentar Instagram Serta Kelas Aktual Berdasarkan Pakar	181
Lampiran K Daftar Kelas Aktual dan Kelas Prediksi	218
Lampiran L Grafik Akurasi, <i>Precision</i> , <i>Recall</i> , <i>F-Measure</i> pada tahap pengujian <i>Lexicon Based Features</i>	222



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Informasi dan komunikasi merupakan kebutuhan primer bagi manusia, sehingga informasi dan komunikasi bergerak cepat seiring dengan perkembangan zaman. Hal tersebut dibantu dengan perkembangan teknologi, persebaran informasi, dan komunikasi yang tak lagi terhitung hari namun hanya terhitung dengan detik. Perkembangan teknologi memengaruhi kebutuhan informasi yang semakin meningkat. Berkembangnya teknologi telah menggeser alat informasi dan komunikasi tradisional yang kita kenal seperti radio, televisi, media cetak, dan surat. Penyebaran informasi maupun untuk melakukan komunikasi tidaklah sulit, karena dapat dilakukan dengan sekejap mata tanpa mengenal jarak, batas, ruang, maupun waktu.

Kehadiran teknologi internet telah mempermudah hidup manusia dalam menggali informasi dan komunikasi. Dampak dari internet tanpa disadari sangat melekat pada manusia, sehingga setiap hari dalam hidup manusia telah bergantung pada teknologi tersebut. Banyak dampak dapat dimanfaatkan oleh beberapa orang yang berupaya untuk mengembangkan suatu media komunikasi maupun untuk menguntungkan diri sendiri dan suatu kelompok.

Media sosial adalah salah satu contoh dari kemajuan teknologi. Munculnya media sosial berdampak pada pengguna media sosial yang terus bertambah disetiap harinya. Tak hanya kalangan orang dewasa, anak-anak yang masih dibangku sekolah dasar-pun telah mengenal media sosial. Namun, tanpa disadari dengan adanya media sosial yang tidak mengenal usia penggunanya. Pengguna media sosial yang berasal dari seluruh kalangan usia dapat memberikan berbagai macam permasalahan. *Bullying* merupakan hal yang kini kerap terjadi di dunia tak terkecuali Indonesia. Permasalahan tersebut merupakan peringatan baik bagi pengguna media sosial, orangtua, kerabat pengguna, dan pemerintah. Datangnya teknologi memberikan dampak positif dan negatif. Permasalahan utama yang kini telah ditimbulkan ialah tindakan *bullying* yang terjadi di dunia maya atau dikenal sebagai perundungan (*cyberbullying*).

Instagram adalah salah satu contoh media sosial yang kita ketahui sebagai media untuk berbagi foto. Layanan yang berguna dalam membagikan foto maupun video ini telah memikat banyak orang, sehingga terdapat kenaikan 100 juta pengguna terhitung sejak April 2017. Pada bulan September 2017 telah tercatat pengguna aktif Instagram mencapai kurang lebih 800 juta, hal ini diungkapkan oleh Carolyn Everson selaku *Vice President Global Marketing Solutions* Facebook (Yusuf, 2017). Tidak hanya orang dewasa namun anak-anak turut menggunakan media sosial ini. Pemanfaatan yang beragam dari pengguna Instagram dibuat sebagai akun pribadi, baik untuk orang biasa maupun artis hingga menjadi sarana bisnis perseorangan. Permasalahan yang sering terjadi adalah tindakan *cyberbullying* pun muncul pada Instagram. Tak banyak pengguna

Instagram yang menyadari bahwa ulasan atau komentar yang dilontarkan pada seseorang ataupun golongan merupakan tindakan *bullying*.

Tindakan *cyberbullying* adalah dampak dari kebebasan berinteraksi di media sosial. Hal ini dilakukan dengan menggunakan media teks maupun gambar secara visual yang menjadi sarana komunikasi. Berdasarkan data yang telah dihimpun oleh *KompasTekno* di tahun 2017, Instagram merupakan media sosial nomor satu yang berkontribusi dalam tindakan *cyberbullying*. Dari data tersebut didapatkan bahwa 42% remaja yang berusia 12 hingga 20 tahun telah menjadi korban *cyberbullying* (Bohang, 2017). Salah satu contoh *cyberbullying* telah dialami oleh seorang remaja asal Indonesia yang berusia 13 tahun di bulan April 2017. Hal tersebut terjadi ketika remaja itu berniat untuk membagikan tiket secara gratis kepada khalayak umum dengan beberapa syarat. Namun hal ini memberikan dampak negatif untuk dirinya. Berbagai kabar tidak baik tersebar secara cepat di dunia maya dan pada akhirnya muncul komentar negatif yang bersifat menyakiti hati anak. Sebagai dampaknya remaja yang berkeseharian riang, menjadi murung dan selalu menangis (Muttya, 2017). Permasalahan *bullying* pada kolom komentar Instagram menjadi hal yang penting untuk dikaji sebagai pemrosesan teks. Dari berbagai macam komentar yang bersifat menyakiti tersebut telah memengaruhi dampak psikis seseorang, sehingga baik untuk diteliti.

Analisis sentimen sangat diperlukan dalam menyaring komentar-komentar di media sosial. Analisis sentimen pada komentar dilakukan untuk mengetahui komentar yang bersifat negatif dan komentar yang bersifat positif. Dari analisis tersebut dapat dilakukan tindakan preventif baik untuk korban maupun pelaku.

Berbagai macam penelitian sosial mengenai *cyberbullying* telah dilakukan. Sebagai contohnya salah satu penelitian yang memberikan hasil bahwa tindakan *bullying* (perundungan) dapat dibagi menjadi dua yaitu, berupa mikro dan makro. Penelitian tersebut memberikan hasil bahwa beberapa diantaranya jejak dari pelaku *bullying* dapat diidentifikasi namun beberapa pelaku lainnya tidak dapat diidentifikasi karena menggunakan akun palsu (Nasrullah, 2015). Penelitian lain yang telah dilakukan ialah proses analisis sentimen yang dilakukan pada objek twitter dengan menerapkan metode *Maximum Entropy* dan *Support Vector Machine* (SVM). Berdasarkan penelitian tersebut hasil akurasi dengan mengimplementasikan metode SVM ialah 86,81% dengan waktu proses 1688 detik dan menggunakan *7 fold cross validation* pada tipe kernel Sigmoid (Putranti & Winarko, 2014). Penelitian serupa membahas mengenai perbandingan hasil dengan mengimplementasikan metode *Support Vector Machine-Particle Swarm Optimization* dan metode yang hanya menggunakan SVM. Metode SVM dengan *Particle Swarm Optimization* (PSO) telah memberikan hasil akurasi sebesar 73,33%, sedangkan jika menerapkan algoritme SVM-PSO hasil akurasi mencapai 76% (Yunita, 2016). Penelitian mengenai metode SVM telah dilakukan pula dengan membandingkan hasil akurasi dengan metode Naïve Bayes dan Maximum Entropy. Berdasarkan penelitian tersebut didapatkan hasil akurasi terbaik ialah kurang lebih sebesar 98% pada metode SVM (K & Shetty, 2017).

Dari berbagai referensi penelitian yang telah dilakukan, metode *Support Vector Machine* menjadi salah satu pilihan metode yang akan digunakan peneliti dalam analisis sentimen. Hal ini menjadi topik peneliti untuk memberikan solusi terhadap *cyberbullying* yang terjadi di media sosial Instagram.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi algoritme *Support Vector Machine* mampu dalam membantu analisis sentimen pada komentar Instagram?
2. Bagaimana tingkat akurasi yang diberikan oleh algoritme *Support Vector Machine* pada analisis sentimen?

1.3 Tujuan

Dari rumusan masalah yang telah diberikan, penelitian ini dilakukan dengan tujuan:

1. Menerapkan algoritme *Support Vector Machine* untuk menganalisis teks positif dan negatif pada komentar Instagram.
2. Mengetahui tingkat akurasi dari hasil perhitungan algoritme *Support Vector Machine* untuk klasifikasi teks positif dan negatif pada komentar Instagram.

1.4 Manfaat

Manfaat dari hasil penelitian ini adalah:

1. Membantu untuk mengetahui komentar positif dan komentar negatif.
2. Mengetahui komentar yang mengandung konten *cyberbullying*.
3. Memberi pengetahuan bagi pengguna Instagram untuk selalu berhati-hati dalam memberikan komentar terhadap sesuatu.

1.5 Batasan Masalah

Untuk melakukan penelitian secara spesifik dan jelas, diperlukan batasan yang diterapkan pada penelitian ini. Batasan yang dilakukan pada penelitian, adalah:

1. Komentar Instagram yang dianalisis adalah komentar berbahasa Indonesia
2. *Sentimen analysis* dilakukan dengan metode klasifikasi
3. Algoritme yang diterapkan adalah *Support Vector Machine*.
4. Komentar Instagram akan dikelompokkan menjadi komentar positif dan komentar negatif.
5. Data komentar Instagram diambil secara *offline* (luar jaringan).

6. Pengujian implementasi dari algoritme *Lexicon Based Features* dilakukan dengan menggunakan dua cara yaitu, normalisasi *min-max* dan perhitungan skor sentimen. Pada perhitungan skor sentimen tidak memperhitungkan syarat POS *Tags* dan perhitungan kalimat negatif.

1.6 Sistematika Pembahasan

Bab 1 Pendahuluan

Bab ini membahas mengenai alasan utama dari suatu topik yang diangkat sebagai objek penelitian. Alasan utama tersebut dituangkan dalam latar belakang masalah, rumusan masalah dilakukannya penelitian, tujuan dilakukan penelitian, manfaat dari hasil penelitian, batasan masalah yang akan dibahas pada penelitian, dan sistematika pembahasan.

Bab 2 Landasan Kepustakaan

Membahas mengenai teori-teori dasar sebagai referensi ataupun pernyataan pendukung dilakukannya penelitian. Teori yang dibahas pada landasan kepastakaan meliputi analisis sentimen, *text mining*, media sosial, *cyberbullying*, klasifikasi, dan metode yang digunakan dalam penelitian.

Bab 3 Metodologi

Pada bab ini menjelaskan tentang metode penelitian yang digunakan dengan pendekatan dalam tahapan penelitian baik dari kebutuhan sistem, perancangan sistem, dan pengumpulan data.

Bab 4 Perancangan

Menjelaskan tentang seluruh perancangan sistem yang akan diterapkan pada penelitian, yaitu analisis sentimen dengan metode klasifikasi pada komentar Instagram.

Bab 5 Implementasi

Pada bab ini metode yang diterapkan untuk memberikan solusi akan dijelaskan secara bertahap. Tahapan dari jalannya algoritme *Support Vector Machine* akan dijelaskan untuk memberikan solusi pada sistem, sehingga memberikan suatu informasi yang bermanfaat.

Bab 6 Pengujian dan Analisis

Pengujian terhadap sistem akan dijelaskan pada bab ini, sehingga dapat mengetahui tingkat keberhasilan sistem dalam memberikan suatu solusi. Analisis dilakukan pada hasil pengolahan data, baik data masukan maupun data yang diproses oleh sistem.

Bab 7 Penutup

Setiap proses dan hasil yang didapatkan selama penelitian berlangsung akan dibahas secara ringkas pada bab kesimpulan. Saran yang diperoleh dari hasil penelitian akan dicantumkan dengan harapan jika terdapat penelitian serupa akan memberikan hasil yang jauh lebih baik dari penelitian sebelumnya.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab landasan kepastakaan disajikan beberapa teori pendukung untuk penelitian. Teori yang akan dibahas mengenai analisis sentimen, *text mining*, *cyberbullying*, Instagram, dan algoritme *Support Vector Machine*.

2.1 Kajian Pustaka

Pada sub-bab kajian pustaka, akan mengacu pada penelitian-penelitian sebelumnya yang berguna dalam mendukung pelaksanaan penelitian. Penelitian terkait yang mengacu pada metode klasifikasi SVM adalah penelitian pada analisis sentimen untuk mengukur tingkat kepuasan pengguna penyedia layanan telekomunikasi pada Twitter. Penelitian tersebut menggunakan jumlah total 300 data, dimana 70% data digunakan sebagai data latih sedangkan 30% data menjadi data uji. Dengan menggunakan metode SVM dan *Lexicon Based Features* didapatkan hasil akurasi sebesar 79%, *precision* sebesar 65%, *recall* sebesar 97%, dan *F-Measure* sebesar 78%. Namun penambahan metode *Lexicon Based Features* memiliki pengaruh pada analisis sentimen, yaitu tingkat akurasi yang lebih rendah dibandingkan ketika metode *Lexicon Based Features* tidak diimplementasikan (Rofiqoh, 2017). Penurunan tingkat akurasi dengan diimplementasikan metode *Lexicon Based Features* berpengaruh terhadap nilai matriks Hessian sehingga dapat menurunkan nilai *support vector*.

Penelitian lainnya adalah penelitian yang berobjek twitter dengan mengkombinasikan metode SVM dan *Lexicon Based Features*. Penelitian yang bertujuan untuk memberikan informasi mengenai kualitas program televisi melalui komentar yang dilontarkan pada media sosial Twitter. Data diambil dari lima program tv dengan yaitu, Program A 300 *tweet*, Program B 300 *tweet*, Program C 500 *tweet*, Program D 500 *tweet*, Program E 600 *tweet*. Dengan metode *Support Vector Machine* didapatkan nilai akurasi tertinggi terdapat pada program A dan B dengan total 300 *tweet*, dengan nilai akurasi yang rata-rata adalah 70% meskipun terdapat perubahan komposisi data (Tiara, et al., 2015).

Penelitian lain dilakukan dalam upaya menganalisa dan memberikan bukti bahwa SVM menghasilkan akurasi tinggi dalam analisis sentimen pada *product review*. *Product review* adalah tanggapan dan pendapat masyarakat terhadap hasil produk atau jasa pada bisnis di internet. Penelitian ini adalah *review* dari berbagai *paper*. Dalam paper ini memberikan analisa terkait penelitian-penelitian analisis sentimen. Data menunjukkan bahwa analisis sentimen memberikan dampak yang besar dalam terkait kualitas hasil produk dan jasa dalam sebuah komunitas masyarakat. Hasil dari perbandingan metode dalam *product review*, didapatkan SVM merupakan metode terbaik yang diterapkan dalam berbagai data dengan tingkat akurasi yang mencapai kurang lebih 98% (K & Shetty, 2017)

Penelitian lain yang berupa analisis sentimen dan mengimplementasikan metode SVM dilakukan dalam upaya untuk mengetahui opini publik terhadap artis. Dari penelitian tersebut, telah menggunakan 300 *review* yang dijadikan

sebagai data latih. 300 *review* tersebut terdiri dari 150 *review* negatif dan 150 *review* positif. Kumpulan data tersebut diolah dengan beberapa tahapan yang dimulai dari tahap *Pre-processing* teks yang kemudian baru dihitung dengan metode SVM dan *Particle Swarm Optimization*. Penelitian tersebut memberikan akurasi berbentuk *confusion matrix* dan kurva ROC. Besar akurasi jika menggunakan algoritme *Support Vector Machine* saja adalah 73,33% dan nilai AUC sebesar 0,779. Namun dengan adanya optimasi *Particle Swarm Optimization* nilai akurasi meningkat menjadi 76% dan AUC menjadi 0,794 (Yunita, 2016). Selain pada opini publik terhadap artis, dilakukan pula penelitian terhadap wacana politik. Penelitian yang membandingkan dua metode yaitu SVM dan Naïve Bayes memberikan tingkat akurasi yang berbeda pada data yang sama. Hasil perbandingan SVM dan *Naive Bayes* pada penelitian ini menunjukkan bahwa SVM memiliki akurasi yang lebih tinggi yaitu 90.50% sedangkan *Naive Bayes* memiliki akurasi sebesar 59.58% (Hidayat, 2015).

Penelitian lain yang mendukung penelitian ini merupakan penelitian pada bidang sosial yang membahas mengenai *cyberbullying*. Penelitian tersebut menggunakan *ranking method* yaitu berupa *page rank* dan *Hyperlink Induced Topic Search* (HITS) untuk mengidentifikasi korban dan pelaku *cyberbullying*. Penelitian ini membantu dalam mendeteksi pesan-pesan yang mengandung *cyberbullying* maupun tidak dengan menerapkan *Probabilistic Latent Semantic Analysis* (PLSA). Hasil terbaik saat proses klasifikasi didapatkan ketika nilai fitur adalah 500 dengan hasil akurasi 99.20% (Nahar, et al., 2012). Kajian pustaka secara mengenai penelitian-penelitian terkait ditunjukkan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No.	Pustaka	Objek dan Metode	Hasil
1.	(Rofiqoh, 2017)	Tingkat kepuasan pengguna pelayanan telekomunikasi pada twitter dengan menggunakan metode <i>Support Vector Machine</i> Dan <i>Lexicon Based Features</i>	Hasil pengujian memiliki nilai optimum dengan parameter nilai <i>degree</i> kernel adalah 2, learning rate sebesar 0.0001 dan iterasi maksimum adalah 50. Hasil akurasi yaitu 79%, <i>precision</i> sebesar 65%, <i>recall</i> sebesar 97%, dan <i>f-measure</i> sebesar 78%.
2.	(Tiara, et al., 2015)	Kualitas program televisi dengan implementasi metode Kombinasi <i>Lexicon-based</i> dan SVM	Dari lima program TV yang direview dengan jumlah <i>Tweets</i> yang berbeda, didapatkan akurasi tertinggi terletak pada program A dan B yang memiliki 300 <i>Tweet</i> . Hasil akurasi yang diberikan berkisar pada nilai 70%.

Tabel 2.1 Kajian Pustaka (Lanjutan)

No.	Pustaka	Objek dan Metode	Hasil
3.	(K & Shetty, 2017)	Review Produk dengan metode <i>Naive Bayes</i> , <i>Support Vector Machine</i> , dan <i>Maximum Entropy</i>	Dengan dilakukannya perbandingan terhadap ketiga metode tersebut pada tiga jenis dataset, didapatkan hasil terbaik pada metode <i>Support Vector Machine</i> . Tingkat akurasi yang diberikan berkisar dinilai 98%.
4.	(Yunita, 2016)	Berita artis dengan menggunakan metode <i>Support Vector Machine</i> (SVM) dan <i>Particle Swarm Optimization</i> (PSO)	Akurasi menggunakan algoritme <i>Support Vector Machine</i> saja adalah 73,33% dan nilai AUC sebesar 0,779. Namun dengan adanya optimasi <i>Particle Swarm Optimization</i> nilai akurasi meningkat menjadi 76% dan AUC menjadi 0,794.
5.	(Hidayat, 2015)	Wacana politik pada media masa <i>online</i> dengan metode <i>Naive Bayes</i> dan <i>Support Vector Machine</i>	Hasil perbandingan SVM dan <i>Naive Bayes</i> pada penelitian ini menunjukkan bahwa SVM memiliki akurasi yang lebih tinggi yaitu 90.50% sedangkan <i>Naive Bayes</i> memiliki akurasi sebesar 59.58%.
6.	(Nahar, et al., 2012)	Deteksi <i>Cyberbullying</i> dengan metode <i>Hyperlink Induced Topic Search</i> (HITS) dan <i>Probabilistic Latent Semantic Analysis</i> (PLSA)	Kestabilan akurasi terjadi ketika nilai fitur terdapat diangkat 500 hingga 4000. Hasil akurasi terbaik sebesar 99.20% ketika nilai fitur terletak pada nilai 500.

2.2 Analisis Sentimen

Analisis sentimen merupakan salah satu cabang ilmu dari *text mining*, *natural language program*, dan *artificial intelegence*. Proses yang dilakukan oleh analisis sentimen untuk memahami, mengekstrak, dan mengolah data teks secara

otomatis sehingga menjadi suatu informasi yang bermanfaat (Akbari, et al., 2012). Selain itu analisis sentimen merupakan bidang ilmu yang menganalisis pendapat, sikap, evaluasi, dan penilaian terhadap suatu peristiwa, topik, organisasi, maupun perseorangan (Liu, 2012).

Analisis sentimen disebut juga sebagai *opinion mining* yang berguna dalam pengelolaan bahasa alami, komputasi linguistik, dan *text mining*. Tujuan yang dilakukan analisis sentimen adalah untuk menentukan perilaku ataupun opini yang diberikan oleh penulis pada topik tertentu. Perilaku tersebut dapat mengindikasikan penilaian serta alasan dan kondisi kecenderungan (Basari, 2013). Disebutkan pula bahwa analisis sentimen memiliki tugas dasar untuk mengelompokkan teks pada kalimat maupun dokumen. Hasil yang diberikan oleh analisis sentimen bisa berupa teks bersifat positif, negatif, dan netral. Tidak hanya mengelompokkan teks secara positif, negatif, dan netral, analisis sentimen dapat menyatakan perasaan emosional, gembira, sedih, dan marah (Manalu, 2014).

2.3 Text Mining

Text mining adalah ilmu yang bertujuan untuk memproses teks agar menjadi informasi yang diperoleh dari peramalan pola dan kecenderungan melalui pola statistik. Teks yang diolah bisa berupa teks terstruktur dan teks tidak terstruktur. *Text mining* mengacu pada *information retrieval*, *data mining*, *machine learning*, statistik dan komputasi linguistik (Jiawei, et al., 2012). *Text mining* bertujuan untuk menganalisis pendapat, sentimen, evaluasi, sikap, penilaian, emosi seseorang sehingga dapat diketahui apakah berkenaan dengan suatu topik, layanan, organisasi, individu, atau kegiatan tertentu (Liu, 2012). Penggunaan dari *text mining* dilakukan untuk klusterisasi, klasifikasi, *information retrieval*, dan *information extraction* (Berry & Kogan, 2010).

2.3.1 Pre-processing

Pre-processing merupakan tahap awal dari *text mining* untuk mengubah data sesuai dengan format yang dibutuhkan. Proses ini dilakukan untuk menggali, mengolah dan mengatur informasi dan untuk menganalisis hubungan tekstual dari data terstruktur dan data tidak terstruktur (Nugroho, 2016). Persiapan data dilakukan untuk diolah pada *knowledge discovery*. Tahapan dari *Pre-processing* meliputi *case folding*, *data cleaning*, normalisasi bahasa, *stopword removal*, *stemming*, tokenisasi.

2.3.1.1 Case Folding

Tahap awal adalah *case folding* yang bertujuan untuk mengubah setiap bentuk kata menjadi sama. Hal ini dilakukan dengan mengubah kata menjadi *lower case* atau huruf kecil.

2.3.1.2 Data Cleansing

Data cleaning merupakan proses pembersihan kata dengan menghilangkan delimiter koma (,), titik (.), dan tanda baca lainnya. Pembersihan kata bertujuan untuk mengurangi *noise*.

2.3.1.3 Normalisasi Bahasa

Pada tahap *pre-processing* dilakukan normalisasi bahasa terhadap kata tidak baku. Tahapan ini bertujuan untuk mengembalikan bentuk penulisan dari masing-masing kata yang sesuai dengan KBBI. Proses ini dilakukan dengan mencocokkan setiap kata pada dokumen data latih maupun data uji dengan kata yang ada pada kamus bahasa tidak baku (Darma, 2017).

2.3.1.4 Stopword Removal

Stopword merupakan daftar kata umum yang tidak memiliki arti penting dan tidak digunakan. Pada proses ini kata umum akan dihapus untuk mengurangi jumlah kata yang disimpan oleh sistem (Manning, et al., 2009).

2.3.1.5 Stemming

Langkah selanjutnya dalam *Pre-processing* teks adalah melakukan *stemming*. Tahapan ini merupakan proses untuk mencari *root* (kata dasar) dari kata hasil *stopword removal* (*filtering*). Terdapat dua aturan dalam melakukan *stemming* yaitu dengan pendekatan kamus dan pendekatan aturan (Utomo, 2013).

Dalam pendekatan aturan untuk *stemming* yaitu dengan cara menghilangkan imbuhan pada kata yang telah melalui proses *stopword removal* (*filtering*). Aturan-aturan yang telah dibuat dalam proses *stemming* pada Bahasa Indonesia (Asian, 2007), antara lain:

1. Imbuhan kata (Afiks) yang terdiri dari prefiks, suffiks, infiks, dan konfiks. Penjelasan dari setiap imbuhan diketahui sebagai berikut:
 - a. Prefiks, imbuhan yang terletak pada awal kata. Prefiks merupakan imbuhan yang kompleks karena beberapa kata data dapat melebur dan berubah ketika diberi imbuhan. Prefiks terdiri dari “se-”, “ke-”, “di-”, “ter-”, “ber-”, “per-”, “pe-”, dan “me-”.
Contoh: Ber-tatap.
 - b. Suffiks, merupakan imbuhan yang terletak pada akhir kata. Contoh dari suffiks adalah “-lah”, “-kah”, “-pun”, “-ku”, “-mu”, “-nya”, “-i”, “-kan”, “-an”.
Contoh: Biar-kan.
 - c. Konfiks, imbuhan ini merupakan imbuhan gabungan dari prefiks dan suffiks. Imbuhan terdapat pada awal dan akhir kata.
Contoh: per-rasa-an.
 - d. Infiks, imbuhan yang terletak pada tengah kata.
Contoh: k-em-ilau.
2. Ambiguitas kata, pada satu kata dapat memiliki dua makna.
Contoh: Berikan → Ber-ikan, dan Berikan → Beri-kan.

3. Perulangan kata, contoh: baju-baju.

Proses *stemming* digunakan dengan menggunakan stemmer Sastrawi yang merupakan *library*. *Library* yang digunakan adalah berbahasa Indonesia yang berbasis Java, C, PHP, dan Python. *Stemmer* Sastrawi merupakan jenis *stemming* yang berbasis algoritma Nazief dan Andriani (Chrismanto & Lukito, 2017).

2.3.1.6 Tokenisasi

Tokenisasi adalah proses untuk memotong document manjadi pecahan kecil yang dapat berupa bab, sub-bab, paragraf, kalimat, dan kata (token). Pada proses ini akan menghilangkan *whitespace*.

2.3.2 Pembobotan TF-IDF

Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah metode yang digunakan untuk menghitung bobot setiap kata yang telah diekstrak. Penggunaan metode ini umumnya dilakukan untuk menghitung kata umum yang ada pada *information retrieval*. Model pembobotan TF-IDF merupakan metode yang mengintegrasikan model *term frequency* (*tf*) dan *inverse document frequency* (*idf*), dimana *term frequency* (*tf*) merupakan proses untuk menghitung jumlah kemunculan term dalam satu dokumen dan *inverse document frequency* (*idf*) digunakan untuk menghitung term yang muncul di berbagai dokumen(komentar) yang dianggap sebagai term umum, yang dinilai tidak penting (Akbari, et al., 2012).

Proses awal yang dilakukan dalam pembobotan TF-IDF dilakukan dengan menghitung *term frequency* $tf_{t,d}$. Dimana t menunjukkan term dalam dokumen d yang berfungsi untuk menunjukkan kemunculan term t pada dokumen d . Hal ini berpengaruh dalam bobot term yang akan semakin tinggi ketika banyak term yang muncul dalam suatu dokumen. Nilai dari tf akan dihitung bobotnya dengan rumus *weighting term frequency* (W_{tf}). Rumus tersebut ditunjukkan pada persamaan 2.1.

$$W_{tf_{t,d}} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

Banyaknya kata yang muncul pada dokumen, umumnya merupakan nilai *term frequency* dari kata yang tidak penting. Untuk menghindari pembobotan pada kata tidak penting maka digunakan pembobotan *document frequency* yang bermaksud untuk menghitung jumlah dokumen yang mengandung term t .

Dari nilai term pada setiap dokumen yang telah ditemukan akan dilakukan proses kebalikan dari pembobotan *document frequency*. Proses pembobotan ini disebut dengan *inverse document frequency*, yang menyatakan bahwa frekuensi dari term yang rendah pada banyak dokumen akan memberikan bobot paling tinggi. Perhitungan ini ditunjukkan dengan rumus persamaan 2.2.

$$idf_t = \log_{10} \frac{N}{df_t} \quad (2.2)$$

Perhitungan pembobotan TF-IDF merupakan perkalian yang dilakukan dari pembobotan term *frequency* dengan *inverse document frequency*. Hal ini ditunjukkan pada rumus persamaan 2.3.

$$W_{t,d} = W_{tf_{t,d}} \times idf_t \quad (2.3)$$

Keterangan:

$W_{tf_{t,d}}$ = bobot kata dalam setiap dokumen

$tf_{t,d}$ = jumlah kemunculan kata t pada dokumen d

N = jumlah dokumen pada kumpulan dokumen

df = jumlah dokumen yang mengandung term

idf_t = bobot inverse dari nilai df

$W_{t,d}$ = pembobotan TF-IDF

2.4 Media Sosial

Media sosial merupakan bagian dari media baru (*new media*), dimana media baru menawarkan digitalisasi, konvergensi interaktifitas dan jaringan baru dalam membuat pesan dan menyampaikan pesan. Media sosial memiliki kekuatan dalam membentuk opini masyarakat dan para pengguna media sosial dapat berpartisipasi, berbagi, dan menciptakan sesuatu hal yang berupa blog, forum dan dunia virtual. Media sosial tidak sama dengan media massa *online* karena penggunaan yang kuat dalam membentuk opini yang berkembang di masyarakat. Pengaruh yang kuat dari media sosial mampu membentuk sikap dan perilaku publik atau masyarakat (Ardianto, 2011).

Media sosial merupakan media *online* yang mempermudah pengguna untuk berbagi informasi, menciptakan konten yang ingin disampaikan kepada orang lain. Media sosial tidak mengenal ruang dan waktu sehingga penyampaian komentar dan masukan dapat dilakukan secara cepat. Penggunaan media sosial telah menjadi kebutuhan primer untuk mayoritas individu di dunia karena penggunaannya telah menjadi bagian hidup sehari-hari.

Berbagai faktor pendorong penggunaan media sosial telah menjadi alasan dasar mengapa media sosial dianggap penting. Faktor pendorong tersebut telah dikatakan oleh McQuail (2000) seperti berikut ini:

1. Faktor informasi.
2. Identitas personal.
3. Faktor integratif dan interaksi sosial.
4. Faktor hiburan.

2.4.1 Instagram

Instagram merupakan salah satu media sosial yang digemari pada zaman ini. Aplikasi *smartphone* ini ditujukan untuk berbagi foto ini, telah mencapai angka statistik yang tinggi disetiap harinya. Berdasarkan data pada tahun 2017, telah didapatkan informasi mengenai pengguna aktif Instagram di Indonesia yang mencapai 45 juta orang. Pengguna aktif Instagram yang telah diakumulasikan pada tahun 2017, telah mencapai 700 juta pengguna aktif disetiap bulannya (Adi & Hidayat, 2017). Namun perkembangan pengguna instagram semakin meningkat seiring dengan fitur-fitur tambahan yang diberikan. Pada bulan September 2017 telah terhitung peningkatan pengguna sebesar 100juta, sehingga jika diakumulasikan terdapat 800 juta pengguna disetiap bulannya (Yusuf, 2017).

Instagram memberikan fasilitas para penggunanya untuk mengambil foto, memberikan *filter* secara digital pada foto, dan kemudian untuk dibagikan ke berbagai media sosial termasuk di akun Instagram tersebut. Seiring dengan perkembangan teknologi, Instagram turut berkembang. Tidak hanya dapat membagikan foto, kini Instagram telah menambahkan fasilitas pengguna untuk mengambil video. Fungsionalitas utama Instagram yang dapat anda rasakan ialah memiliki akun pribadi Instagram, dapat mengikut dan tidak mengikuti (*follow-unfollow*) akun Instagram lainnya, menyukai foto orang lain, menyimpan foto orang lain, dan memberi komentar.

2.5 Cyberbullying

Seiring dengan perkembangan zaman, teknologi pun ikut berkembang. Berkembangnya teknologi memberikan pengaruh terhadap kehidupan sosial. Seperti pada tindakan *bullying*. Mulanya tindakan *bullying* menyerang secara fisik maupun psikologi secara langsung, namun kini tindakan tersebut dapat dilakukan pada dunia maya yang dikenal dengan *cyberbullying*. *Cyberbullying* merupakan suatu tindakan tidak menyenangkan yang dilakukan secara sengaja dan terus menerus melalui teks elektronik (Stauffer, et al., 2012).

Berdasarkan sumber lain mengatakan bahwa *cyberbullying* merupakan tindakan *bullying* yang dilakukan pada dunia *cyber*. Dimana dalam tindakannya dapat dibagi menjadi beberapa kriteria dan dilakukan secara berulang-ulang. Terdapat beberapa aspek yang memenuhi pada *cyberbullying* seperti, *flaming*, *harrasment*, *cyberstalking*, dan lainnya (Pratiwi, 2017).

Cyberbullying memiliki jenis dengan masing-masing definisinya (Willard, 2007), seperti yang diketahui sebagai berikut:

1. *Flaming*, perkelahian dengan media pesan elektronik dengan bahasa yang menunjukkan amarah dan bersifat kasar.
2. *Harassment*, mengirim pesan jahat, kasar, dan menghina secara berulang kali.
3. *Denigration*, mengunggah gossip atau rumor mengenai seseorang dengan tujuan menjatuhkan reputasi orang tersebut.

4. *Impersonation*, berpura-pura menjadi orang lain dengan tujuan mengunggah mengenai orang (yang ditiru) sehingga membahayakan orang tersebut.
5. *Outing*, menyebarkan rahasia atau informasi mengenai seseorang, membagikan gambar secara *online*.
6. *Trickery*, mengungkapkan informasi seseorang atau mengenai hal yang memalukan dan dibagikan secara *online*.
7. *Exclusion*, tidak menganggap seseorang pada grup *online* dengan secara sengaja.
8. *Cyberstalking*, perilaku berulang yang dapat bersifat melecehkan, menghina hingga mengancam sehingga mengakibatkan ketakutan.

Cyberbullying merupakan bentuk serangan yang bersifat dengki untuk memberikan kepuasan atau kesenangan pelaku dengan cara menyiksa orang lain. Perempuan lebih banyak terlibat dalam *cyberbullying*, baik sebagai pelaku maupun sebagai korban. Dimana 50% korban *cyberbullying* umumnya tidak mengetahui identitas pelaku bully meski hanya *gender*. *Cyberbullying* merupakan tindakan yang mengakibatkan serangan psikologi, emosional, dan trauma sosial (Kowalski & Limber, 2007).

Dampak dari perbuatan *cyberbullying* dapat memengaruhi korban. Rasa kepercayaan diri korban akan menurun, jika korban masih dalam bangku sekolah akan berpengaruh pada penurunan peringkat disekolah. Selain muncul perasaan gelisah dan depresi sehingga korban merasa hidupnya tidaklah nyaman. Dengan seringnya korban mengalami *cyberbullying*, maka lingkungan korban akan semakin menjauh. Hal ini menjadi permasalahan serius ketika *cyberbullying* menjadi hal yang lebih berbahaya daripada *bully* pada biasanya. Karena efek yang dampak yang lebih berbahaya ketika perasaan takut dan depresi berubah menjadi pemarah dan frustrasi, bahkan korban dapat melakukan bunuh diri (Hinduja & J.Patchin, 2010).

Pemilihan kata yang dilontarkan menjadi kunci utama apakah seseorang mengarah pada tindakan *bullying* atau tidak. Contoh kata yang tergolong pada kata positif dan kata negatif digambarkan pada tabel berikut:

Tabel 2.2 Contoh kata positif

Positif	
Cekatan	Berani
Sesuai	Suci
Indah	Teliti
Wibawa	Unggul
Sabar	Pesona



Tabel 2.3 Contoh kata negatif

Negatif	
Abnormal	Pengecut
Aneh	Banci
Bodoh	Buruk
Gila	Jelek
Khianat	Munafik

2.6 Lexicon Based Features

Lexicon Based Features merupakan suatu kesepakatan dalam pendekatan yang meliputi frase, bentuk ekspresi, atau konten yang berupa teks yang umumnya terdapat pada obrolan, dialog, *post*, *review*, dan lainnya. Namun tak jarang bahwa hanya meliputi frase sederhana, tetapi dapat berupa kalimat majemuk yang kompleks. Metode ini berguna dalam mengekstrak sentimen dari berbagai sumber media teks yang bekerja dengan mengkombinasikan *lexical knowledge* dan klasifikasi teks (Melville, et al., 2011). Meski pendekatan ini tepat, mudah dipahami, dan mudah untuk diterapkan pada *machine learning* tetapi memiliki kekurangan dimana memerlukan waktu yang lebih banyak. Hal disebabkan karena perlunya keterlibatan manusia dalam proses analisis teks. *Lexicon Based Features* merupakan pendekatan yang menggunakan suatu kamus sentimen berisi kata positif dan kata negatif yang dibandingkan dan dicocokkan dengan kata pada kalimat untuk diketahui tingkat polaritasnya (Peng, 2011).

Selain itu *Lexicon Based Features* dapat dikatakan sebagai fitur berdasarkan lexicon atau fitur berdasarkan *knowledge* yang terdapat pada kamus. *Lexicon Based Features* ini berfokus pada pengambilan suatu opini pada teks dimana akan dicari tahu tingkat polaritas berdasarkan kecocokan kata dari lexicon. Lexicon merupakan suatu himpunan yang telah diketahui sentimennya (Desai & Mehta, 2016).

Langkah yang diperlukan ketika menggunakan *Lexicon Based Features* yaitu melakukan pembobotan dengan memuat kamus yang mengandung kata yang bersentimen. Kamus tersebut memiliki beberapa jenis dengan kata kunci negatif dan kata kunci positif (Buntoro, et al., 2014). Proses pembobotan yang dilakukan yaitu menghitung jumlah kata berdasarkan masing-masing fitur yang sebelumnya telah dipilah terlebih dahulu. Kata-kata yang dibobotkan akan dicocokkan dengan kamus sentimen, sehingga setiap fiturnya yang berbobot dapat digunakan pada proses selanjutnya.

2.6.1 Normalisasi *Min-Max*

Berdasarkan penelitian yang dilakukan oleh Rofiqoh dilakukan pembobotan lexicon menggunakan metode normalisasi min-max dengan nilai maksimum 0,9 dan nilai minimum 0,1 (Rofiqoh, 2017). Hal ini diperuntukan untuk menormalisasi

data sehingga data tersebut berada pada *range* tertentu (Junaedi, et al., 2011). Tujuan dalam menormalisasi data ialah untuk meminimalisir kesalahan pada proses *data mining* (Wirawan & Eksistyanto, 2015). Rumus matematika yang digunakan dalam metode ini adalah:

$$v'_i = \frac{v_i - \min_a}{\max_a - \min_a} (\mathit{newmax} - \mathit{newmin}) + \mathit{newmin} \quad (2.4)$$

Persamaan diatas merupakan rumus dalam menghitung nilai data yang telah dinormalisasi. Menurut persamaan diatas, diberikan keterangan rumus:

- v'_i = hasil normalisasi data ke- i
 v_i = data ke- i yang dinormalisasi
 \min_a = data minimum pada kumpulan data a
 \max_a = data maksimum pada kumpulan data a
 newmax = nilai normalisasi maksimum
 newmin = nilai normalisasi minimum

2.6.2 Skor Sentimen

Pembobotan *Lexicon Based* lainnya dilakukan oleh Peng dengan mempertimbangkan skor sentimen dari setiap komentar. Pada pembobotan ini diperlukan suatu kamus lexicon pula. Tahapan yang perlu dilalui pada pembobotan *Lexicon Based* menggunakan perhitungan skor sentimen adalah sebagai berikut (Peng, 2011):

1. Memuat kamus, termasuk kata-kata dan *POS tags*.
2. Parse komentar ke dalam *POS tags*. Hanya kata yang memiliki tag yang benar yang akan dihitung.
3. Hitung skor sentimen pada setiap komentar. Skor sentimen dihitung dengan mencari jumlah kata bersentimen positif dan kata bersentimen negatif. Skor sentimen didapatkan dari jumlah polaritas sentimen positif dikurangi dengan jumlah polaritas sentimen negatif.
4. Kalimat yang merupakan kalimat negatif diperhitungkan dengan menambahkan tanda minus ke skor sentimen.

2.7 Klasifikasi

Klasifikasi merupakan suatu proses pengelompokan data ataupun kumpulan fakta yang telah memenuhi suatu kriteria tertentu. Pendapat lain dari klasifikasi adalah model pada bidang ilmu *data mining* dimana *classifier* dikonstruksi untuk melakukan prediksi kategori atau kelas dari suatu data (Han & Kamber, 2006). Klasifikasi termasuk dalam *supervised learning* karena dalam pengelompokannya telah disediakan label kategori atau *value* untuk masing-masing pola dalam data latih. Dapat disimpulkan bahwa proses klasifikasi akan menghasilkan suatu kelompok yang telah diketahui kategori atau label kelasnya.

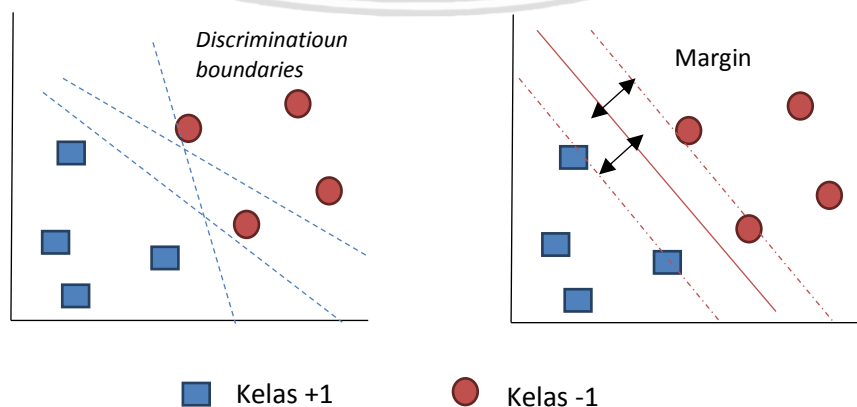
Klasifikasi dibedakan menjadi dua kelompok, yaitu klasifikasi sederhana dan klasifikasi kompleks. Klasifikasi sederhana merupakan klasifikasi yang mengelompokkan objek menjadi dua kategori atau kelas. Namun pada klasifikasi kompleks akan mengelompokkan objek menjadi tiga atau lebih kategori. Proses klasifikasi merupakan proses pengelompokan objek yang disesuaikan dengan kelas yang ada.

Pengklasifikasian suatu data perlu melalui 2 proses terlebih dahulu. Proses awal yang perlu dilakukan adalah pelatihan atau *training* yang dilakukan untuk menganalisis data latih untuk menjadi model prediksi. Setelah proses pelatihan terpenuhi, baru dijalankan proses klasifikasi. Proses klasifikasi dilakukan untuk mengestimasi akurasi data yang didapat dari hasil model prediksi yang diuji dengan data test atau uji. Jika akurasi yang didapat sesuai, maka model tersebut dapat digunakan untuk prediksi kelas atau kategori data yang belum diketahui.

2.7.1 Support Vector Machine

Support Vector Machine atau SVM merupakan salah satu teknik untuk memprediksi yang baik dalam pengklasifikasian dan regresi (Santosa, 2007). Algoritme SVM termasuk dalam algoritme *supervised learning* yang dapat digunakan untuk mengklasifikasikan teks secara otomatis. Penggunaan algoritme SVM yang bertujuan untuk klasifikasi teks dengan menggunakan bobot index term sebagai fitur, dirintis oleh Thorsten Joachim. Pembelajaran SVM telah dipopulerkan sejak tahun 1992 oleh Boser, Guyon, dan Vapnik (Paramita, 2008).

SVM merupakan metode yang dapat menyelesaikan permasalahan secara linier maupun permasalahan non-linier. Dalam menyelesaikan permasalahan non-linier digunakan konsep kernel pada ruang kerja berdimensi tinggi, dimana akan mencari *hyperplane* yang dapat memaksimalkan margin antar kelas data. *Hyperplane* berguna dalam memisahkan 2 kelompok *class +1* dan *class -1* dimana setiap *class* memiliki *pattern* masing-masing. *Hyperplane* terbaik ditemukan dengan mengukur margin *hyperplane* dan mencari titik maksimal. Margin merupakan jarak *hyperplane* dengan *pattern* terdekat dari masing-masing *class*. Himpunan data yang terletak pada margin disebut dengan *support vector*.



Gambar 2.1 Pemisah *Hyperplane* Terbaik

Sumber: (Hasanah, 2016)

Data yang ada dinotasikan sebagai $\vec{x}_i \in \mathbb{R}^d$ dengan label masing-masing dinotasikan $y_i \in \{-1, +1\}$ untuk $i = 1, 2, \dots, l$, dimana l adalah banyak data. Dapat diasumsikan bahwa *class* +1 dan *class* -1 dapat terpisah secara sempurna dengan *hyperplane* dengan dimensi d , sehingga didefinisikan:

$$\vec{x} \cdot \vec{w} + b = 0 \tag{2.5}$$

Pada pattern \vec{x}_i yang termasuk pada *class* +1 dirumuskan sebagai pertidaksamaan,

$$\vec{x}_i \cdot \vec{w} + b \leq -1 \tag{2.6}$$

Sedangkan pattern \vec{x}_i yang termasuk pada *class* -1 dirumuskan sebagai pertidaksamaan,

$$\vec{x}_i \cdot \vec{w} + b \geq +1 \tag{2.7}$$

Untuk menemukan nilai margin terbesar didapatkan dengan memaksimalkan nilai jarak antar *hyperplane* dan titik terdekatnya. Hal ini didapatkan dengan rumus:

$$\frac{1}{\|\vec{w}\|} \tag{2.8}$$

Dari persamaan diatas diketahui bahwa teknik margin yang dijelaskan adalah Teknik *hard margin*. Namun dalam mengimplementasikan *hard margin* memiliki kendala dikarenakan hasil *hyperplane* yang sempurna, sehingga sistem tidak memiliki data pembelajaran. Selain itu kendala lain ditimbulkan karena jenis data pembelajaran yang masih *non-linear separable*. Dalam mengatasi masalah-masalah tersebut maka digunakan teknik *soft margin* yang memperkenalkan variabel *slack* dan merupakan galat dari data setiap data pembelajaran. Untuk penggunaan variabel *slack* ditunjukkan pada *sequential learning*.

Dalam mengambil keputusan dengan metode SVM digunakan fungsi kernel $K(x_i, x_d)$. Pada umumnya terdapat beberapa persamaan kernel yang dapat digunakan, yaitu dengan linear, polynomial, *radial basis function (RBF)*, dan sigmoid. Persamaan kernel ditunjukkan sebagai rumus berikut:

Linear,

$$K(x_i, x_d) = X_i^T X_j \tag{2.9}$$

Polynomial,

$$K(x_i, x_d) = (X_i^T X_j + 1)^d, \gamma > 0 \tag{2.10}$$

Radial Basis Function (RBF),

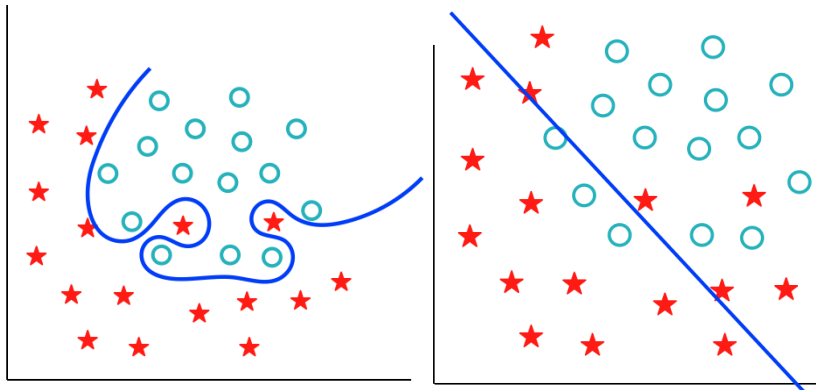
$$K(x_i, x_d) = \exp(-\gamma \|X_i - X_j\|^2), \gamma > 0 \tag{2.11}$$

Sigmoid,

$$K(x_i, x_d) = \tan \gamma X_i^T X_j + r \tag{2.12}$$

Pada sistem klasifikasi penelitian ini akan menerapkan persamaan kernel polynomial dikarenakan kernel polynomial merupakan kernel yang sederhana

untuk diimplementasikan. Pada kernel polynomial terdapat parameter *degree* yang dapat diubah untuk mencari nilai optimal dalam proses klasifikasi.



Gambar 2.2 Overfitting dan Underfitting

Namun dalam proses klasifikasi data dapat terjadi permasalahan ketika model data yang terlalu akurat, yaitu mempelajari data secara detail termasuk *noise* data. Hal ini menyebabkan data latih memiliki hasil yang baik namun hasil klasifikasi data yang buruk. Hal ini biasa disebut dengan *overfitting*. Selain itu terdapat masalah lainnya yang disebut dengan *underfitting*, dimana terdapat model yang tidak dapat memodelkan data latihnya (Browniee, 2016).

2.7.1.1 Sequential Training

Terdapat tiga jenis algoritme untuk memproses data latih dari *Support Vector Machine* yang dapat digunakan, yaitu *Quadratic Programming*, *Sequential Minimal Optimization*, dan *Sequential Training*. Dalam penggunaan algoritme tersebut perlu untuk memperhatikan setiap kekurangan maupun kelebihan dari algoritme. *Quadratic Programming* merupakan proses perumusan yang memberikan hasil berupa analisa numerik yang memakan waktu cukup lama dengan algoritme yang kompleks. *Sequential Minimal Optimization* adalah pengembangan dari *Quadratic Programming* dengan algoritme kompleks yang hanya mampu memberikan optimasi kecil. *Sequential Training* adalah algoritme yang sederhana tanpa memakan waktu yang banyak, dengan proses sebagai berikut (Vijayakumar, 1999):

1. Inisialisasi terhadap berbagai parameter, seperti α_i , γ , C , dan ϵ .

Keterangan:

- α_i = alfa, untuk mencari support vector
- γ = konstanta gamma untuk mengontrol kecepatan
- C = variabel *slack*
- ϵ = epsilon digunakan untuk mencari nilai error

2. Hitung matriks *Hessian* yang didapat dari perkalian antar kernel polynomial dan y yang merupakan vector bernilai 1 dan -1. Persamaan dari matriks Hessian adalah:

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.13)$$

Dengan nilai i dan $j = 1, 2, 3, \dots, n$

Keterangan:

x_i = data ke- i

x_j = data ke- j

y_i = kelas data ke- i

y_j = kelas data ke- j

$K(x_i, x_j)$ = fungsi kernel

3. Lakukan perhitungan berikut hingga iterasi data i hingga j :

a. $E_i = \sum_{j=1}^i a_j D_{ij}$ (2.14)

Keterangan:

a_j = alfa ke- j

D_{ij} = matriks Hessian

E_i = error rate

b. $\delta\alpha_i = \min(\max[\gamma(1 - E_i), \alpha_i], C - \alpha_i)$ (2.15)

Keterangan:

α_i = alfa ke- i

γ = konstanta gamma

E_i = error rate

C = variabel slack

c. $\alpha_i = \alpha_i + \delta\alpha_i$ (2.16)

Keterangan:

α_i = alfa ke- i

$\delta\alpha_i$ = delta alfa ke- i

d. Lakukan ketiga langkah diatas secara berulang hingga mencapai batas maksimum iterasi.

e. Proses diatas akan didapatkan nilai dari support vector (SV), dimana nilai $SV=(\alpha_i > threshold_{SV})$. Setelah itu, perlu dilakukan perhitungan pada nilai bias b yang diperoleh dari Persamaan 2.17.

$$b = -\frac{1}{2} (\sum_{i=1}^N \alpha_i y_i K(x_i, x^-) + \sum_{i=1}^N \alpha_i y_i K(x_i, x^+)) \quad (2.17)$$

f. Untuk mengetahui hasil klasifikasi teks pada kelas sentimen tertentu maka dilakukan proses perhitungan fungsi $f(x)$. Jika hasil dari fungsi tersebut bernilai negatif, maka dokumen terklasifikasi pada sentimen kelas negatif *cyberbullying*. Dan jika nilai fungsi bernilai positif, maka dokumen terklasifikasi pada kelas sentimen positif *cyberbullying*. Fungsi $f(x)$ diperoleh pada Persamaan 2.18.



$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x) + b \tag{2.18}$$

2.8 Evaluasi

Evaluasi merupakan tahapan dalam upaya untuk mengukur keberhasilan suatu sistem dengan membandingkan hasil perolehan implementasi dengan kriteria standar yang telah ditetapkan (Parikh & M.M, 2009). Umumnya untuk mengevaluasi hasil implementasi pada sentimen analisis menggunakan *confusion matrix*. Pengukuran evaluasi dilakukan berdasarkan *confusion matrix* yang diperlihatkan pada Tabel 2.4.

Tabel 2.4 Confusion Matrix

Classification	Predicted Positives	Predicted Negatives
<i>Actual Positive Cases</i>	<i>Number of True Positive Cases (TP)</i>	<i>Number of False Negative Cases (FN)</i>
<i>Actual Negatives Cases</i>	<i>Number of False Positive Cases (FP)</i>	<i>Number of True Negative Cases (TN)</i>

Dari Tabel 2.14 diketahui bahwa *true positive* merupakan jumlah dokumen yang prediksi kelasnya bernilai positif dan kelas aktualnya bernilai positif. *False negative* adalah jumlah dokumen yang diprediksi menjadi kelas negatif oleh sistem, namun kelas aktual dari dokumen adalah positif. *False positive* adalah jumlah dokumen yang diprediksi sebagai kelas positif oleh sistem tetapi kelas aktualnya adalah negatif. Sedangkan *true negative* ialah jumlah dokumen kelas yang diberikan oleh sistem dan kelas aktualnya bernilai sama, yaitu negatif.

Perhitungan yang dilakukan dalam tahap evaluasi berupa *Accuracy*, *Precision*, *Recall*, dan *F-Measure* didefinisikan pada persamaan berikut:

$$Accuracy = \frac{TN+TP}{TN+TP+FP+FN} \tag{2.19}$$

$$Precision = \frac{TP}{TP+FP} \tag{2.20}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.21}$$

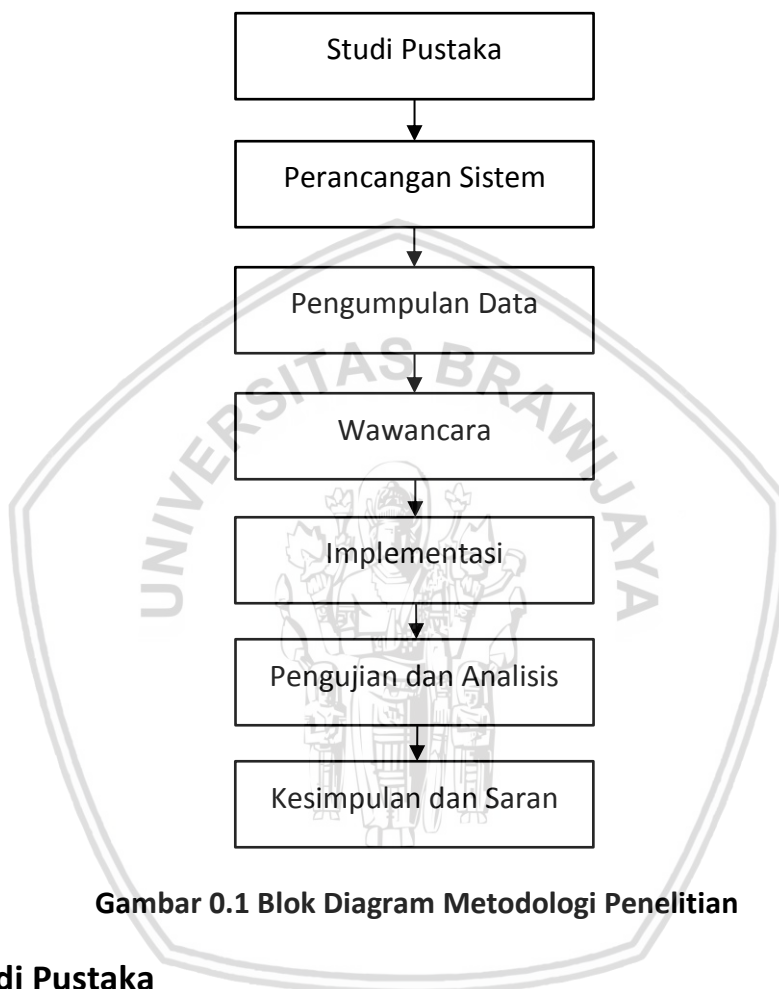
$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.22}$$

Accuracy merupakan evaluasi yang dilakukan dengan menghitung seluruh keadaan yang diprediksikan dengan nilai yang benar terhadap seluruh keadaan yang diprediksi. Proses evaluasi *precision* merupakan perhitungan pada kondisi benar, yaitu kelas aktual dan kelas prediksi yang sama (positif) terhadap seluruh kondisi yang diprediksi positif. *Recall* adalah perhitungan pada kondisi benar yaitu, merupakan kelas data positif terhadap seluruh kondisi aktual yang bernilai positif. Sedangkan *F-Measure* adalah perhitungan yang melibatkan *precision* dan *recall* untuk dicari nilai tengah pada kedua evaluasi tersebut.



BAB 3 METODOLOGI

Pada bab metodologi penelitian dilakukan beberapa tahapan untuk menyelesaikan penelitian “Analisis Sentimen *Cyberbullying* Pada Komentar Instagram Dengan Metode Klasifikasi *Support Vector Machine*” yang diilustrasikan pada blok diagram seperti pada Gambar 3.1:



Gambar 0.1 Blok Diagram Metodologi Penelitian

3.1 Studi Pustaka

Pada studi pustaka membahas mengenai teori pendukung yang digunakan sebagai acuan dalam menyelesaikan masalah. Penelitian ini memerlukan untuk mempelajari bidang ilmu berkaitan yaitu, analisis sentimen pada Instagram yang menggunakan metode *Support Vector Machine*. Teori pendukung yang dibahas pada penelitian ini adalah:

- Analisis Sentimen.
- *Text mining*.
- Media sosial.
- *Cyberbullying*.
- Klasifikasi.

- Algoritme *Support Vector Machine*.

Literatur tersebut didapatkan dari buku, jurnal, artikel, dan dokumen *project*.

3.2 Perancangan Sistem

Untuk mendapatkan hasil klasifikasi yang sesuai, maka perlu melalui beberapa tahapan terlebih dahulu. Tahap awal dalam proses klasifikasi adalah menghitung bobot kata yang terkumpul dengan menggunakan metode TF-IDF. Setelah bobot kata diketahui maka akan dilakukan proses *training* dengan menggunakan fungsi kernel dan algoritme *Support Vector Machine* untuk mengetahui hasil klasifikasi sehingga dapat diketahui analisis sentimen yang ingin diketahui.

3.3 Pengumpulan Data

Untuk menyelesaikan permasalahan diperlukan data yang akan diproses terlebih dahulu. Data yang digunakan pada penelitian ini diambil dari kumpulan komentar yang ada pada beberapa akun aktif di aplikasi Instagram atau www.instagram.com. Dataset yang digunakan berupa teks digital yang diambil secara random.

3.4 Wawancara

Tahap wawancara diperlukan dalam penelitian ini. Bahasan yang dilakukan pada tahapan ini adalah untuk mengetahui lebih dalam mengenai *cyberbullying* sehingga perlu dilakukan wawancara dengan pakar. Urgensi mengenai tindakan *cyberbullying* dan bagaimana bentuk *cyberbullying* perlu diketahui lebih lanjut, sehingga data yang digunakan dapat menunjang penelitian.

3.5 Implementasi

Sistem diaplikasikan dengan membuat program dengan bahasa pemrograman *python* untuk memberikan solusi pada analisis sentimen pada komentar Instagram dengan diimplementasikan algoritme *Support Vector Machine* yang mengacu pada perancangan sistem yang telah diuraikan. Implementasi juga akan dilakukan dengan perhitungan manual dengan algoritme *Support Vector Machine* untuk klasifikasi dalam upaya mengetahui hasil analisis sentimen.

Secara rinci dalam mengimplementasikan sistem dibutuhkan perangkat keras, perangkat lunak, dan data perlu dipenuhi untuk mengimplementasi sistem. Adapun kebutuhah-kebutuhan tersebut adalah:

- Kebutuhan perangkat keras:
 - Komputer Core i5
 - RAM 8GB
 - ROM 500GB
- Kebutuhan perangkat lunak:

- Seluruh sistem operasi Windows 8.1.
 - Python 2.6.7
 - PyCharm Community Edition 2017 2.3
 - Microsoft Excel.
- Kebutuhan data:
Data komentar yang ada pada kolom komentar Instagram.

3.6 Pengujian dan Analisis

Pengujian yang dilakukan bertujuan dalam memberikan informasi akurat mengenai tingkat akurasi algoritme yang dilakukan oleh sistem. Algoritme tersebut akan memberikan suatu penilaian terhadap kualitas klasifikasi pada aplikasi yang dijalankan. Pengujian dilakukan dengan menggunakan *confusion matrix* dan kemudian akan menghitung tingkat akurasi, *precision*, *recall*, dan *f-measure*.

Untuk menganalisis sistem akan dilakukan perbandingan dari hasil pengujian yang telah dilakukan. Hasil analisis sistem akan memberikan informasi mengenai analisis sentimen komentar pada Instagram, sebagai bentuk solusi dalam mendeteksi *cyberbullying* yang terjadi. Selain itu analisis dilakukan untuk menentukan nilai akurasi yang telah diberikan oleh perhitungan sistem dengan perhitungan secara manual.

3.7 Kesimpulan dan Saran

Untuk mendapatkan suatu kesimpulan maka perlu diselesaikan seluruh tahapan perancangan dan analisis metode sehingga dapat menjawab rumusan masalah yang telah diuraikan. Penelitian yang tidak luput dari kekurangan akan membutuhkan saran sehingga segala kesalahan dan kekurangan dalam melakukan penelitian maupun hasil penelitian yang diperoleh dapat diperbaiki dan dipertimbangkan untuk dilakukan penelitian kembali dengan tujuan mengembangkan penelitian.



BAB 4 ANALISIS DAN PERANCANGAN SISTEM

Pada bab empat akan diuraikan mengenai permasalahan yang diangkat, deskripsi umum sistem yang akan diterapkan, algoritma yang akan diimplementasikan yang bertujuan untuk menyelesaikan permasalahan, dan perancangan pengujian.

4.1 Deskripsi Permasalahan

Perkembangan zaman yang diiringi dengan perkembangan teknologi menjadi suatu sorotan utama pada masa kini. Komunikasi yang dilakukan dalam kehidupan sehari-hari yang tidak lepas dari penggunaan teknologi memicu bertambahnya berbagai jenis dari media sosial. Mulanya sosial media terpopuler dari kalangan anak-anak hingga dewasa adalah facebook. Namun kini penggunaan facebook telah tergeser oleh penggunaan Instagram, yang merupakan salah satu media sosial terpopuler dimasa kini.

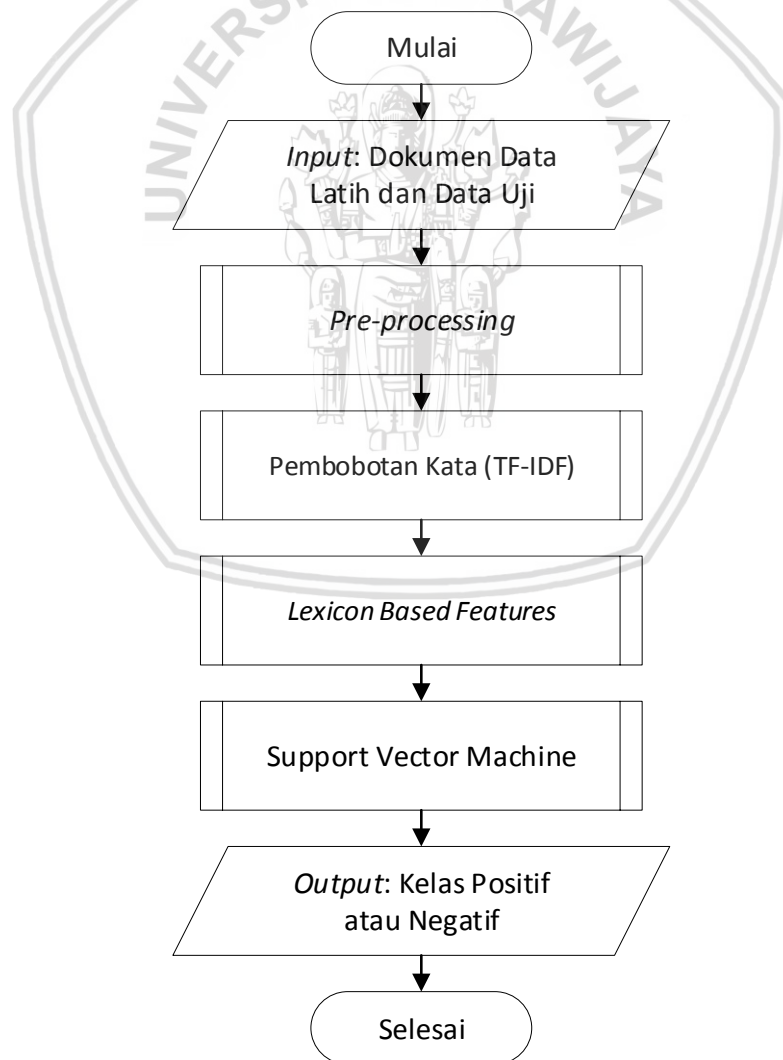
Instagram yang merupakan media sosial berbentuk aplikasi di *smartphone*, memiliki fitur untuk membagi setiap foto maupun video yang berharga pada seluruh pengguna Instagram yang diperkenankan untuk melihat. Fitur lainnya yang dapat dinikmati yaitu untuk mengikuti dan tidak mengikuti pengguna lainnya, layanan untuk memberi pesan secara langsung pada pengguna tertentu, memberi tanda suka pada foto atau video, dan memberikan komentar pada foto atau video yang telah diunggah. Dari beberapa fitur tersebut telah terjadi penyalahgunaan oleh pengguna Instagram sendiri. Seperti pemberian komentar yang tidak layak ditujukan pada orang lain. Hal ini memicu suatu tindakan *cyberbullying* yang dapat berakibat buruk bagi para korban khususnya korban yang masih dibawah umur (anak-anak dan remaja). Tindakan *cyberbullying* yang telah marak di Instagram seiring dengan berbagai unggahan foto, namun sangat disayangkan bahwa banyak pengguna tanpa sadar telah memberikan ujaran kebencian yang merupakan salah satu bentuk dari *cyberbullying*.

Untuk mengetahui bagaimana penggunaan Instagram khususnya pada pemberian komentar pada foto-foto yang telah diunggah, maka diperlukan suatu analisis sentimen pada komentar-komentar yang telah dilontarkan pada kolom komentar Instagram. Analisis sentimen yang akan diterapkan akan mengelompokkan teks dalam kalimat atau dokumen untuk diketahui teks yang digolongkan sebagai positif mengandung *cyberbullying* atau teks yang digolongkan sebagai negatif *cyberbullying*. Untuk mengetahui hasil analisis sentimen dari komentar Instagram, akan diimplementasikan metode TF-IDF dalam pembobotan kata dan metode klasifikasi *Support Vector Machine* yang berfungsi dalam mencari *hyperplane* atau fungsi pemisah antar kelas data.

4.2 Deskripsi Umum Sistem

Analisis sentimen *cyberbullying* pada komentar instagram dengan metode klasifikasi *Support Vector Machine* merupakan suatu sistem yang dikembangkan untuk membantu menganalisis komentar-komentar pada Instagram yang mengandung *cyberbullying* atau tidak. Data yang digunakan pada penelitian ini adalah data dari kolom komentar Instagram yang telah diambil yang terdiri dari data latih dan data uji.

Gambaran umum sistem yang akan diimplementasikan, mulanya data latih dan data uji akan dimasukkan ke dalam *Pre-processing* teks yang bertujuan untuk mengolah data agar dapat dianalisis pada algoritma *Support Vector Machine*. Setelah didapatkan data hasil dari *Pre-processing*, akan dilakukan pembobotan kata dengan menggunakan rangkaian hitungan TF-IDF. Setelah diketahui bobot dari setiap kata, maka hasil perhitungan dari TF-IDF akan masuk pada tahap klasifikasi dengan metode *Support Vector Machine* yang memberikan hasil bahwa data termasuk pada kelas positif *cyberbullying* atau negatif *cyberbullying*. Alur proses dari deskripsi umum sistem akan ditunjukkan pada Gambar 4.1.



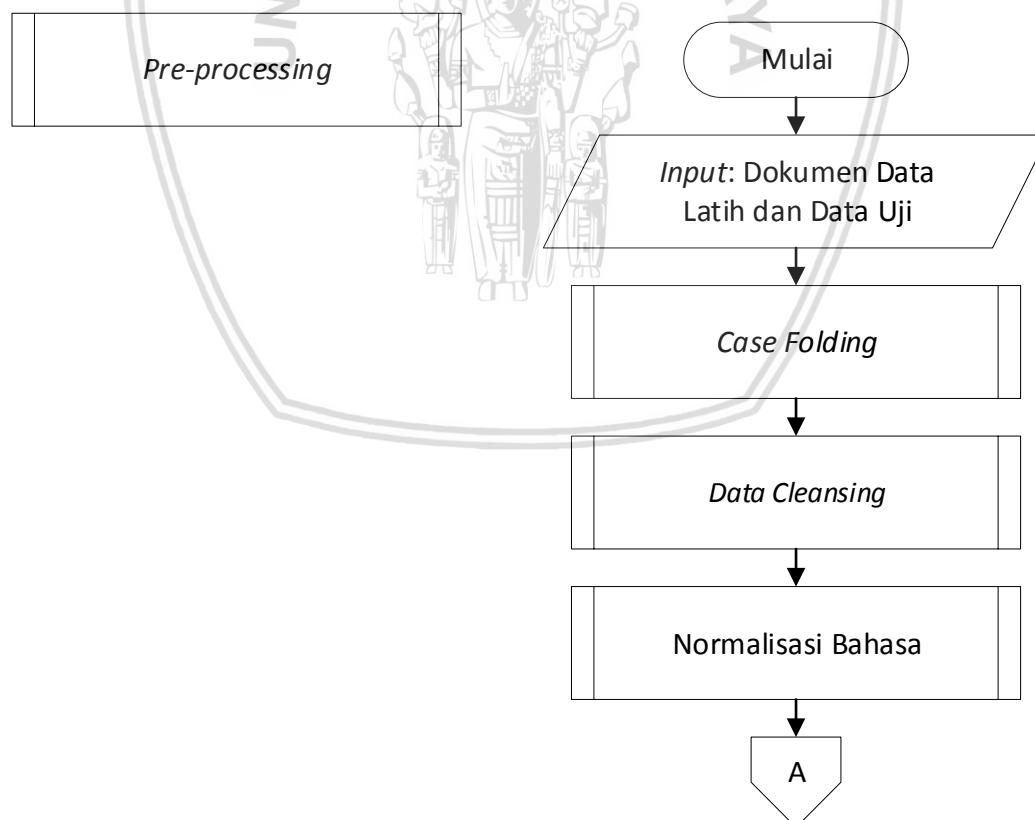
Gambar 0.1 Deskripsi Umum Sistem

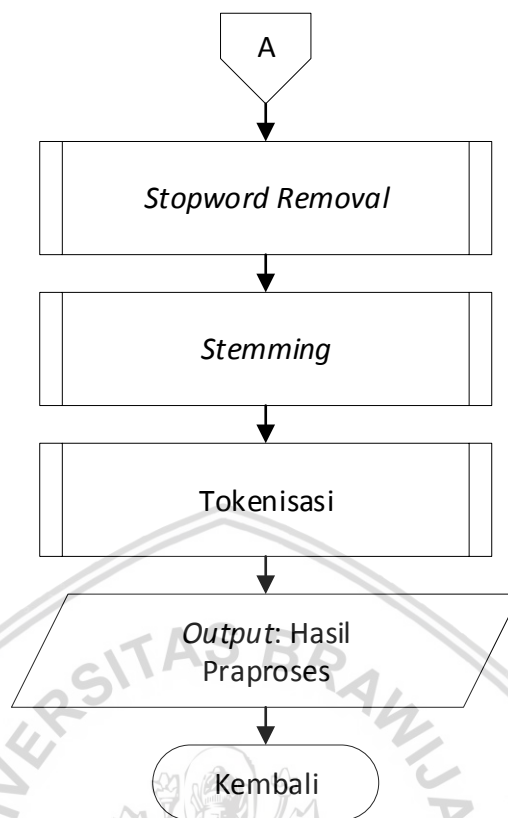
Alur proses sistem yang telah ditunjukkan pada Gambar 4.1 dilakukan dengan tahapan:

1. Memasukkan dataset yang berupa data latih dan data uji.
2. Memasuki tahapan *Pre-processing*.
3. Hasil *Pre-processing* akan memasuki tahapan pembobotan kata (TF-IDF).
4. Pembobotan *Lexicon Based Features* dilakukan setelah tahapan pembobotan kata selesai dilakukan.
5. Tahapan akhir yang dilakukan oleh sistem adalah proses klasifikasi dengan algoritme *Support Vector Machine*.
6. *Output* (keluaran) yang dihasilkan berupa kelas prediksi dari data uji.

4.3 Pre-processing

Pre-processing merupakan tahapan awal yang akan dilalui dalam memproses teks. Pada penelitian ini akan dilakukan tahapan *Pre-processing* dengan tahapan case folding, data cleansing, normalisasi kata tidak baku, *stopword removal*, *stemming*, dan tokenisasi. Alur proses pada tahapan ini akan ditunjukkan pada Gambar 4.2.





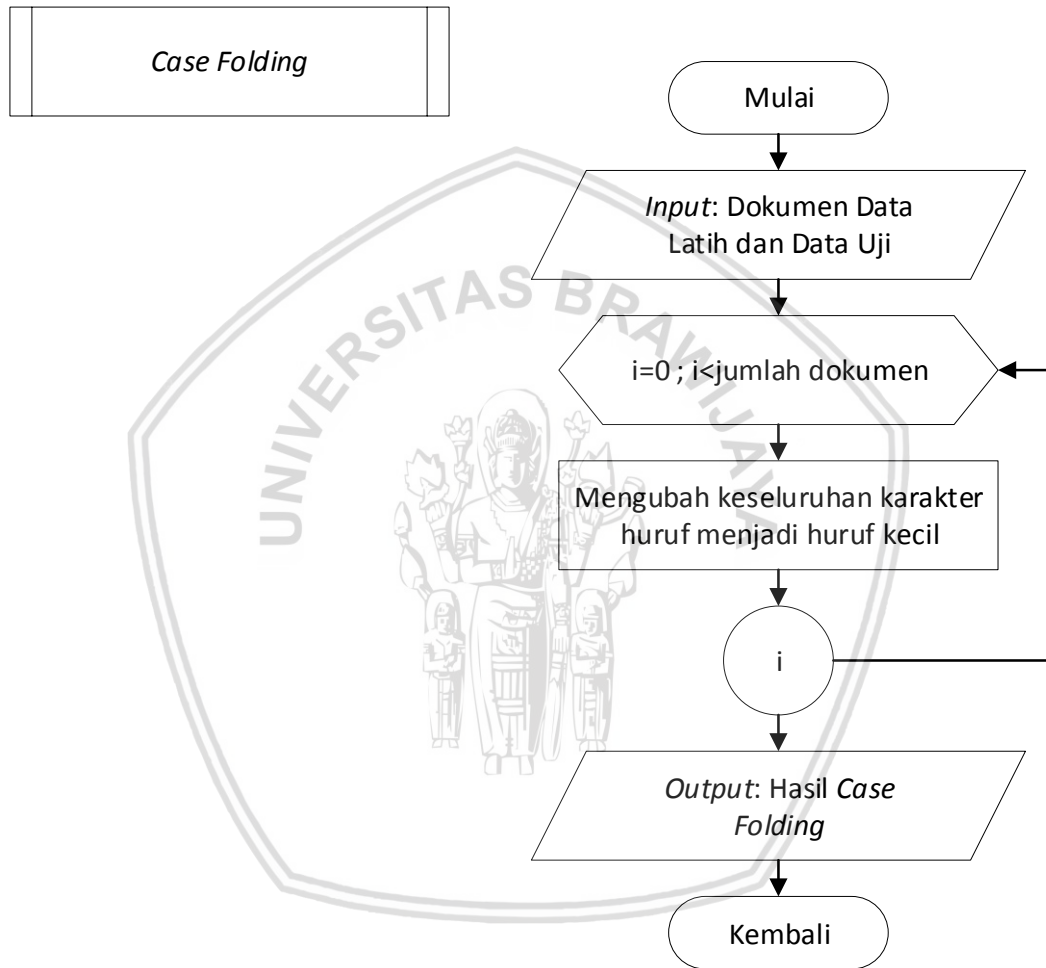
Gambar 0.2 Alur Proses *Pre-processing*

Berdasarkan Gambar 4.2 alur pada tahapan *Pre-processing* akan melalui beberapa sub-proses. Sub-proses yang dilalui oleh sistem, dijelaskan sebagai berikut:

1. Memasukkan dataset yang berupa data latih dan data uji.
2. Menjalankan tahapan sub-proses, *case folding*.
3. Hasil dari *case folding* memasuki tahapan sub-proses, *data cleaning*.
4. Hasil *data cleaning* akan diproses pada tahapan sub-proses normalisasi bahasa.
5. Hasil dari tahapan normalisasi bahasa, melalui tahapan selanjutnya yaitu *stopword removal*.
6. Hasil dari tahapan *stopword removal* diproses kembali pada tahapan *stemming*.
7. Tahapan sub-proses terakhir pada *Pre-processing* adalah tokenisasi pada hasil *stemming*.
8. *Output* yang diberikan berupa hasil dari *Pre-processing*.

4.3.2 Case Folding

Tahapan awal dari *Pre-processing* adalah *case folding*. Pada tahapan ini seluruh teks baik pada data latih maupun data uji akan dimasukkan. Setelah seluruh teks dimasukkan, dilakukan proses untuk mengubah seluruh teks atau karakter menjadi huruf kecil atau *lowercase*. Proses yang telah berhasil, menghasilkan data *output* dari *case folding* yang kemudian akan dilanjutkan pada tahapan *Pre-processing* selanjutnya. Alur proses kerja pada *case folding* ditunjukkan pada Gambar 4.3.



Gambar 0.3 Alur Proses Case Folding

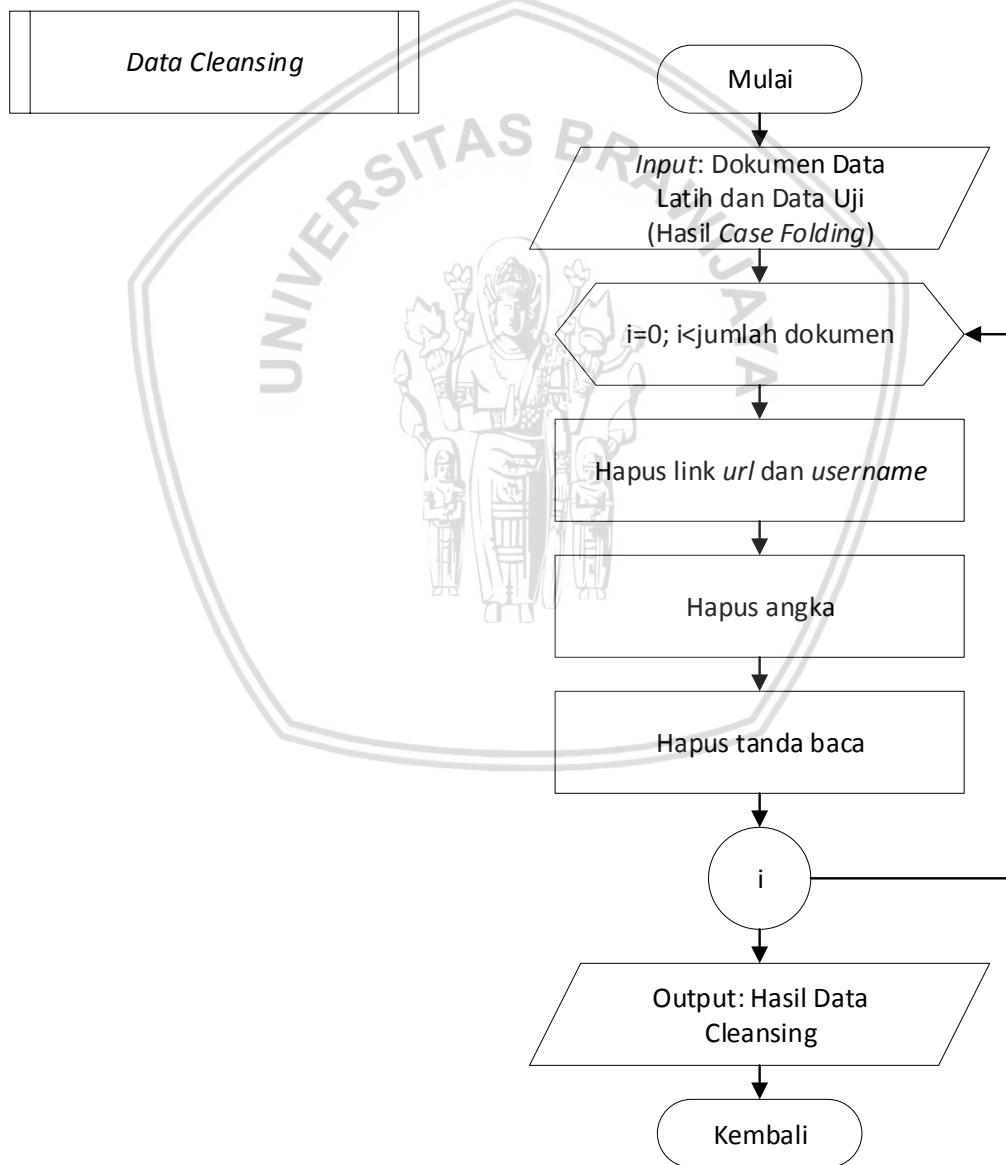
Tahapan pada proses *case folding* yang telah tertera pada Gambar 4.3 dijelaskan sebagai berikut:

1. Memasukkan dataset yang berupa data latih dan data uji,
2. Perulangan terhadap jumlah dokumen dataset yang dimasukkan.
3. Proses mengubah keseluruhan karakter pada dokumen menjadi huruf kecil.
4. *Output* berupa hasil proses *case folding*.



4.3.3 Data cleaning

Dalam melakukan *Pre-processing* teks, perlu dilalui tahapan *data cleaning*. Tahapan ini akan menjalankan fungsinya untuk membersihkan data dari karakter-karakter tertentu. Proses jalannya *data cleaning* dimulai dari menginput data dari hasil *case folding*. Data yang telah dimasukkan dibaca dalam bentuk matrix dimana akan dilakukan perulangan pada jumlah dokumen (komentar) dan *perulangan* pada kata (term) pada setiap dokumen (komentar). Mulanya dijalankan *perulangan* pada baris dan dilakukan penghapusan pada kata atau karakter yang berupa *link url*, *username* (ditandai dengan karakter '@'), tanda baca, dan angka. Setelah setiap dokumen yang mengandung kata (term) telah bersih dari link url, *username*, tanda baca, dan angka, maka telah didapat *output* dari *data cleaning*. Alur proses yang jelas pada data cleansing ditunjukkan pada Gambar 4.4.



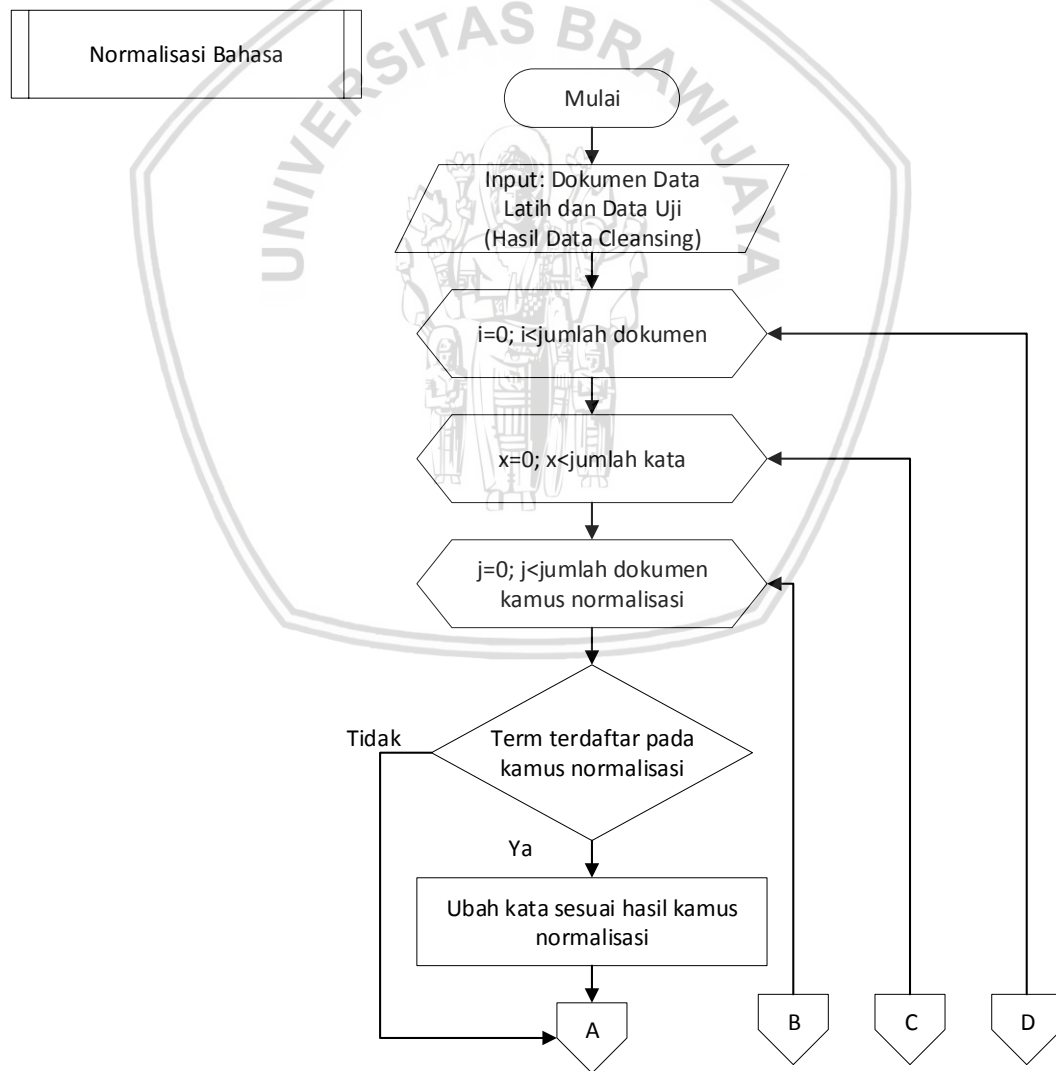
Gambar 0.4 Alur Proses *Data cleaning*

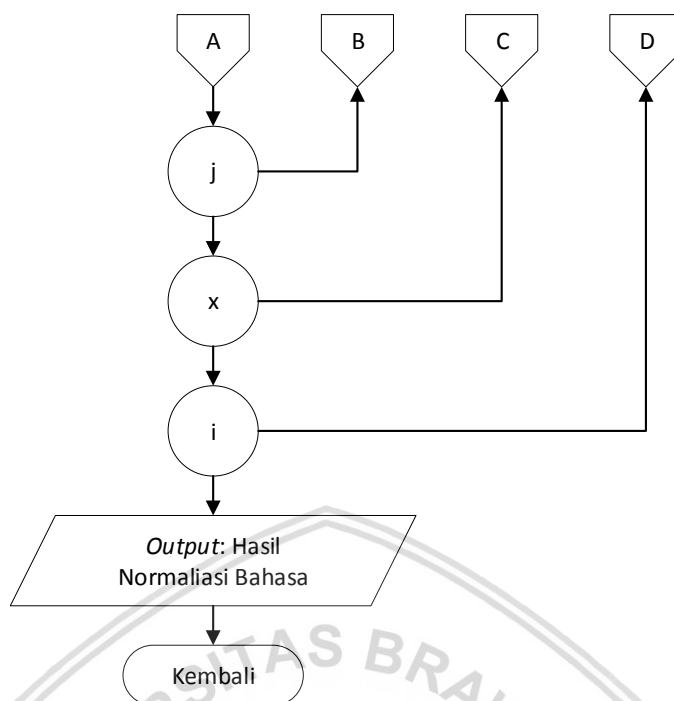
Tahapan proses pada *data cleaning* yang telah tertera pada Gambar 4.4 dijelaskan sebagai berikut:

1. Masukkan dataset dari hasil *case folding*.
2. Perulangan terhadap jumlah dokumen dataset yang dimasukkan dengan proses untuk menghapus link *url*, username, angka, dan tanda baca.
3. *Output* pada proses ini berupa hasil *data cleaning*.

4.3.4 Normalisasi Bahasa

Normalisasi bahasa dilakukan pada tahapan *Pre-processing* untuk mengubah kata yang tidak baku menjadi kata baku sesuai dengan yang ada pada KBBI. Proses normalisasi memerlukan kamus yang memuat daftar kata tidak baku untuk diperiksa pada data latih maupun data uji. Alur proses normalisasi bahasa ditunjukkan pada Gambar 4.5.





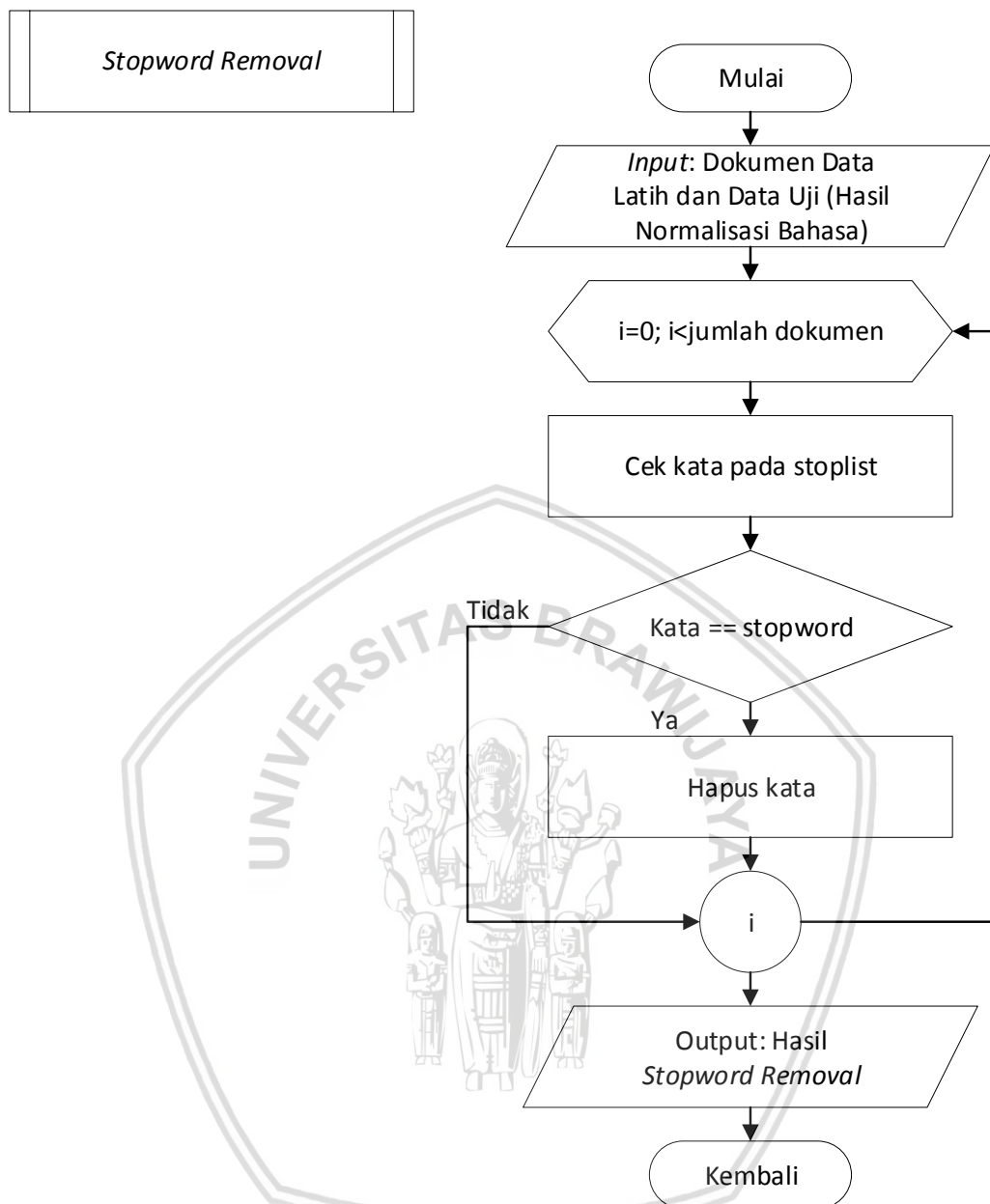
Gambar 0.5 Alur Proses Normalisasi Bahasa

Tahapan proses normalisasi bahasa yang ditunjukkan pada Gambar 4.5 dijelaskan sebagai berikut:

1. Masukkan dataset adalah dari hasil *data cleaning*.
2. Perulangan bersarang (*nested loop*) pada jumlah dokumen dataset, jumlah kata (panjang kata pada satu baris dokumen), dan jumlah dokumen kamus normalisasi (panjang dokumen kamus normalisasi). Pada perulangan tersebut terdapat kondisi if-else, jika terdapat kata yang sama pada kamus normalisasi, maka kata tersebut diubah sesuai hasil normalisasi yang ada pada kamus.
3. *Output* yang diberikan adalah hasil normalisasi bahasa.

4.3.5 Stopword Removal

Tahapan filterisasi dilakukan pada *Pre-processing* data. Filterisasi dilakukan dengan *stopword removal*, yaitu menghapus kata (term) yang tidak memiliki makna penting dalam dokumen. Proses pada *stopword removal* ini digambarkan pada Gambar 4.6. Awal proses *stopword removal* dilakukan dengan menginputkan dokumen hasil tokenisasi yang kemudian masuk pada tahap *perulangan* untuk mengecek setiap kata pada dokumen, apakah termasuk dalam daftar *stoplist*. Jika kata telah terdefiniskan pada daftar *stoplist*, maka kata tersebut akan dihapus.



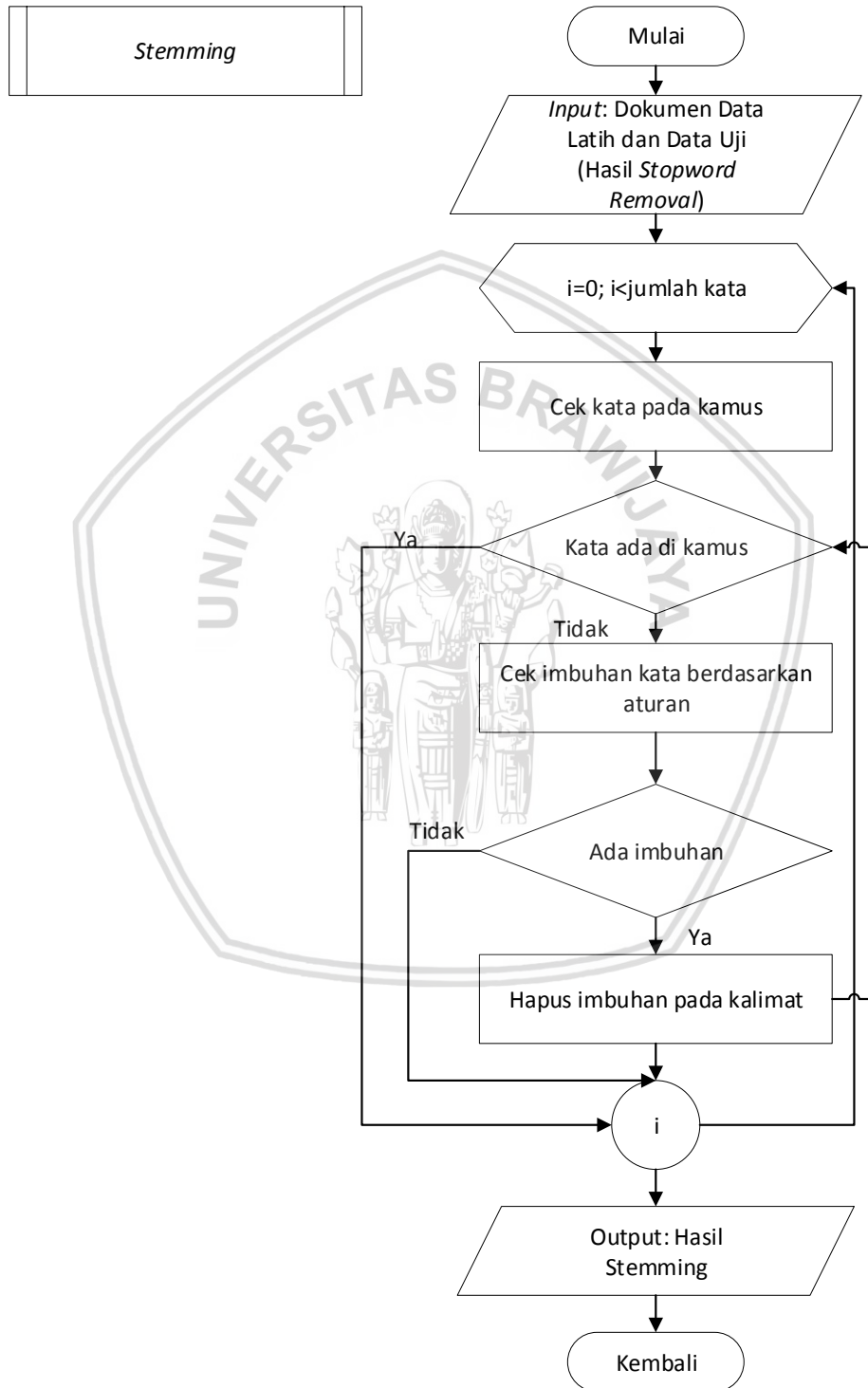
Gambar 0.6 Alur Proses *Stopword Removal*

Tahapan proses pada *stopword removal* yang telah ditunjukkan pada Gambar 4.6 dijelaskan pada keterangan berikut:

1. Masukkan dataset adalah hasil normalisasi bahasa.
2. Perulangan yang dilakukan pada jumlah kata (banyak kata pada satu baris dokumen) dengan proses yang dijalankan adalah mengecek setiap kata pada dokumen dan dibandingkan dengan daftar *stoplist*. Jika terdapat kata yang sama pada dokumen masukkan dan daftar *stoplist*, maka kata tersebut dihapuskan.
3. Output pada tahapan ini adalah hasil *stopword removal*.

4.3.6 Stemming

Tahapan dari subproses dari *Pre-processing* teks adalah *stemming*. Proses yang dilakukan untuk mencari kata dasar pada data latih maupun data uji dilakukan dengan menghapus setiap kata imbuhan yang ada. Alur proses pada *stemming* ditunjukkan pada Gambar 4.7.



Gambar 0.7 Alur Proses *Stemming*

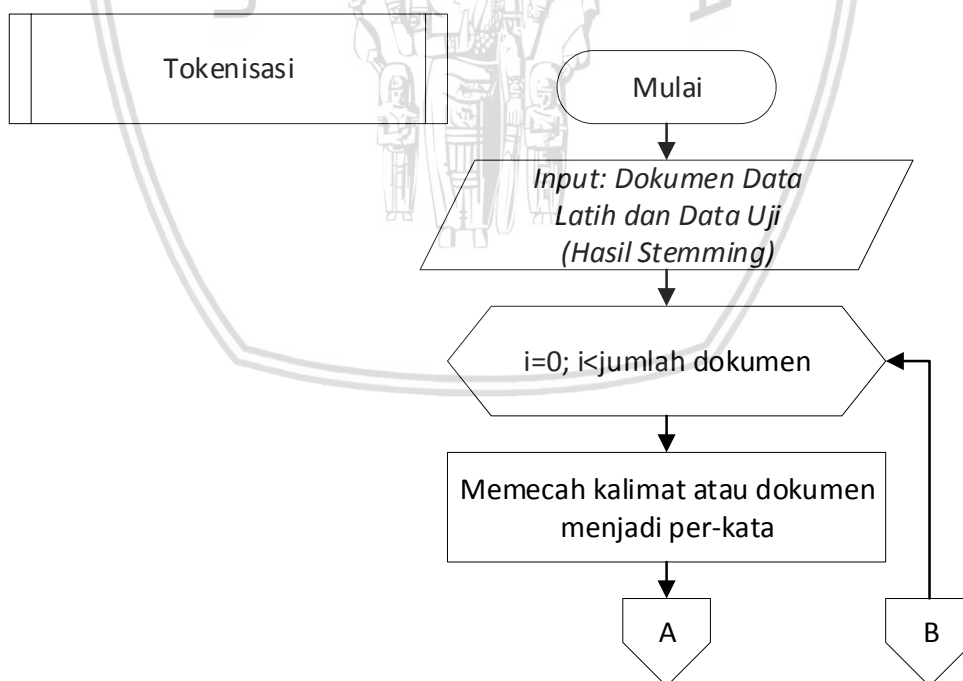


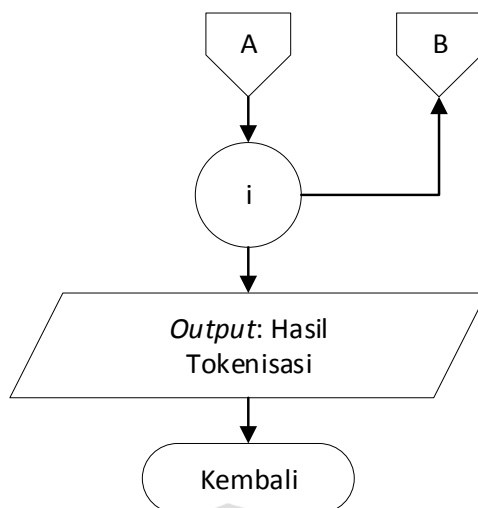
Tahapan proses *stemming* yang ditunjukkan pada Gambar 4.7 dijelaskan pada keterangan berikut:

1. Masukkan dataset adalah hasil dari *stopword removal*.
2. Perulangan pada jumlah kata (panjang kata pada satu baris dokumen) dan melakukan proses pengecekan kata apakah telah terdapat pada kamus.
3. Kata yang terdapat pada kamus akan menghentikan prosesnya. Namun kata yang tidak terdaftar pada kamus akan melakukan proses pengecekan imbuhan berdasarkan aturan.
4. Jika tidak terdapat imbuhan maka proses berhenti. Namun jika terdapat imbuhan, dilakukan proses penghapusan imbuhan.
5. *Output* yang diberikan pada proses ini adalah hasil *stemming*.

4.3.7 Tokenisasi

Tahapan tokenisasi merupakan tahapan akhir yang dilakukan oleh sistem yang berfungsi dalam memecah dokumen, paragraf, kalimat menjadi kata-kata tunggal seperti yang ditunjukkan Gambar 4.8. Proses ini didahului dengan memasukkan dokumen dari proses *stemming*. Proses *perulangan* dilakukan pada data yang telah dimasukkan. Dokumen yang telah dibaca oleh sistem akan dipecah kalimat atau dokumen menjadi perkata. *Output* yang diberikan adalah hasil tokenisasi yang disimpan selama *perulangan* dijalankan.





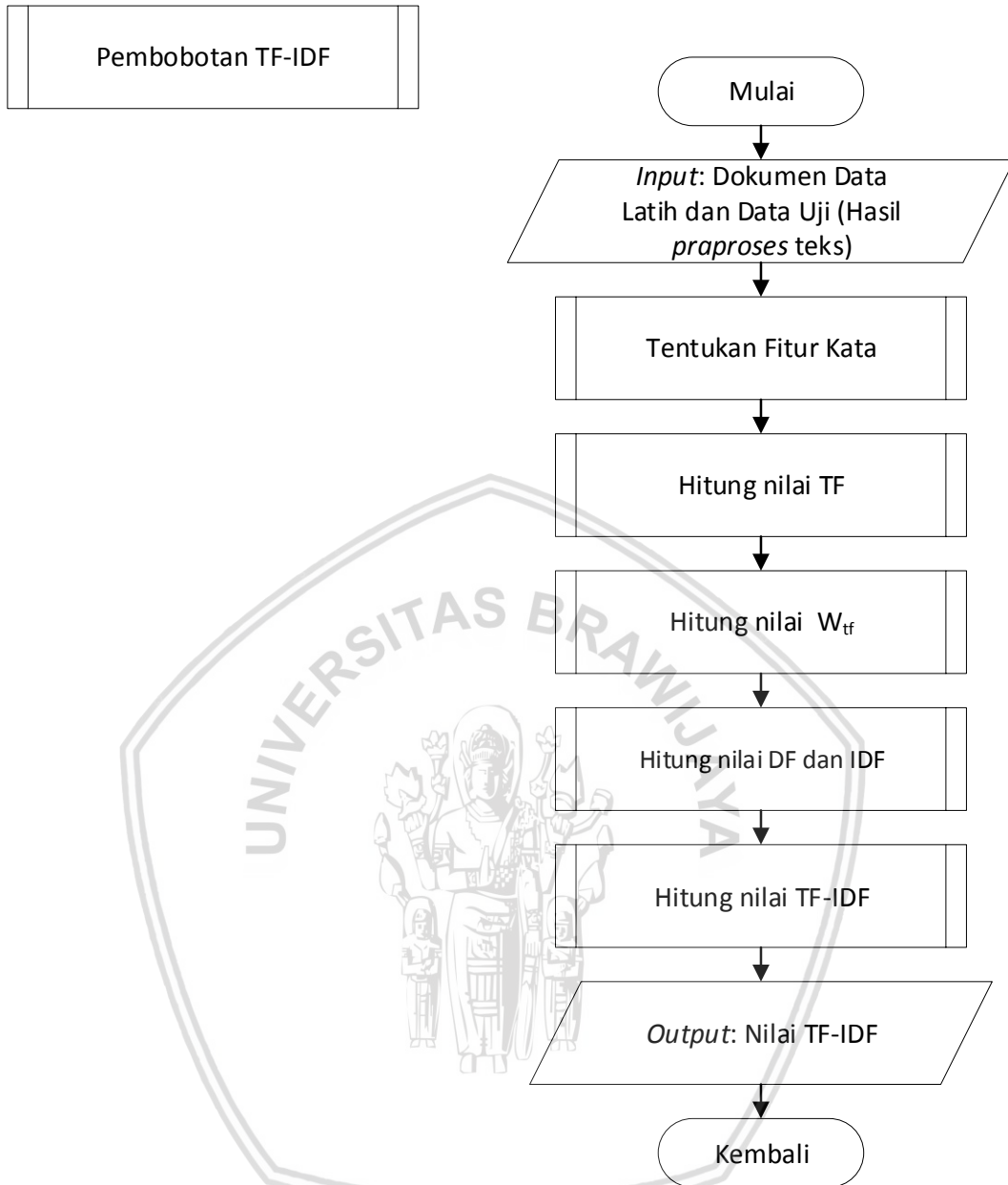
Gambar 0.8 Alur Proses Tokenisasi

Tahapan proses tokenisasi ditunjukkan pada Gambar 4.8 yang dapat dijelaskan pada keterangan berikut:

1. Masukkan dataset dari hasil *stemming*.
2. Perulangan pada jumlah dokumen dataset dengan melakukan proses pemecahan kalimat menjadi per-kata.
3. *Output* dari tahapan ini adalah hasil tokenisasi.

4.4 Pembobotan TF-IDF

Data yang telah melalui tahapan *Pre-processing* telah siap untuk diolah. Pada data mentah tersebut akan dilakukan proses pembobotan pada setiap kata (term). Hasil dari tahapan *Pre-processing* akan dicari fitur-fitur kata untuk dihitung frekuensi dari setiap kata (TF) dan dihitung nilai bobot frekuensi pada setiap kata (W_{tf}). Dari frekuensi kata yang telah ditemukan akan memasuki proses perhitungan dokumen yang mengandung sejumlah kata (DF) dan menghitung inverse dari DF (idf_t). Dari proses perhitungan term frekuensi hingga inverse dari DF, keduanya akan dikalikan dan diproses menjadi pembobotan TF-IDF. Hasil dari pembobotan ini yang akan digunakan dalam proses klasifikasi dengan metode SVM. Alur prosesnya ditunjukkan pada Gambar 4.9.



Gambar 0.9 Alur Proses Pembobotan TF-IDF

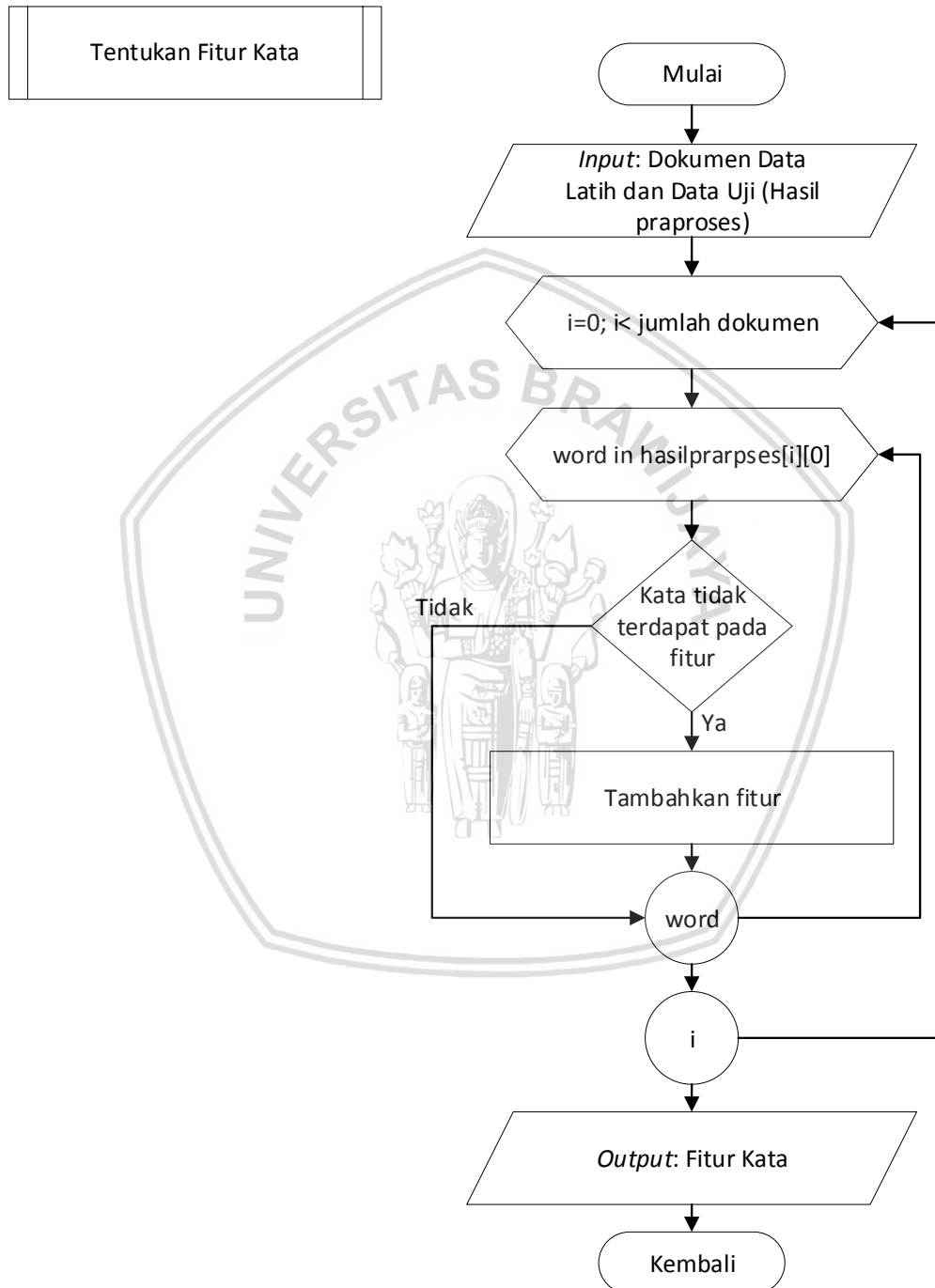
Tahapan proses pembobotan TF-IDF dipecah menjadi beberapa sub-proses seperti yang ditunjukkan pada Gambar 4.9 dengan langkah proses sebagai berikut:

1. Masukkan dataset dari hasil *Pre-processing*.
2. Sub-proses dalam menentukan fitur kata
3. Sub-proses perhitungan nilai TF
4. Sub-proses perhitungan nilai W_{tf}
5. Sub-proses perhitungan nilai DF dan IDF
6. Sub-proses perhitungan nilai TF-IDF
7. Output yang dihasilkan adalah nilai TF-IDF



4.4.2 Tentukan Fitur Kata

Pada tahapan ini dilakukan untuk mendata kata-kata unik yang ada pada dokumen yang telah dimasukkan. Kata-kata tersebut akan dijadikan fitur untuk perhitungan TF-IDF. Alur proses dari menentukan fitur kata ditunjukkan pada Gambar 4.10.



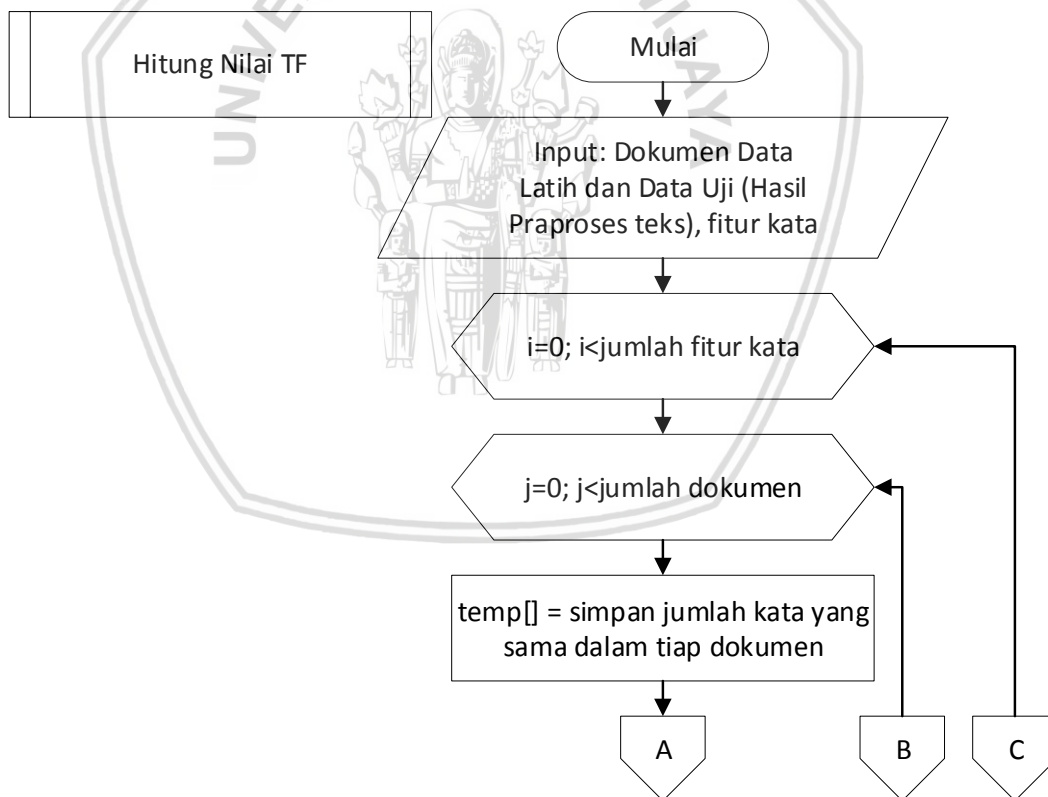
Gambar 0.10 Alur Proses Menentukan Fitur Kata

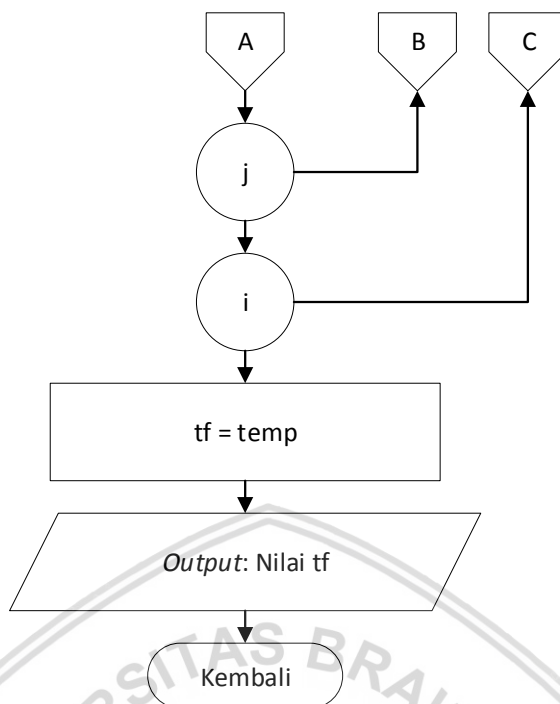
Tahapan proses dalam menentukan fitur kata ditunjukkan pada Gambar 4.10 yang dijelaskan pada keterangan berikut:

1. Memasukkan dataset dari hasil *Pre-processing* teks sebelumnya.
2. Perulangan bersarang yang dilakukan hingga jumlah maksimal dokumen dan pada jumlah kata perbaris dari hasil *Pre-processing* (*word in hasilprarposes [i][0]*).
3. Dalam perulangan dilakukan proses kondisi *if-else*. Jika belum terdapat kata pada list maka disimpan sebagai fitur. Namun jika sudah terdapat kata yang sama pada list maka melakukan pengecekan pada kata selanjutnya.
4. *Output* yang dihasilkan berupa hasil fitur kata.

4.4.3 Hitung nilai TF

Proses pembobotan pada setiap term ditunjukkan pada Gambar 4.11, dimana data yang dimasukkan adalah hasil dari *Pre-processing*. Untuk memulai proses perhitungan TF akan dilakukan perulangan pada jumlah fitur kata yang telah ditentukan dan pada Panjang dokumen yang dimasukkan. Jika terdapat kata (term) yang sama maka nilai TF akan bertambah seiring dengan jumlah frekuensinya. Proses perhitungan TF digambarkan dengan rumus seperti pada persamaan 2.1.





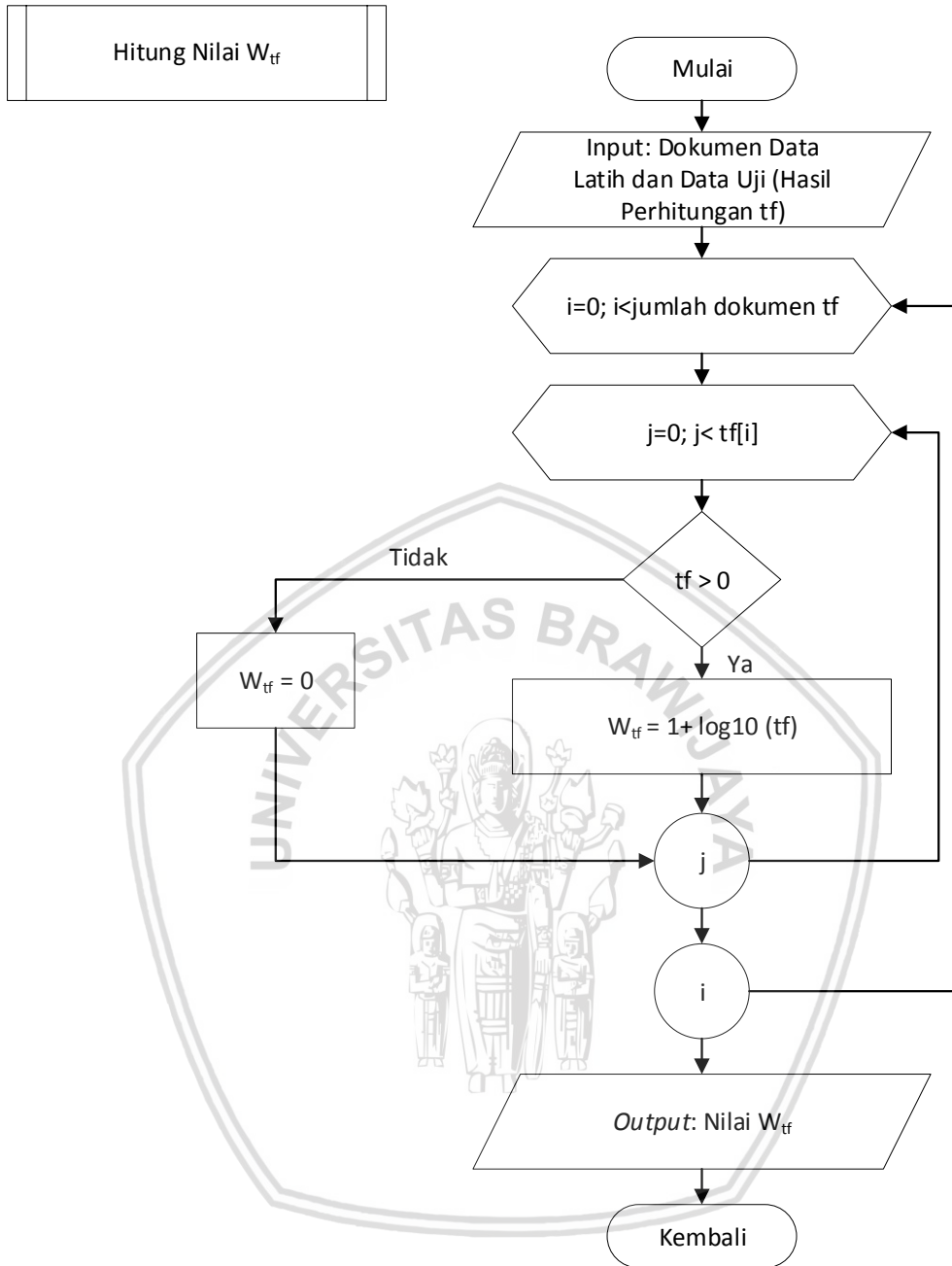
Gambar 0.11 Alur Proses Hitung Nilai TF

Tahapan proses perhitungan nilai TF yang ditunjukkan pada Gambar 4.11 dapat diuraikan pada keterangan berikut:

1. Masukkan data berupa hasil *Pre-processing* teks dan hasil pengumpulan fitur kata.
2. Perulangan bersarang pada jumlah fitur kata yang didapatkan dan pada jumlah dokumen data.
3. Pada perulangan bersarang dijalankan proses penyimpanan nilai tf pada list temp (sementara).
4. Pada list temp yang telah menampung hasil tf akan dialihkan pada list tf.
5. *Output* yang dihasilkan adalah nilai tf.

4.4.4 Hitung nilai W_{tf}

Proses selanjutnya dalam tahapan perhitungan TF-IDF adalah menghitung bobot dari setiap term yang telah ditemukan frekuensinya atau perhitungan *weighting term frequency* (W_{tf}). Dalam melakukan pembobotan akan dilakukan input dokumen dari hasil perhitungan TF. Alur proses yang ditunjukkan dalam tahap ini ditunjukkan pada Gambar 4.12.



Gambar 0.12 Alur Proses Hitung Nilai W_{tf}

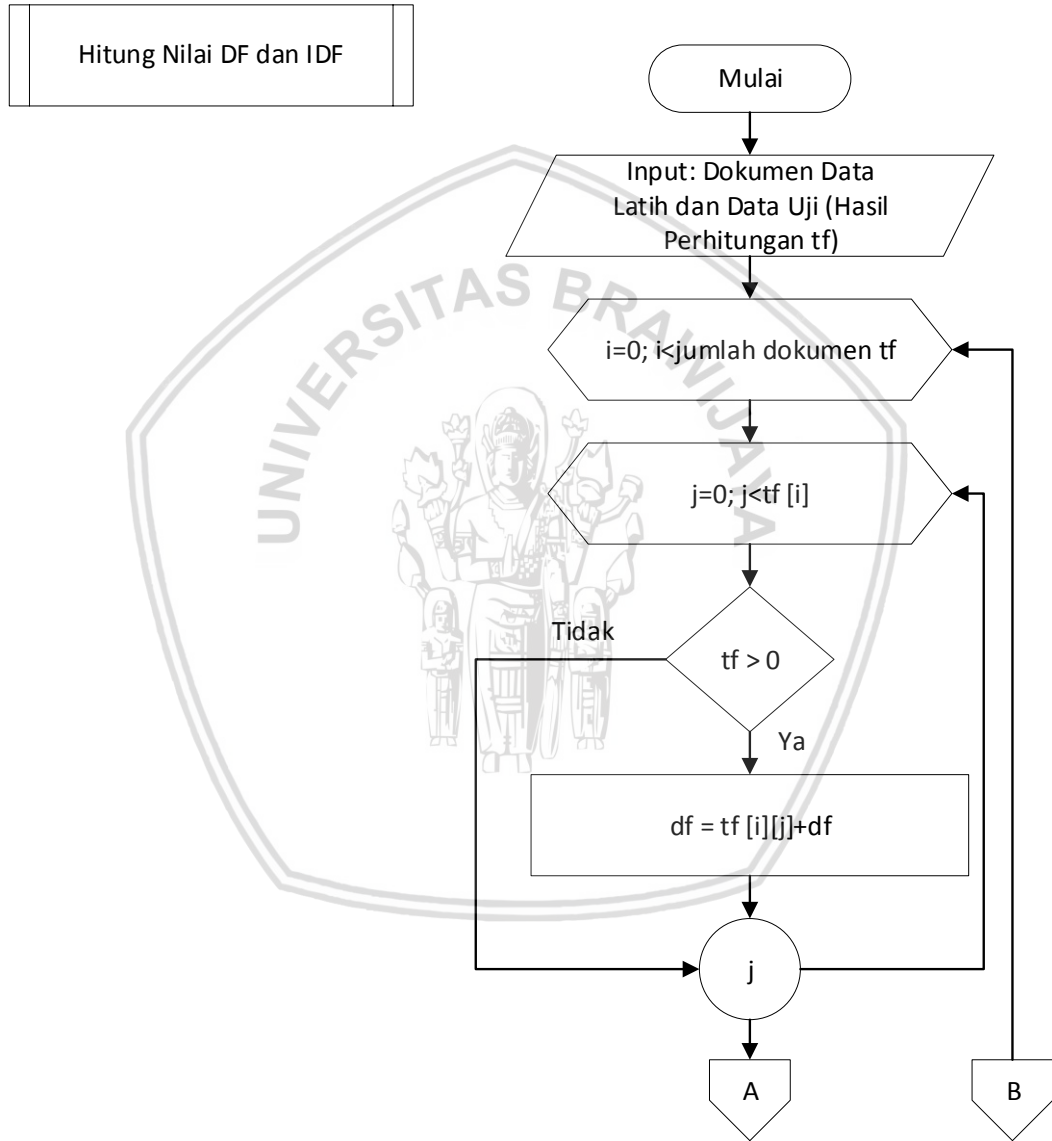
Tahapan proses pada perhitungan nilai W_{tf} ditunjukkan pada Gambar 4.12 diuraikan pada keterangan berikut:

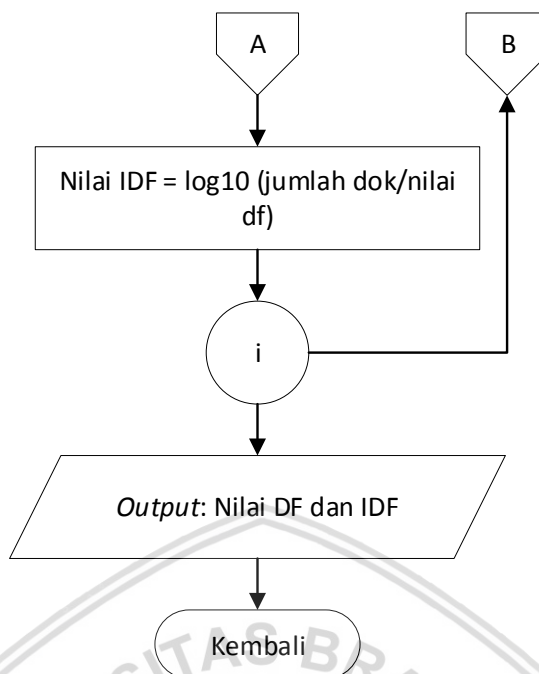
1. Masukkan dataset dari hasil perhitungan nilai tf .
2. Perulangan bersarang pada panjang dokumen tf dan perulangan pada banyak dokumen yang dihitung berdasarkan perbaris pada dokumen tf .
3. Pada perulangan terdapat kondisi *if-else*. Jika terdapat nilai tf yang lebih dari 0 maka akan dihitung nilai W_{tf} sesuai dengan Persamaan 2.1. Namun jika nilai tf tidak lebih dari 0, maka nilai W_{tf} adalah 0.
4. *Output* yang dihasilkan adalah nilai W_{tf}



4.4.5 Hitung Nilai DF dan IDF

Subproses yang dilakukan pada pembobotan TF-IDF adalah menghitung nilai *inverse document frequency*. Untuk memulai perhitungan, perlu diketahui nilai document frequency (DF). Nilai DF ditunjukkan dengan jumlah dokumen yang mengandung kata tersebut. Namun untuk nilai IDF didapatkan dari kebalikan nilai DF, sehingga semakin sedikit nilai DF yang ditemukan maka bobot kata akan yang ditemukan akan semakin tinggi pada nilai IDF. Perhitungan ini ditunjukkan pada Gambar 4.13.





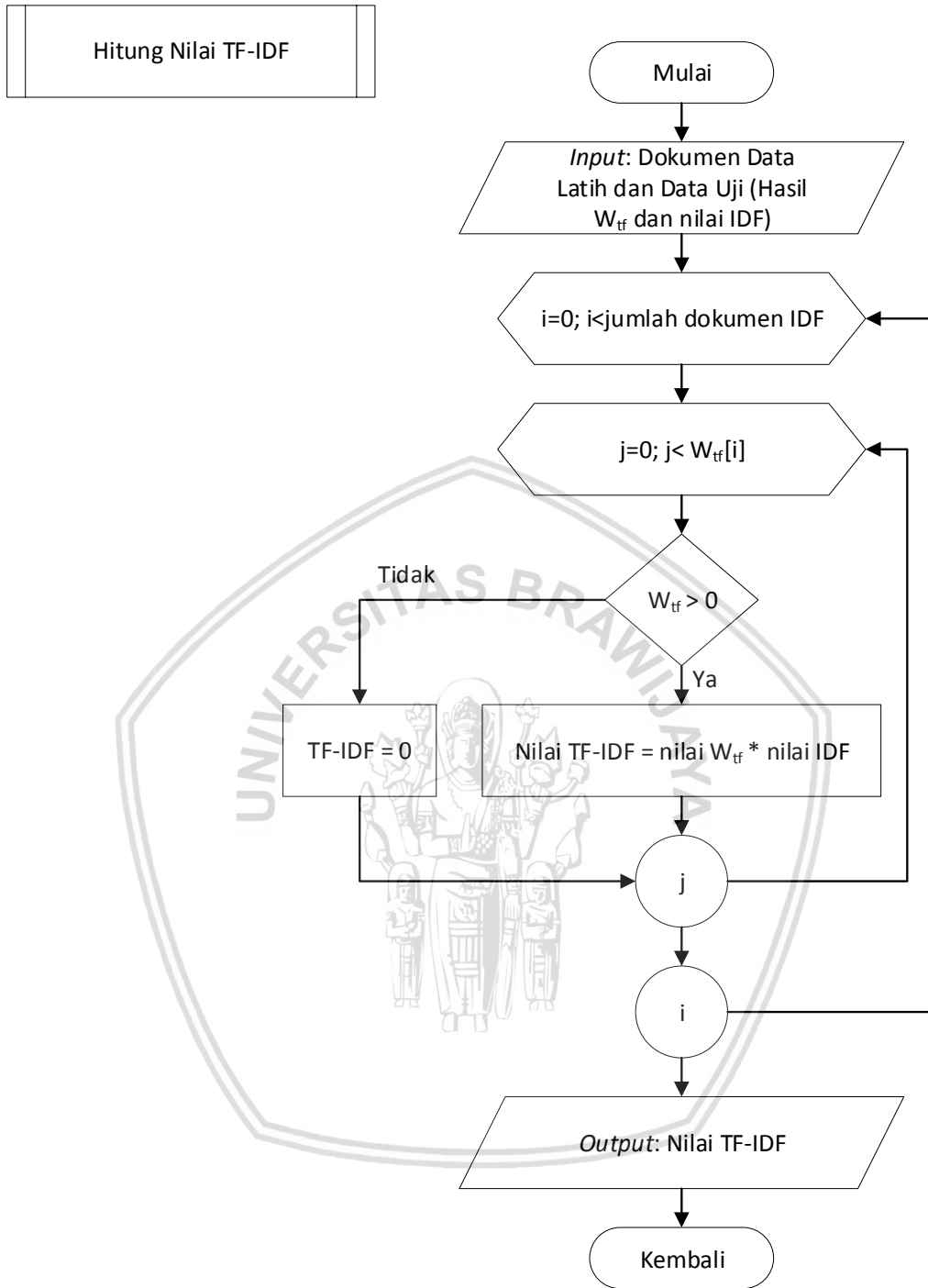
Gambar 0.13 Alur Proses Hitung Nilai DF dan IDF

Tahapan proses perhitungan nilai DF dan IDF ditunjukkan pada Gambar 4.13 yang dapat diuraikan pada keterangan berikut:

1. Masukkan dataset dari hasil perhitungan nilai tf.
2. Perulangan bersarang pada jumlah dokumen tf dan banyak dokumen perbaris pada dokumen tf.
3. Proses kondisi *if-else* dilakukan pada perulangan adalah pengecekan nilai tf yang lebih dari 0. Jika nilai tf lebih dari 0 maka dihitung nilai df, yaitu menghitung jumlah fitur (kata) pada keseluruhan dokumen. Kondisi *if* yang tidak terpenuhi akan menjalankan perulangan kembali.
4. Setiap fitur yang telah diketahui nilai df akan dihitung nilai idf sesuai dengan Persamaan 2.2.
5. *Output* yang dihasilkan adalah nilai DF dan IDF.

4.4.6 Hitung Nilai TF-IDF

Pembobotan kata akan didapatkan ketika perhitungan nilai TF-IDF dilakukan. Perhitungan TF-IDF didapatkan dari hasil perkalian nilai W_{tf} (*Weighting term frequency*) dan nilai IDF (*inverse document frequency*). Perkalian ini menunjukkan bahwa setiap kata yang tertinggi akan mewakili kata yang jarang ditemukan pada dokumen lainnya dan begitu juga sebaliknya. Subproses ini ditunjukkan pada Gambar 4.14.



Gambar 0.14 Alur Proses Hitung Nilai TF-IDF

Tahapan proses perhitungan nilai TF-IDF ditunjukkan pada Gambar 4.14 dapat diuraikan pada keterangan berikut:

1. Masukkan dataset dari hasil perhitungan nilai W_{tf} dan IDF.
2. Perulangan bersarang pada jumlah dokumen IDF dan banyak banyak dokumen perbaris pada data W_{tf} ($W_{tf}[i]$).

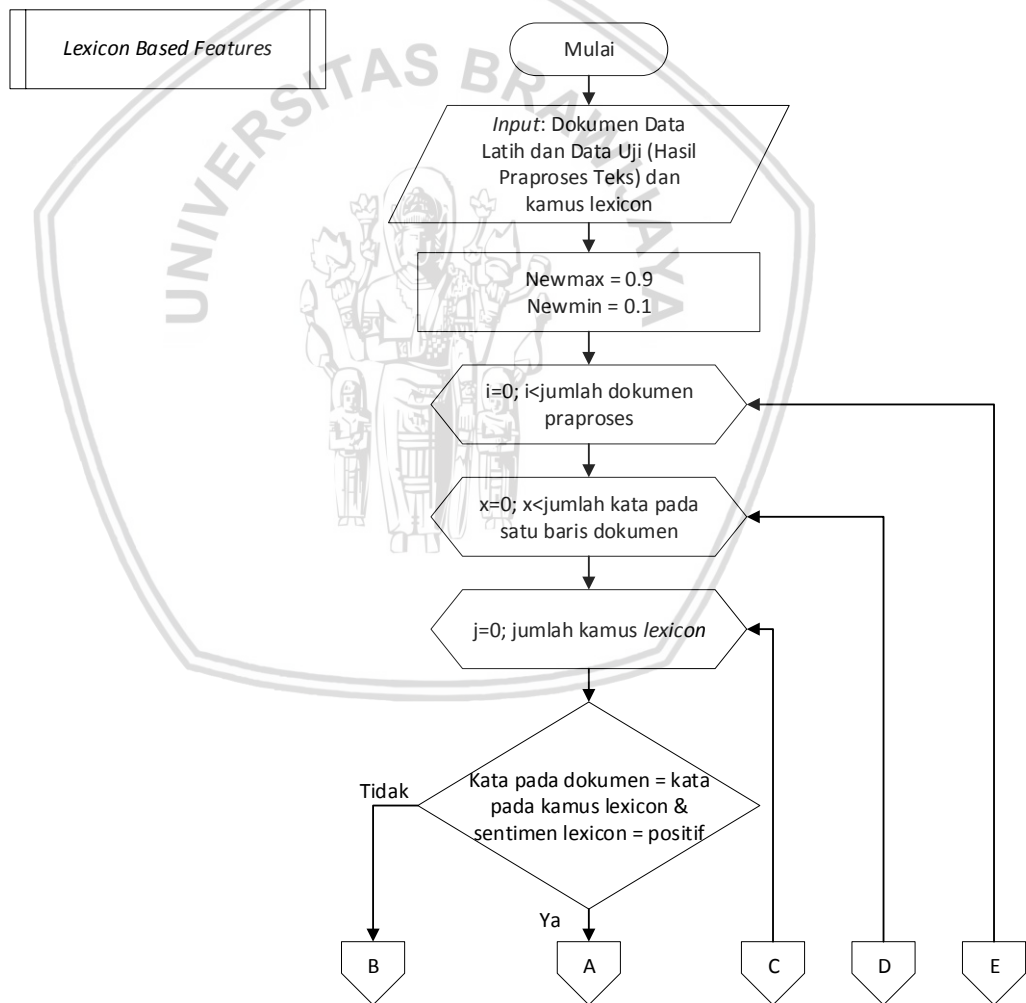


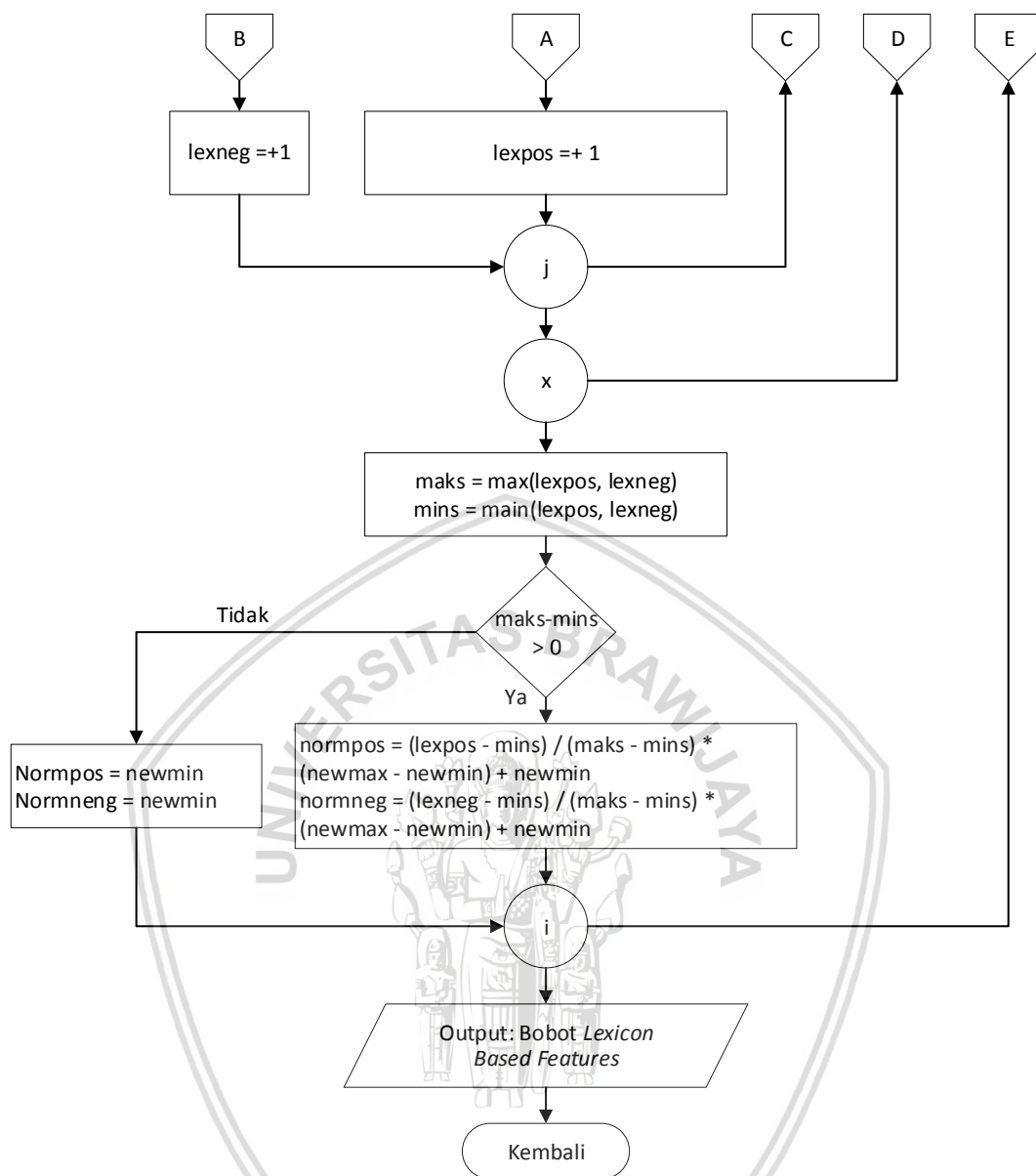
3. Dilakukan proses pengecekan nilai W_{tf} . Jika nilai W_{tf} lebih dari 0 maka dilakukan perhitungan sesuai dengan Persamaan 2.3. Namun jika kondisi *if* tidak terpenuhi, nilai TF-IDF adalah 0.
4. *Output* yang dihasilkan adalah nilai TF-IDF.

4.5 Lexicon Based Features

Lexicon Based Features merupakan metode yang diterapkan pada pengujian sistem ini. Dengan diterapkannya metode *Lexicon Based Features* akan menambah pula bobot dari dokumen yang memiliki fitur-fitur bersentimen positif ataupun negatif. Alur proses pada implementasi metode *Lexicon Based Features* ditunjukkan pada Gambar 4.15 dan Gambar 4.16.

4.5.1 Normalisasi Min-Max





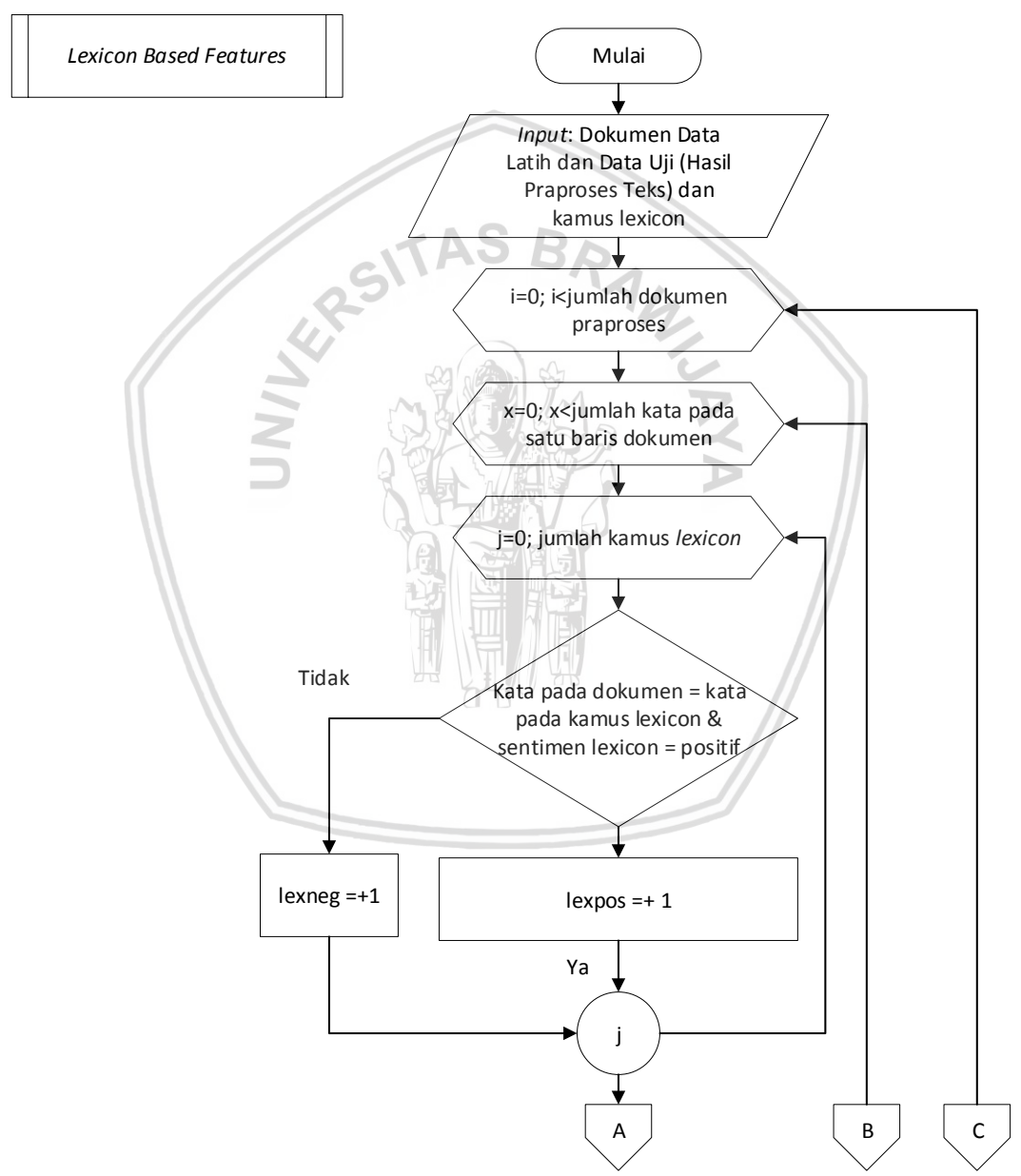
Gambar 0.15 Alur Proses *Lexicon Based Features* dengan normalisasi *min-max*

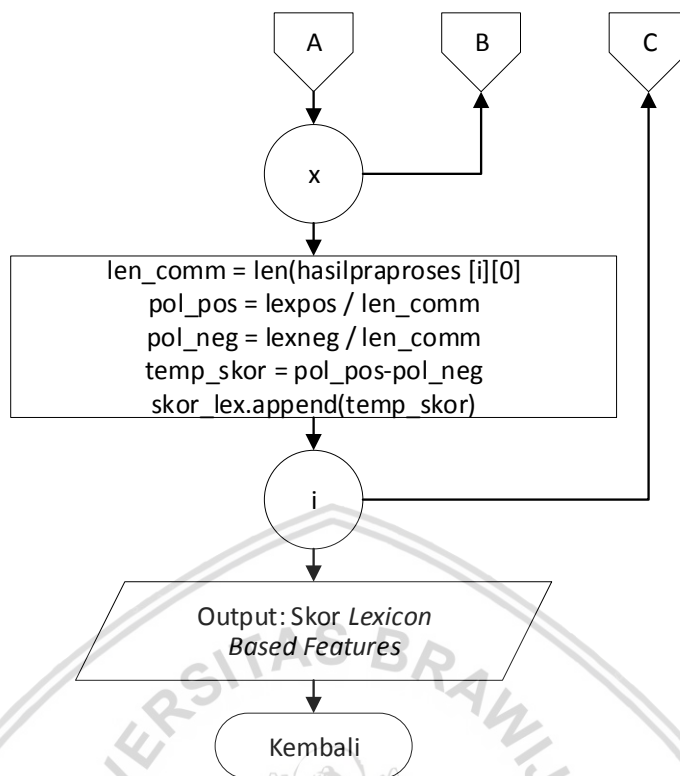
Tahapan proses perhitungan bobot lexicon ditunjukkan pada Gambar 4.15 yang dapat diuraikan pada keterangan berikut:

1. Masukkan dataset dari hasil *Pre-processing* .
2. Inialisasi variabel *newmax* yang bernilai 0.9 dan *newmin* dengan 0.1.
3. Perulangan bersarang pada jumlah dokumen *Pre-processing* dan jumlah kata pada satu baris dokumen *Pre-processing*.
4. Dalam perulangan dilakukan proses kondisi *if-else* yang berguna untuk mengecek setiap kata pada prarposes dengan kamus lexicon yang ada. Jika kata bersentimen positif maka variabel *Lexpos* bertambah 1 dan jika kata bersentimen negatif maka variabel *Lexneg* bertambah 1 seiring perulangan yang dilakukan.

5. Perbandingan nilai maksimal dan minimal dilakukan antara nilai yang ada pada variabel *lexneg* dan *lexpos*.
6. Kondisi *if-else* dilakukan dengan kondisi pengurangan pada variabel maks dan mins yang kurang dari 0. Jika kondisi terpenuhi dilakukan normalisasi min-max sesuai dengan Persamaan 2.4. Jika kondisi tidak terpenuhi maka hasil normalisasi positif dan negatif secara otomatis bernilai *newmin*.
7. *Output* yang dihasilkan adalah nilai bobot *Lexicon Based Features*

4.5.2 Skor Sentimen





Gambar 0.16 Alur Proses *Lexicon Based Features* dengan skor sentimen

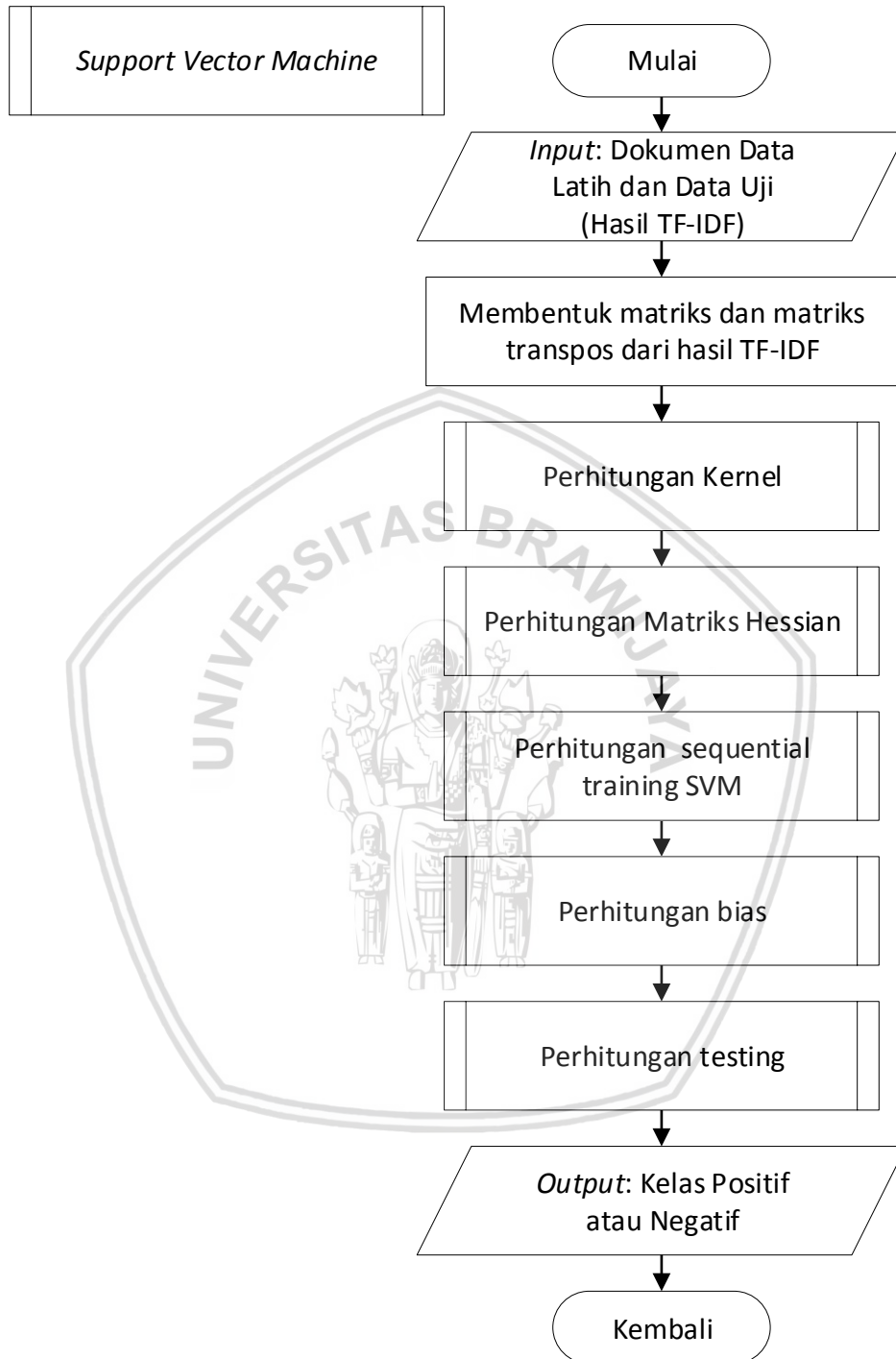
Tahapan proses perhitungan bobot lexicon ditunjukkan pada Gambar 4.16 yang dapat diuraikan pada keterangan berikut:

1. Masukkan dataset dari hasil *Pre-processing* .
2. Perulangan bersarang pada jumlah dokumen *Pre-processing* dan jumlah kata pada satu baris dokumen *Pre-processing*.
3. Dalam perulangan dilakukan proses kondisi *if-else* yang berguna untuk mengecek setiap kata pada prarproses dengan kamus lexicon yang ada. Jika kata bersentimen positif maka variabel Lexpos bertambah 1 dan jika kata bersentimen negatif maka variabel Lexneg bertambah 1 seiring perulangan yang dilakukan.
4. Perhitungan nilai polaritas pada setiap sentimen positif dan negatif pada setiap dokumen komentar. Nilai polaritas pada sentimen positif akan dikurangi dengan polaritas sentimen negatif dan akan menjadi skor sentimen
5. *Output* yang dihasilkan adalah nilai bobot *Lexicon Based Features*

4.6 Support Vector Machine

Metode *Support Vector Machine* merupakan metode yang digunakan untuk analisis sentimen pada penelitian ini. Perhitungan dengan klasifikasi ini dilakukan setelah melalui tahap *Pre-processing* dan pembobotan nilai TF-IDF. Hasil yang akan ditentukan dengan metode ini adalah klasifikasi kelas positif dan kelas negatif yang

didapat berdasarkan bobot pada setiap fitur dokumen teks. Alur proses metode *Support Vector Machine* ditunjukkan pada Gambar 4.17.



Gambar 0.17 Alur Proses *Support Vector Machine*

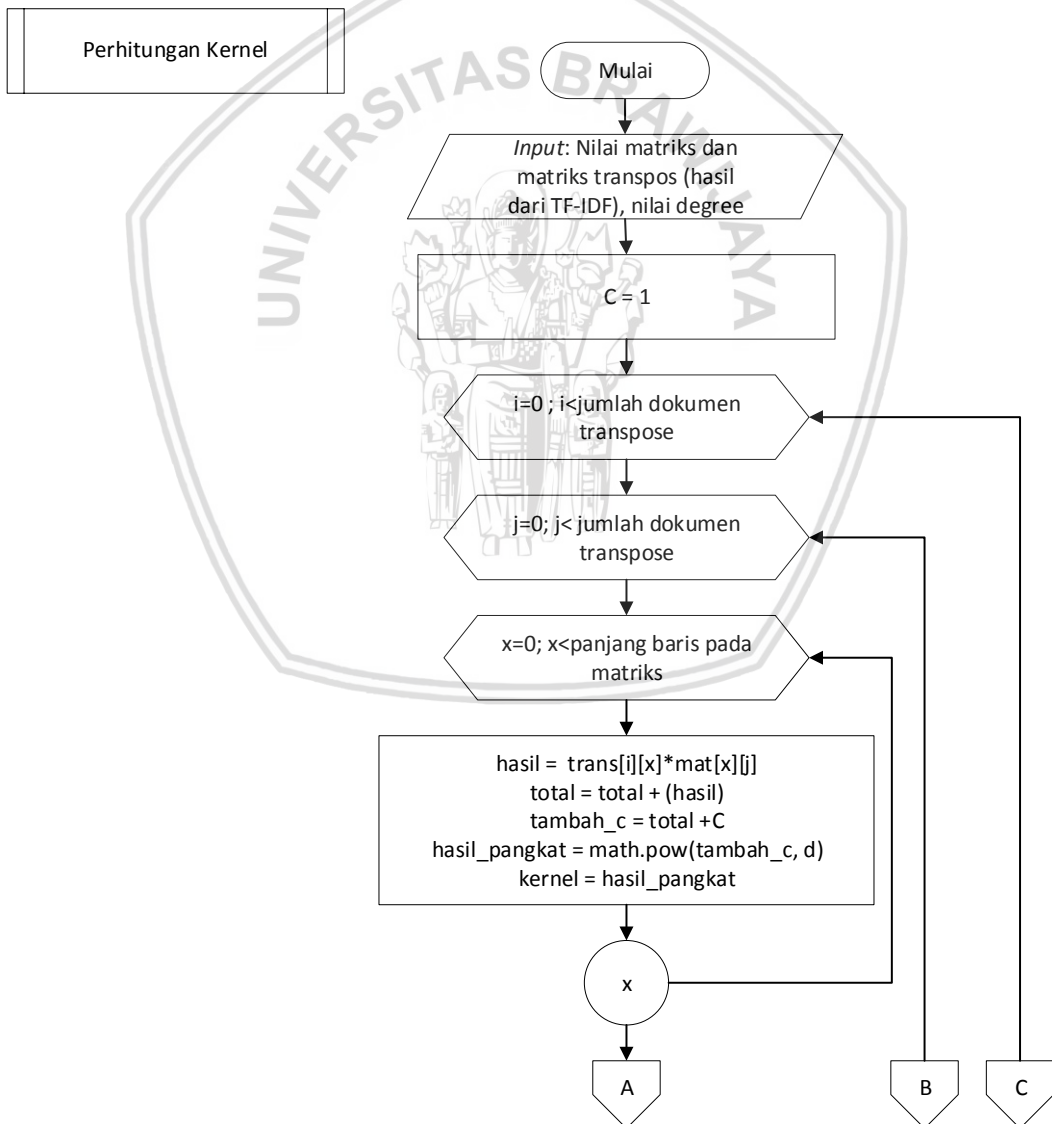
Tahapan alur proses *Support Vector Machine* melalui beberapa sub-proses, diantaranya adalah:

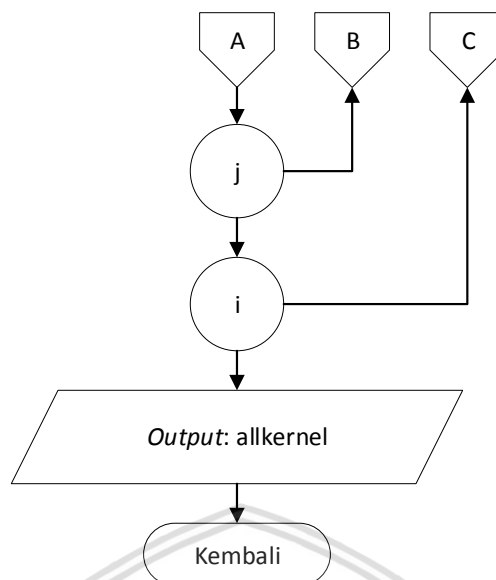
1. Masukkan dataset dari hasil perhitungan TF-IDF.
2. Proses pembentukan matriks dan matriks transpos dari dataset TF-IDF.

3. Sub-proses perhitungan Kernel Polynomial.
4. Sub-proses perhitungan Matriks Hessian.
5. Sub-proses perhitungan *Sequential Training SVM*.
6. Sub-proses perhitungan bias.
7. Sub-proses perhitungan data testing.
8. *Output* hasil kelas prediksi berupa kelas positif dan kelas negatif.

4.6.2 Perhitungan Kernel

Tahap awal untuk memulai metode *Support Vector Machine* yaitu dengan perhitungan kernel yang menggunakan jenis kernel polynomial sesuai dengan Persamaan 2.10. Data yang digunakan merupakan hasil dari perhitungan pembobotan TF-IDF. Tahapan perhitungan kernel dapat dilihat pada Gambar 4.18.





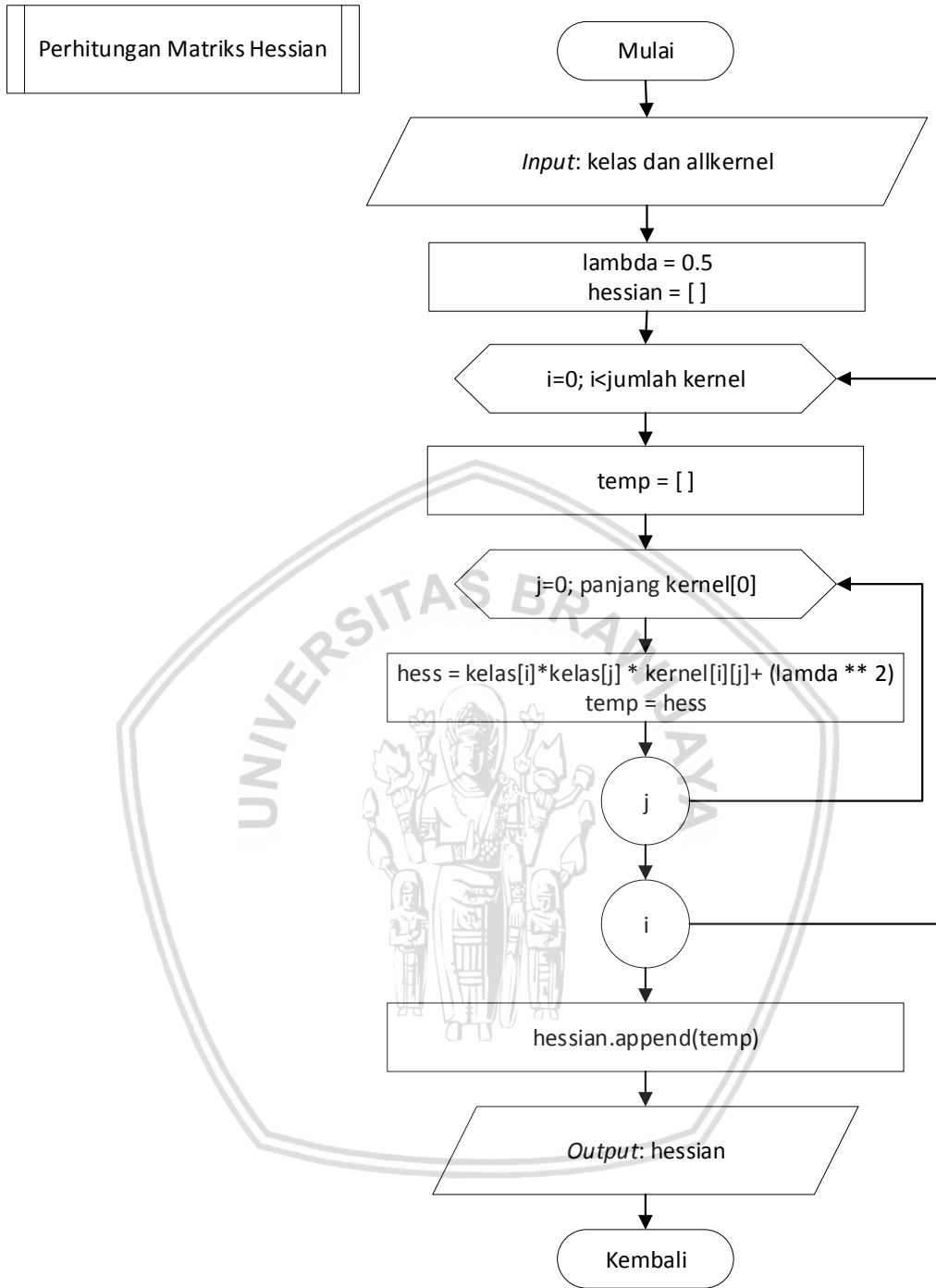
Gambar 0.18 Alur Proses Perhitungan Kernel

Tahapan proses perhitungan kernel dengan menggunakan jenis Kernel Polynomial ditunjukkan pada Gambar 4.18 dapat diuraikan pada keterangan berikut:

1. Masukkan dataset adalah data latih dari hasil perhitungan TF-IDF dalam bentuk matriks dan matriks transpos. Masukkan lainnya adalah parameter nilai *degree*.
2. Inisialisasi konstanta C yang dibutuhkan dalam perhitungan kernel polynomial.
3. Perulangan bersarang terhadap panjang dokumen TF-IDF yang telah ditranspos dan panjang baris pada matriks TF-IDF yang tidak di transpos.
4. Dalam perulangan terdapat proses perhitungan nilai kernel, yaitu mengkalikan nilai dari matriks transpos TF-IDF dengan matriks TF-IDF yang disimpan pada variabel hasil. Proses perhitungan kernel selanjutnya sesuai dengan Persamaan 2.10.
5. *Output* yang dihasilkan pada tahapan ini adalah matriks yang berisi nilai kernel.

4.6.3 Perhitungan Matriks Hessian

Perhitungan matriks hessian dilakukan pada tahap kedua klasifikasi dengan metode *Support Vector Machine*. Perhitungan ini dilakukan pada hasil perhitungan kernel data ke-x dan data ke-y dengan hasil matriks perkalian kelas positif (1) dan kelas negatif (-1) yang kemudian dikalikan dengan lambda yang telah ditentukan. Alur proses dalam perhitungan matriks hessian ditunjukkan pada Gambar 4.19.



Gambar 0.19 Alur Proses Perhitungan Matriks Hessian

Tahapan proses perhitungan matriks Hessian yang ditunjukkan pada Gambar 4.19 dapat diuraikan pada keterangan berikut:

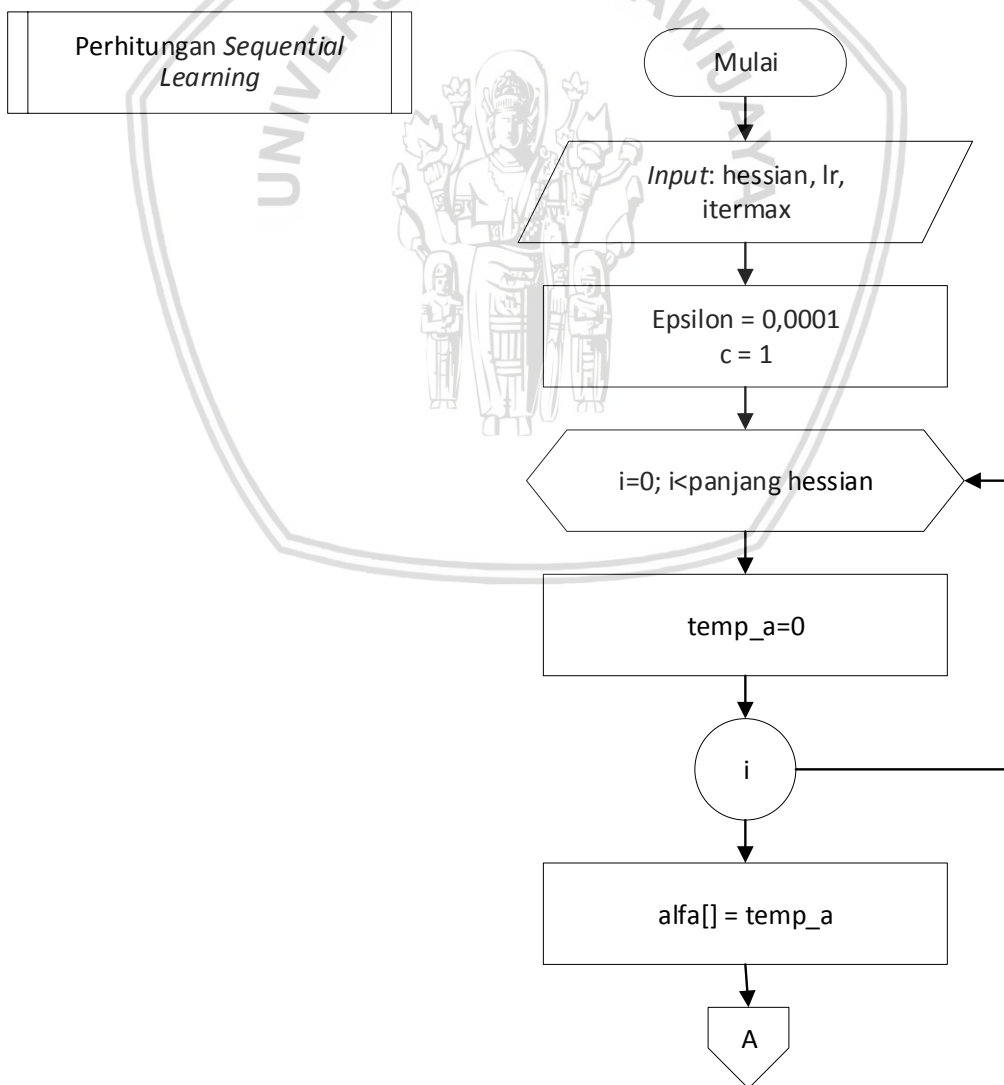
1. Masukkan yang dibutuhkan adalah kelas dari data latih dan hasil perhitungan kernel.
2. Inisialisasi variabel lambda yang bernilai 0.5.
3. Perulangan bersarang pada panjang kernel data latih dan panjang data pada satu baris kernel.

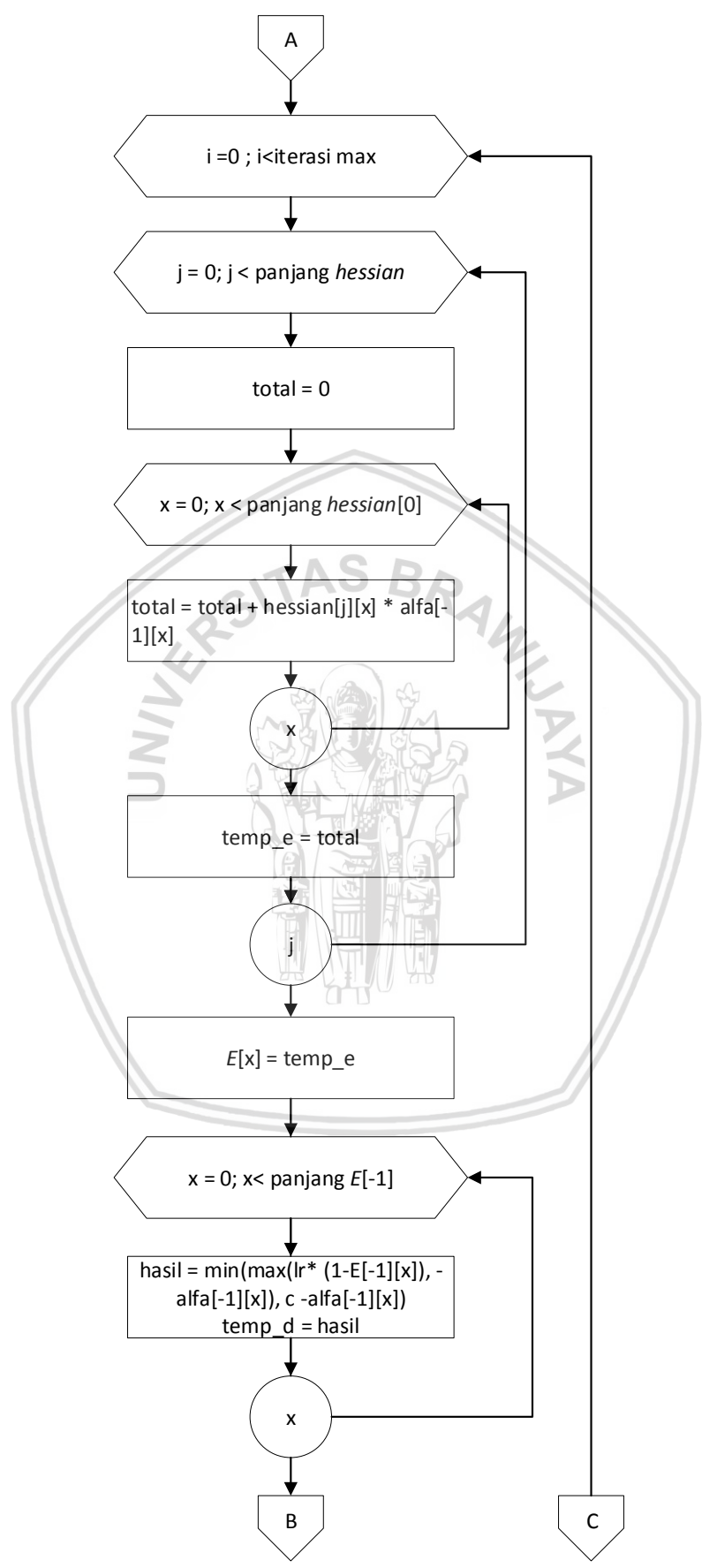


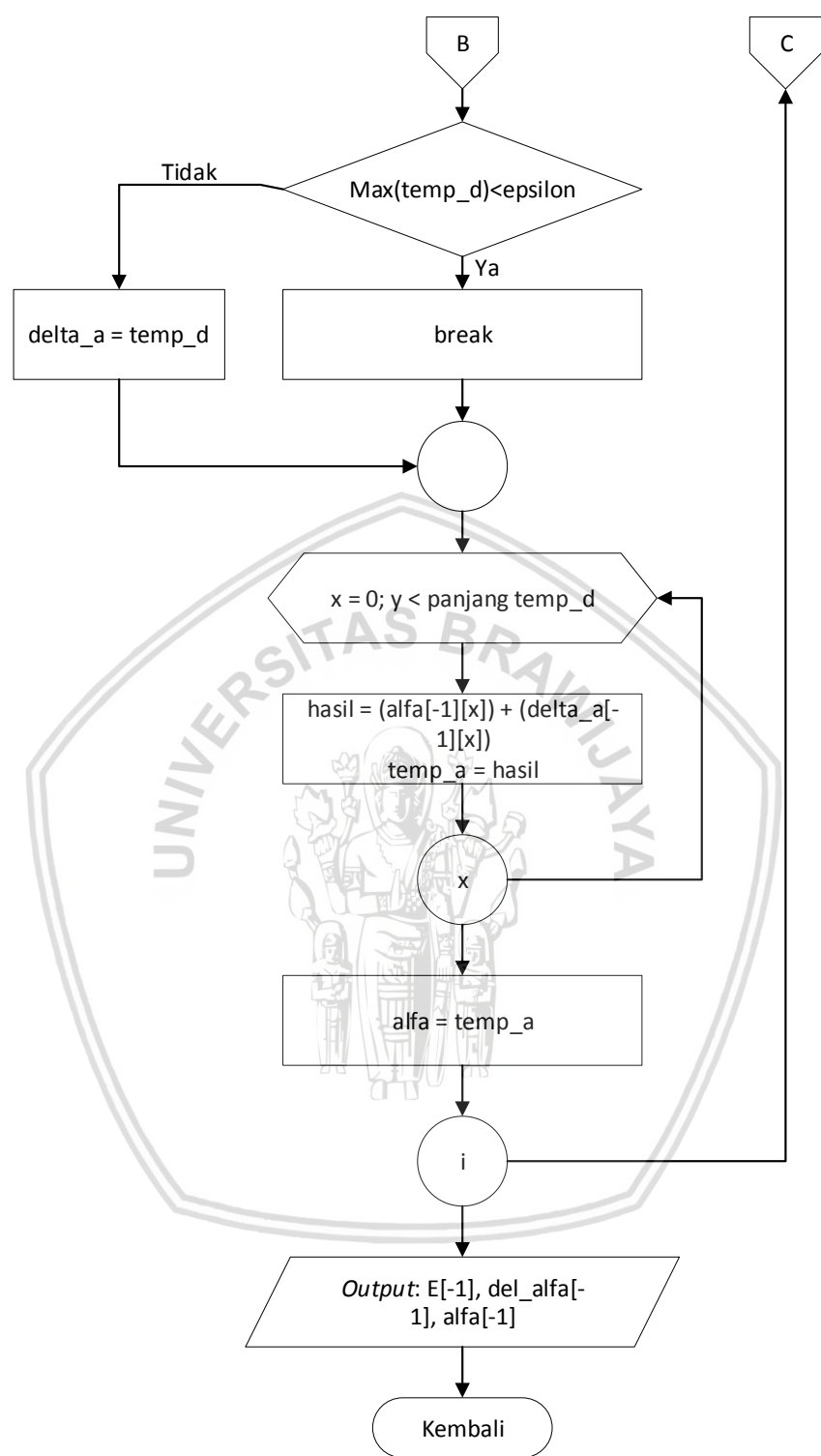
4. Dalam perulangan dilakukan proses perhitungan nilai matriks *hessian* sesuai dengan Persamaan 2.13 yang hasilnya ditampung pada list temp.
5. Setelah seluruh nilai matriks *hessian* ditemukan dan telah ditampung pada list temp, selanjutnya nilai keseluruhan tersebut akan dimasukkan pada list *hessian*.
6. *Output* yang dihasilkan pada tahapan ini adalah matriks *hessian*.

4.6.4 Perhitungan *Sequential Training SVM*

Subproses pada metode SVM adalah perhitungan *sequential training*. Pada tahap ini dilakukan perhitungan nilai E , $\delta\alpha$, dan α . Proses perhitungan *sequential training* dilakukan sebanyak jumlah iterasi maksimum yang dimasukkan. Untuk menghentikan iterasi dapat dilakukan dengan cara membandingkan nilai $\delta\alpha$ dengan nilai epsilon ϵ . Jika nilai $\delta\alpha$ lebih kecil dari nilai epsilon, maka iterasi akan berhenti sebelum mencapai iterasi maksimum. Nilai akhir dari $\delta\alpha$ merupakan nilai support vector, yaitu titik data terdekat dengan hyperplane antar kelas data. Alur proses perhitungan *sequential training SVM* ditunjukkan pada Gambar 4.20.







Gambar 0.20 Alur Proses Perhitungan *Sequential Learning SVM*

Tahapan proses pada perhitungan *sequential learning SVM* ditunjukkan pada Gambar 4.20 dapat diuraikan pada keterangan berikut:

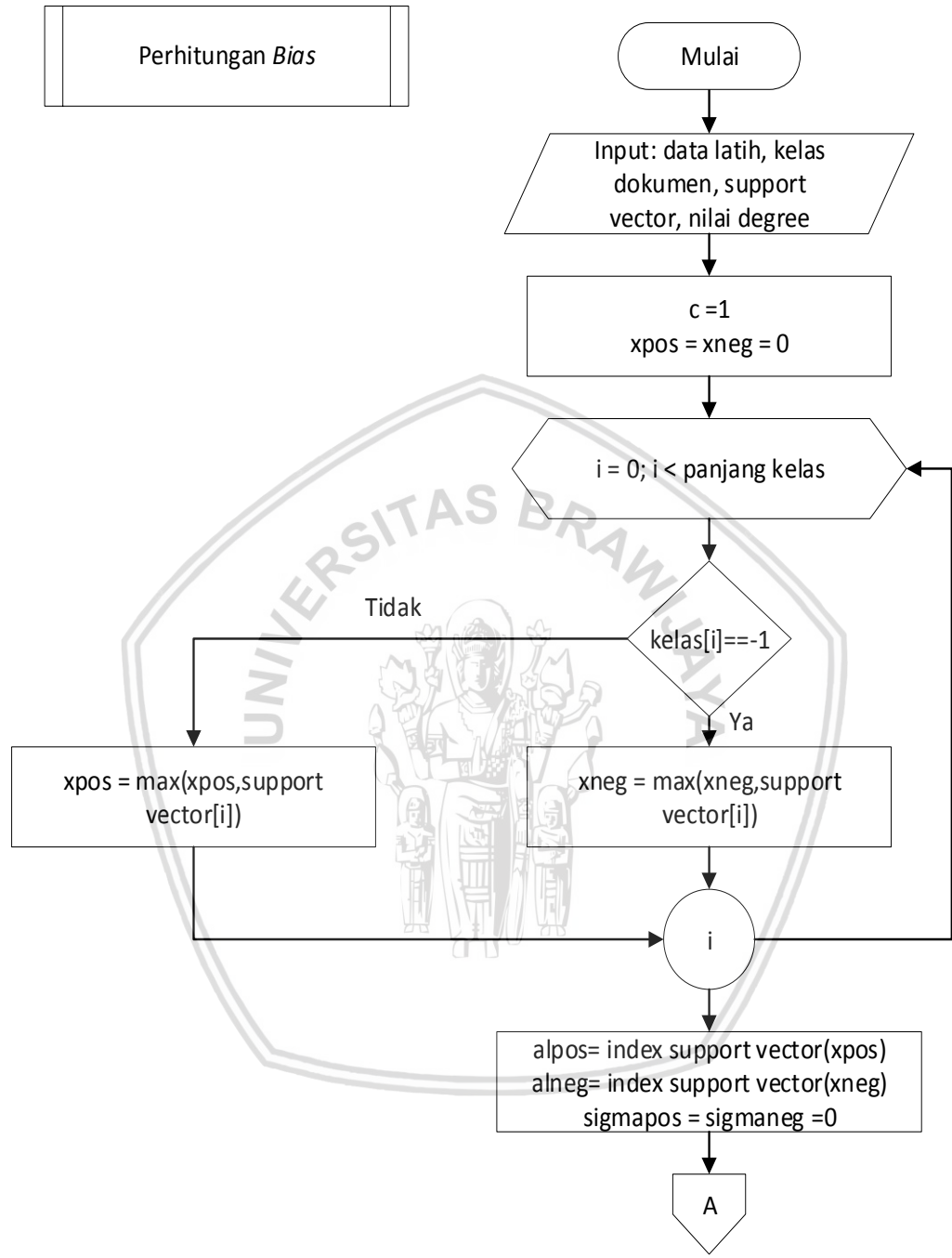
1. Masukkan dataset adalah hasil perhitungan matriks hessian, parameter nilai *learning rate*, dan iterasi maksimum yang dihendaki.

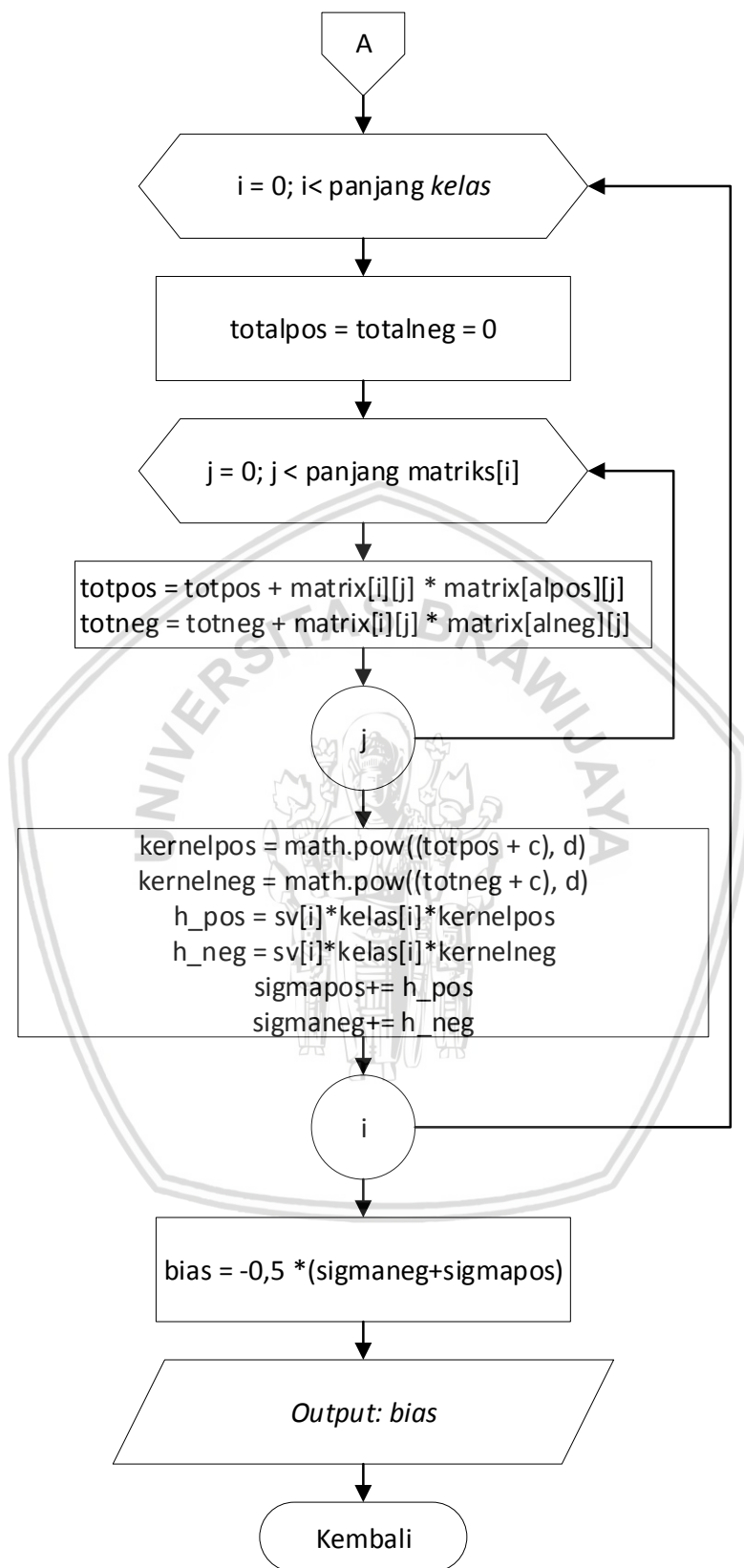
2. Inisialisasi variabel epsilon yang bernilai 0,00011 dan nilai konstanta c yang bernilai 1.
3. Perulangan pada panjang data matriks hessian yang digunakan untuk menginisialisasi nilai alfa awal.
4. Perulangan bersarang yang dilakukan dengan kondisi hingga iterasi maksimum yang dimasukkan, panjang data matriks hessian, panjang data perbaris pada matriks hessian.
5. Didalam perulangan panjang data matriks hessian dilakukan inisialisasi variabel total yang mulanya bernilai 0.
6. Proses perhitungan awal pada *sequential learning* adalah mencari nilai E yang disimpan pada variabel total.
7. Hasil setiap perhitungan nilai E yang telah dimasukkan pada variabel total akan disimpan pada list temp_e dan kemudian disimpan kembali pada list E pada kolom-kolom tertentu sesuai dengan perhitungannya.
8. Setelah nilai E didapatkan, maka dihitung nilai delta alfa terlebih dahulu dengan membandingkan nilai minimum dan maksimum yang nilainya disimpan pada variabel hasil. Kemudian dari variabel hasil akan disimpan pada list temp_d. Perhitungan nilai delta alfa sesuai dengan Persamaan 2.15.
9. Setelah proses perulangan pada pencarian delta alfa selesai dilakukan, maka akan dibandingkan dengan nilai épsilon yang telah diinisialisasikan. Jika kondisi terpenuhi maka sistem akan *break*. Namun jika tidak terpenuhi maka list temp_d dimasukkan pada list delta_a.
10. Dalam perulangan iterasi terdapat perulangan dimana akan berhenti jika telah mencapai panjang data dari temp_d.
11. Dalam proses perulangan tersebut dilakukan perhitungan alfa pada setiap dokumen dan disimpan pada variabel hasil. Perhitungan alfa ditemukan dengan menjumlahkan nilai alfa sebelumnya dengan nilai delta alfa. Perhitungan ini dapat dilihat pada Persamaan 2.16.
12. Nilai alfa yang telah ditemukan disimpan pada list temp_a.
13. Setelah proses perulangan dalam mencari nilai alfa berakhir, maka nilai alfa yang sebelumnya disimpan pada temp_a disimpan kembali pada list alfa.
14. *Output* yang dihasilkan pada tahapan ini adalah nilai E, delta alfa, dan alfa.

4.6.5 Perhitungan Bias

Tahapan akhir untuk analisis sentimen dengan metode SVM adalah menghitung nilai bias yang berguna dalam pembentukan hyperplane. Tahap ini akan memilih nilai α tertinggi pada tiap kelas positif dan kelas negatif dari support vector. Nilai α tertinggi menunjukkan sebagai pembatas antar kelas data. Hasil support vector yang sebelumnya didapat akan dikalikan dengan hasil perhitungan

kernel polynomial yang memiliki nilai α tertinggi. Perhitungan bias dirumuskan seperti persamaan 2.16. Alur proses ini ditunjukkan pada Gambar 4.21.





Gambar 0.21 Alur Proses Perhitungan Bias

Tahapan proses perhitungan bias yang ditunjukkan pada Gambar 4.21 dapat diuraikan pada keterangan berikut:

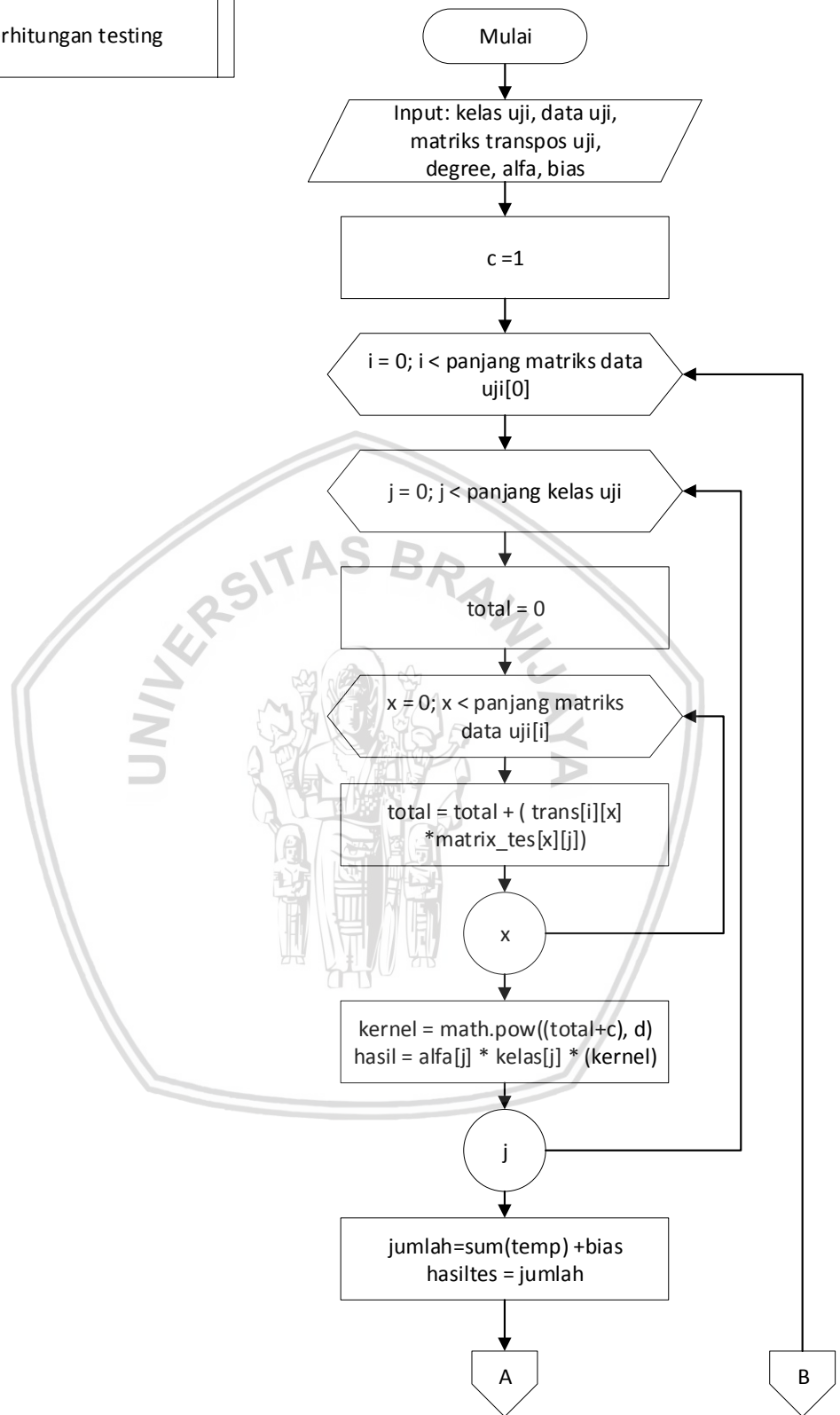


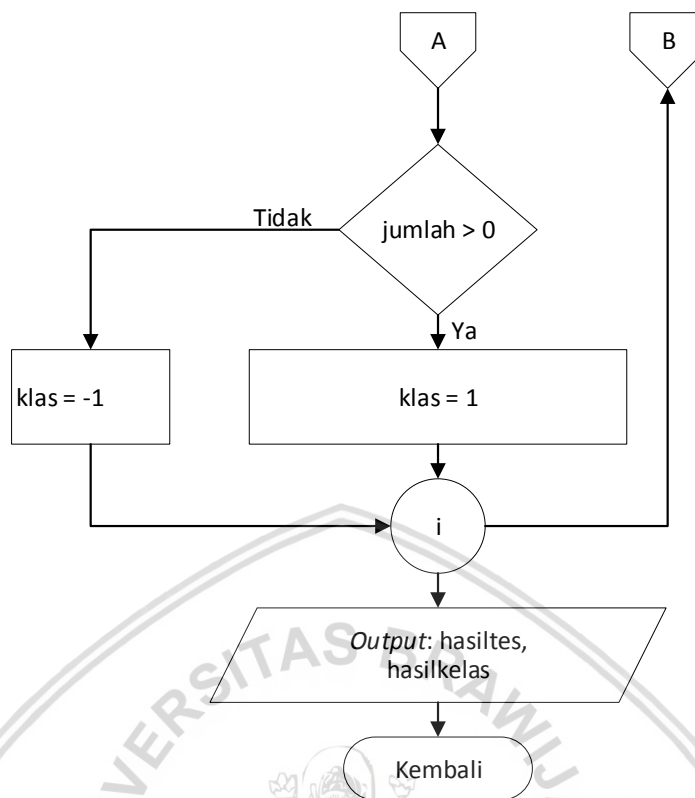
1. Masukkan dataset adalah matriks data latih, nilai support vector, dan nilai *degree*.
2. Inisialisasi dari nilai C, x_{neg} , dan x_{pos} .
3. Perulangan awalnya dilakukan dengan batas panjang kelas data latih.
4. Proses yang dilakukan pada perulangan tersebut adalah membandingkan nilai *support vector* pada setiap kelas positif dan negatif untuk dicari nilai *support vector* tertinggi pada setiap kelas.
5. Setelah perulangan selesai dilakukan terdapat variabel $alpos$ dan $alneg$ yang menyimpan nilai *support vector* untuk kelas positif dan negatif.
6. Perulangan bersarang dilakukan hingga panjang kelas data latih dan panjang data perbaris pada matriks data latih.
7. Dalam perulangan bersarang terdapat proses perhitungan bias yang membutuhkan perkalian kernel dari dokumen yang memiliki nilai alfa tertinggi pada setiap kelas positif dan negatif. Proses perhitungan bias terdapat pada Persamaan 2.17.
8. *Output* yang dihasilkan adalah nilai bias.

4.6.6 Perhitungan *Testing*

Tahapan untuk mendapatkan hasil analisis sentimen adalah menghitung data uji yang ingin diketahui kelasnya. Kelas yang ditunjukkan adalah kelas positif *cyberbullying* atau kelas negatif *cyberbullying*. Data uji yang dicari kelasnya perlu melalui dari tahapan *Pre-processing* hingga perhitungan *Support Vector Machine* sama seperti yang dilakukan pada data latih. Untuk mencari kelas data uji maka data uji tersebut akan dikalikan dengan data latih untuk mendapatkan hasil kernel polynomial. Hasil perkalian tersebut akan dihitung kembali dengan nilai α terakhir yang telah ditemukan dan dikalikan dengan setiap kelas pada dokumen data latih. Hasil akhir dari perhitungan tersebut akan dijumlahkan dengan nilai bias yang telah didapat. Jika hasil perhitungan lebih dari 0, maka data uji tersebut merupakan kelas positif. Namun jika hasilnya kurang dari 0, maka data uji tersebut termasuk pada kelas negatif. Alur proses pada tahapan ini ditunjukkan pada Gambar 4.22.

Perhitungan testing





Gambar 0.22 Alur Proses Perhitungan Testing

Tahapan proses perhitungan data testing ditunjukkan pada Gambar 4.22 yang dapat diuraikan pada keterangan berikut:

1. Masukkan dataset yaitu berupa data uji, kelas uji, nilai *degree*, nilai alfa, dan hasil perhitungan bias.
2. Inialisasi nilai konstanta C yang digunakan untuk perhitungan kernel polynomial.
3. Perhitungan bersaranga hingga batas panjang data pada baris matriks data uji, panjang data kelas uji, dan panjang data perbaris matriks data uji.
4. Dalam perulangan dilakukan proses perkalian antara matriks data uji dengan matriks transposnya. Setiap hasil perkalian yang didapatkan disimpan dalam variabel total dan diakumulasikan.
5. Proses perkalian matriks dilakukan untuk perhitungan kernel polynomial. Perhitungan kernel polynomial ditunjukkan pada Persamaan 2.10.
6. Setiap hasil perkalian yang telah dilakukan antara data uji kemudian dijumlahkan dengan nilai bias yang telah ditemukan.
7. Hasil tes yang ditemukan pada setiap data uji kemudian diklasifikasikan dengan kondisi *if-else*. Jika hasil tes tersebut bernilai kurang dari 0, maka diklasifikasikan menjadi kelas negatif. Jika hasil tes bernilai lebih dari 0, maka hasil klasifikasinya adalah kelas positif.

8. *Output* yang dihasilkan adalah hasil matriks data uji dan hasil akhir kelas uji.

4.7 Manualisasi Metode Support Vector Machine

Tahapan manualisasi merupakan tahapan perhitungan pada data latih dan data uji secara manual. Dokumen teks yang didapatkan secara manual dari komentar-komentar di instagram akan diproses dari *Pre-processing* teks, pembobotan TF-IDF hingga perhitungan *Support Vector Machine*. Data latih yang digunakan telah diketahui kelas sentimennya, baik kelas sentimen negatif dan kelas sentimen positif. Namun data uji yang merupakan dokumen teks masih belum diketahui kelas sentimennya. Untuk mengetahui kelas sentimen dari data uji akan dilakukan klasifikasi dengan meted *Support Vector Machine*. Sampel data latih dan data uji yang digunakan pada tahap manualisasi dapat dilihat pada Tabel 4.1 dan 4.2.

Tabel 0.1 Data latih

No	Komentar	Sentimen	Kelas
1	Yang jd masalah disini adalah 1. mereka ciuman dgn org bnyk yang SEHARUSNYA sih punya rasa MALU, kecuali mgkn lg mabok jd gak ada rasa malu. 2. Mereka berteman?? Helooow sebuah pertemanan tdk akan melibatkan ciuman BIBIR, coba aja kalian bayangin, sanggup gak ciuman bibir ma TEMAN? pasti lbh dr teman lah itu. Dasar Kelakuan Kids jaman now yang emg bego, bukan krn kebanyakan micin lho yah 😊	Positif	1
2	Kalau ada tempat khusus buat undangan fans bunmai saat pernikahan..aku mau dink diundang yang VIP coz penggemar berat sejak masih ama pinkan. Nih cewek terkeren sepanjang masa bunda maia. Gak bosan blass 😊	Negatif	-1
3	@yusnitarinna918 iya betul mba. Ariel gak ganteng ariel punya kharisma yang menarik. Udah tua pun skrang agak keriput wajahnya ttp aja kharisma nya sebagai lelaki keliatan lebih terpancar. Buktinya mama, dan tante saya makin tergilas sama ariel dan saya pun mengakui kalau ariel itu menarik untuk di lihat 😊😊😊	Negatif	-1
4	banyak banget emasnya.jadi salpok diemasnya.tp sumpah lagunya dan music nya keren bngat.apalagi konsep mv nya ia ambil handel.kan keren menurut saya.kalo denada taro kutang diluar itu krn masih minim perestasi.tp kalo agnes itu hal wajar.prestasi dan dunia fashion kalo artis indonesia ia selalu unggul.krn ia berani meski gak cocok ia pake	Negatif	-1

Tabel 4.1 Data latih (Lanjutan)

No	Komentar	Sentimen	Kelas
5	krn sebagian besar rakyat indonesia itu bodoh. Tdk bisa membedakan mana yang baik mana yang buruk. Sdh tau prilaku buruk tdk tau sopan santun itu yang dipilih jd panutan. Lebih mengidolakan yang murahan dr pd yang berkelas dan ber etika saya pun mengakui kalau ariel itu menarik untuk di lihat 😊😊😊	Negatif	-1
6	Kereeen nih...enak bgt didengernya...gak bosan puter ulang2...daripada yang lipsing2 gajelas joget2 gtu...	Negatif	-1

Tabel 0.2 Data uji

No	Komentar	Sentimen
1	@fahrianalimbong TOLOL!! Gak ada hubungan nya keguguran dgn pake hijab syar'i yang lo bilang bayi nya kepanasan didalem gak ada hubungan nya woyyyy!! Otak sama jempol lo gak sinkron sih ya jadinya asal nulis komentar!	?
2	Gilaaaaa pelakor banyak bnget laki2 gatal juga banyak alasan untuk berpoligami woiii mikir pake otak haduh saya pikir dia org pintar tau agama mana baik mana buruk.... Allah menguji kesetiaan dan nafsunya eh malah kaya gtu... Menyakiti istri sama sja menutup pintu rejekimu... Anak istri kau korbakan! Sbg seorg imam keluarga dia telah gagal mmpertahankan rmh tangganya mencoreng nama baik nya sndiri... Smua org suka skrg benci sama dia.. M	?
3	RATU PELAKOR SEJAGAT RAYA, DIMANA2 LONTE YA GAK PUNYA MALU. YANG NGERASA LONTE DISINI TOLONG PUNYA MALU DIKIT	?

4.7.1 Manualisasi Case Folding

Tahapan awal dari proses manualisasi adalah proses manualisasi case folding terhadap data latih dan data uji. Proses ini adalah mengubah seluruh karakter pada data latih maupun data uji menjadi huruf kecil. Manualisasi *case folding* yang dilakukan pada data latih dan data uji diperlihatkan pada Tabel 4.3 dan 4.4.

Tabel 0.3 Data latih *proses case folding*

No	Data Asli	Data Setelah <i>Case folding</i>
1	Yang jd masalah disini adalah 1. mereka ciuman dgn org bnyk yang SEHARUSNYA sih punya rasa MALU, kecuali mgkn lg mabok jd gak ada rasa malu. 2. Mereka berteman?? Helooow sebuah pertemanan tdk akan melibatkan ciuman BIBIR, coba aja kalian bayangin, sanggup gak ciuman bibir ma TEMAN? pasti lbh dr teman lah itu. Dasar Kelakuan Kids jaman now yang emg bego, bukan krn kebanyakan micin lho yah 😊	yang jd masalah disini adalah 1. mereka ciuman dgn org bnyk yang seharusnya sih punya rasa malu, kecuali mgkn lg mabok jd gak ada rasa malu. 2. mereka berteman?? helooow sebuah pertemanan tdk akan melibatkan ciuman bibir, coba aja kalian bayangin, sanggup gak ciuman bibir ma teman? pasti lbh dr teman lah itu. dasar kelakuan kids jaman now yang emg bego, bukan krn kebanyakan micin lho yah ??
2	Kalau ada tempat khusus buat undangan fans bunmaï saat pernikahan..aku mau dink diundang yang VIP coz penggemar berat sejak masih ama pinkan. Nih cewek terkeren sepanjang masa bunda maia. Gak bosen blasss 😊	kalau ada tempat khusus buat undangan fans bunmai saat pernikahan..aku mau dink diundang yang vip coz penggemar berat sejak masih ama pinkan. nih cewek terkeren sepanjang masa bunda maia. gak bosen blasss ??
3	@yusnitarinna918 iya betul mba. Ariel gak ganteng ariel punya kharisma yang menarik. Udah tua pun skrang agak keriput wajahnya ttp aja kharisma nya sebagai lelaki keliatan lebih terpancar. Buktinya mama, dan tante saya makin tergilal2 sama ariel dan sa	@yusnitarinna918\xa0iya betul mba. ariel gak ganteng ariel punya kharisma yang menarik. udah tua pun skrang agak keriput wajahnya ttp aja kharisma nya sebagai lelaki keliatan lebih terpancar. buktinya mama, dan tante saya makin tergilal2 sama ariel dan sa
4	banyak banget emasnya.jadi salpok diemasnya.tp sumpah lagunya dan music nya keren bngat.apalagi konsep mv nya ia ambil handel.kan keren menurut saya.kalo denada taro kutang diluar itu krn masih minim perestasi.tp kalo agnes itu hal wajar.prestasi dan dunia fashion kalo artis indonesia ia selalu unggul.krnl ia berani meski gak cocok ia pake	banyak banget emasnya.jadi salpok diemasnya.tp sumpah lagunya dan music nya keren bngat.apalagi konsep mv nya ia ambil handel.kan keren menurut saya.kalo denada taro kutang diluar itu krn masih minim perestasi.tp kalo agnes itu hal wajar.prestasi dan dunia fashion kalo artis indonesia ia selalu unggul.krnl ia berani meski gak cocok ia pake

Tabel 4.3 Data latih proses case folding (Lanjutan)

No	Data Asli	Data Setelah Case folding
5	krn sebagian besar rakyat indonesia itu bodoh. Tdk bisa membedakan mana yang baik mana yang buruk. Sdh tau prilaku buruk tdk tau sopan santun itu yang dipilih jd panutan. Lebih mengidolakan yang murahan dr pd yang berkelas dan ber etika	krn sebagian besar rakyat indonesia itu bodoh. tdk bisa membedakan mana yang baik mana yang buruk. sdh tau prilaku buruk tdk tau sopan santun itu yang dipilih jd panutan. lebih mengidolakan yang murahan dr pd yang berkelas dan ber etika
6	Kereeen nih...enak bgt didengernya...gak bosen puter ulang2...daripada yang lipsing2 gajelas joget2 gtu...	kereeen nih...enak bgt didengernya...gak bosen puter ulang2...daripada yang lipsing2 gajelas joget2 gtu...

Tabel 0.4 Data uji proses case folding

No	Data Asli	Data Setelah Case Folding
1	@fahrianalimbong TOLOL!! Gak ada hubungan nya keguguran dgn pake hijab syar'i yang lo bilang bayi nya kepanasan didalem gak ada hubungan nya woyyyy!! Otak sama jempol lo gak sinkron sih ya jadinya asal nulis komentar!	@fahrianalimbong tolol!! gak ada hubungan nya keguguran dgn pake hijab syar'i yang lo bilang bayi nya kepanasan didalem gak ada hubungan nya woyyyy!! otak sama jempol lo gak sinkron sih ya jadinya asal nulis komentar!
2	Gilaaaaa pelakor banyak bnget laki2 gatal juga banyak alasan untuk berpoligami woiii mikir pake otak haduh saya pikir dia org pintar tau agama mana baik mana buruk.... Allah menguji kesetiaan dan nafsunya eh malah kaya gtu... Menyakiti istri sama sja menutup pintu rejekimu... Anak istri kau korbakan! Sbg seorg imam keluarga dia telah gagal mmpertahankan rmh tangganya mencoreng nama baik nya sndiri... Smua org suka skrg benci sama dia.. M	gilaaaaa pelakor banyak bnget laki2 gatal juga banyak alasan untuk berpoligami woiii mikir pake otak haduh saya pikir dia org pintar tau agama mana baik mana buruk.... allah menguji kesetiaan dan nafsunya eh malah kaya gtu... menyakiti istri sama sja menutup pintu rejekimu... anak istri kau korbakan! sbg seorg imam keluarga dia telah gagal mmpertahankan rmh tangganya mencoreng nama baik nya sndiri... smua org suka skrg benci sama dia.. m
3	RATU PELAKOR SEJAGAT RAYA, DIMANA2 LONTE YA GAK PUNYA MALU. YANG NGERASA LONTE DISINI TOLONG PUNYA MALU DIKIT	ratu pelakor sejagat raya, dimana2 lonte ya gak punya malu. yang ngerasa lonte disini tolong punya malu dikit



4.7.2 Manualisasi Data Cleaning

Tahapan yang dilalui setelah *case folding* adalah *data cleaning*. Proses manualisasi pada tahapan ini adalah menghapus setiap karakter kata berupa *username, link url, hashtag*. Hasil manualisasi tahapan *data cleaning* pada data latih dan data uji dapat dilihat pada Tabel 4.5 dan 4.6.

Tabel 0.5 Data latih proses *data cleaning*

No	Data Asli	Data Setelah <i>Cleansing</i>
1	yg jd masalah disini adalah 1. mereka ciuman dgn org bnyk yg seharusnya sih punya rasa malu, kecuali mgkn lg mabok jd gak ada rasa malu. 2. mereka berteman?? helooow sebuah pertemanan tdk akan melibatkan ciuman bibir, coba aja kalian bayangin, sanggup gak ciuman bibir ma teman? pasti lbh dr teman lah itu. dasar kelakuan kids jaman now yg emg bego, bukan krn kebanyakan micin lho yah ??	yang jd masalah disini adalah mereka ciuman dgn org bnyk yang seharusnya sih punya rasa malu kecuali mgkn lg mabok jd gak ada rasa malu mereka berteman helooow sebuah pertemanan tdk akan melibatkan ciuman bibir coba aja kalian bayangin sanggup gak ciuman bibir ma teman pasti lbh dr teman lah itu dasar kelakuan kids jaman now yang emg bego bukan krn kebanyakan micin lho yah
2	kalau ada tempat khusus buat undangan fans bunmai saat pernikahan..aku mau dink diundang yg vip coz penggemar berat sejak masih ama pinkan. nih cewek terkeren sepanjang masa bunda maia. gak bosan blsss ??	kalau ada tempat khusus buat undangan fans bunmai saat pernikahan aku mau dink diundang yang vip coz penggemar berat sejak masih ama pinkan nih cewek terkeren sepanjang masa bunda maia gak bosan blsss
3	@yusnitarinna918\xa0iya betul mba. ariel gak ganteng ariel punya kharisma yag menarik. udah tua pun skrang agak keriput wajahnya ttp aja kharisma nya sebagai lelaki keliatan lebih terpancar. buktinya mama, dan tante saya makin tergila2 sama ariel dan sa	iya betul mba ariel gak ganteng tapi ariel punya kharisma yang menarik udah tua pun skrang agak keriput wajahnya ttp aja kharisma nya sebagai lelaki keliatan lebih terpancar buktinya mama dan tante saya makin tergila sama ariel dan sa
4	banyak banget emasnya.jadi salpok diemasnya.tp sumpah lagunya dan music nya keren bngat.apalagi konsep mv nya ia ambil handel.kan keren menurut saya.kalo denada taro kutang diluar itu krn masih minim perestasi.tp kalo agnes itu hal wajar.prestasi dan dunia fashion kalo artis indonesia ia	banyak banget emasnya jadi salpok diemasnya tp sumpah lagunya dan music nya keren bngat apalagi konsep mv nya ia ambil handel kan keren menurut saya kalo denada taro kutang diluar itu krn masih minim perestasi tp kalo agnes itu hal wajar prestasi dan dunia fashion kalo artis indonesia ia selalu unggul



Tabel 4.5 Data latih proses *data cleaning* (Lanjutan)

No	Data Asli	Data Setelah <i>Cleansing</i>
	selalu unggul.krn ia berani meski gak cocok ia pake	krn ia berani meski gak cocok ia pake
5	krn sebagian besar rakyat indonesia itu bodoh. tdk bisa membedakan mana yg baik mana yg buruk. sdh tau prilaku buruk tdk tau sopan santun itu yg dipilih jd panutan. lebih mengidolakan yg murahan dr pd yg berkelas dan ber etika	krn sebagian besar rakyat indonesia itu bodoh tdk bisa membedakan mana yang baik mana yang buruk sdh tau prilaku buruk tdk tau sopan santun itu yang dipilih jd panutan lebih mengidolakan yang murahan dr pd yang berkelas dan ber etika
6	kereeen nih...enak bgt didengernya...gak bosan puter ulang2...daripada yg lipsing2 gajelas joget2 gtu...	kereeen nih enak bgt didengernya gak bosan puter ulang daripada yang lipsing gajelas joget gtu

Tabel 0.6 Data uji proses *data cleaning*

No	Data Asli	Data Setelah <i>Cleansing</i>
1	@fahrianalimbong tolo!! gak ada hubungan nya keguguran dgn pake hijab syar'i yg lo bilang bayi nya kepanasan didalem gak ada hubungan nya woyyyy!! otak sama jempol lo gak singkron sih ya jadinya asal nulis komentar!	tolol gak ada hubungan nya keguguran dgn pake hijab syar'i yang lo bilang bayi nya kepanasan didalem gak ada hubungan nya woyyyy otak sama jempol lo gak singkron sih ya jadinya asal nulis komentar
2	gilaaaaa pelakor banyak bnget laki2 gatal juga banyak alasan untuk berpoligami woiii mikir pake otak haduh saya pikir dia org pintar tau agama mana baik mana buruk.... allah menguji kesetiaan dan nafsunya eh malah kaya gtu... menyakiti istri sama sja menutup pintu rejekimu... anak istri kau korbakan! sbg seorg imam keluarga dia telah gagal mmpertahankan rmh tangganya mencoreng nama baik nya sndiri... smua org suka skrg benci sama dia.. m	gilaaaaa pelakor banyak bnget laki gatal juga banyak alasan untuk berpoligami woiii mikir pake otak haduh saya pikir dia org pintar tau agama mana baik mana buruk allah menguji kesetiaan dan nafsunya eh malah kaya gtu menyakiti istri sama sja menutup pintu rejekimu anak istri kau korbakan sbg seorg imam keluarga dia telah gagal mmpertahankan rmh tangganya mencoreng nama baik nya sndiri smua org suka skrg benci sama dia m'



Tabel 4.6 Data uji proses *data cleaning* (Lanjutan)

No	Data Asli	Data Setelah Cleansing
3	ratu pelakor sejagat raya, dimana2 lonte ya gak punya malu. yg ngerasa lonte disini tolong punya malu dikit	ratu pelakor sejagat raya dimana lonte ya gak punya malu yang ngerasa lonte disini tolong punya malu dikit

4.7.3 Manualisasi Normalisasi Bahasa

Manualisasi yang dilakukan setelah tokenisasi adalah normalisasi bahasa. Setiap kata yang tidak baku atau *alay* akan diproses pada tahapan ini. Kata-kata pada setiap dokumen data latih dan data uji diperiksa pada kamus yang berisi daftar kata tidak baku. Jika terdapat kata tidak baku, maka kata tersebut akan digantikan dengan kata baku. Hasil manualisasi dari normalisasi bahasa baku dan tidak baku pada data latih dan data uji ditunjukkan pada Tabel 4.7 dan 4.8.

Tabel 0.7 Data Latih Proses Normalisasi Bahasa

No	Data Asli	Data Setelah Normalisasi
1	yang jd masalah disini adalah mereka ciuman dpn org bnyk yang seharusnya sih punya rasa malu kecuali mgkn lg mabok jd gak ada rasa malu mereka berteman helooow sebuah pertemanan tdk akan melibatkan ciuman bibir coba aja kalian bayangin sanggup gak ciuman bibir ma teman pasti lbh dr teman lah itu dasar kelakuan kids jaman now yang emg bego bukan krn kebanyakan micin lho yah	yang jadi masalah disini adalah mereka ciuman depan orang banyak yang seharusnya sih punya rasa malu kecuali mungkin lagi mabuk jadi tidak ada rasa malu mereka berteman halo sebuah pertemanan tidak akan melibatkan ciuman bibir coba saja kalian bayangin sanggup tidak ciuman bibir sama teman pasti lebih dari teman lah itu dasar kelakuan anak zaman sekarang yang memang bego bukan karena banyak micin lo ya
2	kalau ada tempat khusus buat undangan fans bunmai saat pernikahan aku mau dink diundang yang vip coz penggemar berat sejak masih ama pinkan nih cewek terkeren sepanjang masa bunda maia gak bosen blass	kalau ada tempat khusus untuk undangan penggemar bunmai saat pernikahan saya mau ding diundang yang vip karena penggemar berat sejak masih sama pinkan nih cewek terkeren sepanjang masa bunda maia tidak bosan sekali



Tabel 4.7 Data Latih Proses Normalisasi Bahasa (Lanjutan)

No	Data Asli	Data Setelah Normalisasi
3	iya betul mba ariel gak ganteng tapi ariel punya kharisma yang menarik udah tua pun skrang agak keriput wajahnya ttp aja kharisma nya sebagai lelaki keliatan lebih terpancar buktinya mama dan tante saya makin tergilgila sama ariel dan sa	iya betul mbak ariel tidak ganteng tapi ariel punya kharisma yang menarik sudah tua pun sekarang agak keriput wajahnya tetap saja kharisma nya sebagai lelaki keliatan lebih terpancar buktinya mama dan tante saya semakin tergilgila sama ariel dan sa
4	banyak banget emasnya jadi salpok diemasnya tp sumpah lagunya dan music nya keren bngat apalagi konsep mv nya ia ambil handel kan keren menurut saya kalo denada taro kutang diluar itu krn masih minim perestasi tp kalo agnes itu hal wajar prestasi dan dunia fashion kalo artis indonesia ia selalu unggul krn ia berani meski gak cocok ia pake	banyak banget emasnya jadi salah fokus diemasnya tapi sumpah lagunya dan musik nya keren banget apalagi konsep mv nya ia ambil handal kan keren menurut saya kalau denada taruh kutang diluar itu karena masih minim prestasi tapi kalau agnes itu hal wajar prestasi dan dunia fashion kalau artis indonesia ia selalu unggul karena ia berani meski tidak cocok ia pakai
5	krn sebagian besar rakyat indonesia itu bodoh tdk bisa membedakan mana yang baik mana yang buruk sdh tau prilaku buruk tdk tau sopan santun itu yang dipilih jd panutan lebih mengidolakan yang murahan dr pd yang berkelas dan ber etika	karena sebagian besar rakyat indonesia itu bodoh tidak bisa membedakan mana yang baik mana yang buruk sudah tahu perilaku buruk tidak tahu sopan santun itu yang dipilih jadi panutan lebih mengidolakan yang murahan dari pada yang berkelas dan ber etika
6	kereeen nih enak bgt didengernya gak bosen puter ulang daripada yang lipsing gajelas joget gitu	keren nih enak banget didengarnya tidak bosan putar ulang daripada yang lipsing tidak jelas joget gitu



Tabel 0.8 Data Uji Proses Normalisasi Bahasa

No	Data Asli	Data Setelah Normalisasi
1	tolol gak ada hubungan nya keguguran dgn pake hijab syar'i yang lo bilang bayi nya kepanasan didalem gak ada hubungan nya woyyyyy otak sama jempol lo gak singkron sih ya jadinya asal nulis komentar	tolol tidak ada hubungan nya keguguran dengan pakai hijab syar'i yang kamu bilang bayi nya kepanasan didalam tidak ada hubungan nya woi otak sama jempol kamu tidak singkron sih ya jadinya asal tulis komentar
2	gilaaaaa pelakor banyak bnget laki gatal juga banyak alasan untuk berpoligami woiii mikir pake otak haduh saya pikir dia org pintar tau agama mana baik mana buruk allah menguji kesetiaan dan nafsunya eh malah kaya gtu menyakiti istri sama sjā menutup pintu rejekimu anak istri kau korbankan sbg seorg imam keluarga dia telah gagal mmpertahankan rmh tangganya mencoreng nama baik nya sndiri smua org suka skrg benci sama dia m'	gila pelakor banyak banget laki gatal juga banyak alasan untuk berpoligami woiii mikir pakai otak haduh saya pikir dia orang pintar tahu agama mana baik mana buruk allah menguji kesetiaan dan nafsunya eh malah kaya gitu menyakiti istri sama saja menutup pintu rezekimu anak istri kamu korbankan sebagai seorang imam keluarga dia telah gagal mempertahankan rumah tangganya mencoreng nama baik nya sendiri semua orang suka sekarang benci sama dia m
3	ratu pelakor sejagat raya dimana lonte ya gak punya malu yang ngerasa lonte disini tolong punya malu dikit	ratu pelakor sejagat raya dimana lonte ya tidak punya malu yang ngerasa lonte disini tolong punya malu dikit

4.7.4 Manualisasi *Stopword Removal*

Subproses yang selanjutnya dijalankan adalah *stopword removal*. Pada manualisasi tahap *stopword removal* yaitu memfilterisasi setiap kata yang telah ditokenisasi. Proses filterisasi ini yaitu memilih kata pada dokumen data latih dan data uji yang cocok dengan *stoplist* yang telah dimasukkan. Jika terdapat kata yang cocok pada *stoplist*, maka kata tersebut akan dihapus dari dokumen karena dianggap sebagai kata yang tidak memiliki makna yang penting. Manualisasi ditunjukkan pada Tabel 4.9 dan 4.10.

Tabel 0.9 Data latihan proses *stopword removal*

No	Data Asli	Data Setelah <i>Stopword Removal</i>
1	<p>yang jadi masalah disini adalah mereka ciuman depan orang banyak yang seharusnya sih punya rasa malu kecuali mungkin lagi mabuk jadi tidak ada rasa malu mereka berteman halo sebuah pertemanan tidak akan melibatkan ciuman bibir coba saja kalian bayangin sanggup tidak ciuman bibir sama teman pasti lebih dari teman lah itu dasar kelakuan anak zaman sekarang yang memang bego bukan karena banyak micin lo ya</p>	<p>ciuman orang malu kecuali mabuk malu berteman pertemanan melibatkan ciuman bibir coba bayangin sanggup ciuman bibir teman teman dasar kelakuan anak zaman bego micin</p>
2	<p>kalau ada tempat khusus untuk undangan penggemar bunmai saat pernikahan saya mau ding diundang yang vip karena penggemar berat sejak masih sama pinkan nih cewek terkeren sepanjang masa bunda maia tidak bosan sekali</p>	<p>khusus undangan penggemar bunmai pernikahan ding diundang vip penggemar berat pinkan cewek terkeren maia bosan</p>
3	<p>iya betul mbak ariel tidak ganteng tapi ariel punya kharisma yang menarik sudah tua pun sekarang agak keriput wajahnya tetap saja kharisma nya sebagai lelaki kelihatan lebih terpancar buktinya mama dan tante saya semakin tergila sama ariel dan sa</p>	<p>iya ariel ganteng ariel kharisma menarik tua keriput wajahnya kharisma lelaki kelihatan terpancar buktinya tergila ariel sa</p>
4	<p>banyak banget emasnya jadi salah fokus diemasnya tapi sumpah lagunya dan musik nya keren banget apalagi konsep mv nya ia ambil handal kan keren menurut saya kalau denada taruh kutang diluar itu karena masih minim prestasi tapi kalau agnes itu hal wajar prestasi dan dunia fashion kalau artis indonesia ia selalu unggul karena ia berani meski tidak cocok ia pakai</p>	<p>emasnya salah fokus diemasnya sumpah lagunya musik keren konsep mv ambil handal keren denada taruh kutang diluar minim prestasi agnes wajar prestasi dunia fashion artis indonesia unggul berani cocok pakai</p>



Tabel 4.9 Data latih proses *stopword removal* (Lanjutan)

No	Data Asli	Data Setelah <i>Stopword Removal</i>
5	karena sebagian besar rakyat indonesia itu bodoh tidak bisa membedakan mana yang baik mana yang buruk sudah tahu perilaku buruk tidak tahu sopan santun itu yang dipilih jadi panutan lebih mengidolakan yang murahan dari pada yang berkelas dan ber etika	rakyat indonesia bodoh membedakan buruk perilaku buruk sopan santun dipilih panutan mengidolakan murahan berkelas ber etika
6	keren nih enak banget didengarnya tidak bosan putar ulang daripada yang lipsing tidak jelas joget gitu	keren enak didengarnya bosan putar ulang lipsing joget gitu

Tabel 0.10 Data uji proses *stopword removal*

No	Data Asli	Data Setelah <i>Stopword Removal</i>
1	tolol tidak ada hubungan nya keguguran dengan pakai hijab syar'i yang kamu bilang bayi nya kepanasan didalam tidak ada hubungan nya woi otak sama jempol kamu tidak sinkron sih ya jadinya asal tulis komentar	tolol hubungan keguguran pakai hijab syar'i bilang bayi kepanasan didalam hubungan otak jempol sinkron tulis komentar
2	gila pelakor banyak banget laki gatal juga banyak alasan untuk berpoligami woiiii mikir pakai otak haduh saya pikir dia orang pintar tahu agama mana baik mana buruk allah menguji kesetiaan dan nafsunya eh malah kaya gitu menyakiti istri sama saja menutup pintu rezekimu anak istri kamu korbakan sebagai seorang imam keluarga dia telah gagal mempertahankan rumah tangganya mencoreng nama baik nya sendiri semua orang suka sekarang benci sama dia m	gila pelakor laki gatal alasan berpoligami mikir pakai otak pikir orang pintar agama buruk allah menguji kesetiaan nafsunya kaya gitu menyakiti istri menutup pintu rezekimu anak istri korbakan imam keluarga gagal mempertahankan rumah tangganya mencoreng nama orang suka benci

Tabel 4.10 Data uji proses *stopword removal* (Lanjutan)

No	Data Asli	Data Setelah <i>Stopword Removal</i>
3	ratu pelakor sejagat raya dimana lonte ya tidak punya malu yang ngerasa lonte disini tolong punya malu dikit	ratu pelakor sejagat raya dimana lonte malu ngerasa lonte tolong malu dikit

4.7.5 Manualisasi *Stemming*

Tahap akhir dari *processing* teks dilakukan dengan menghapus setiap imbuhan pada kata. Subproses ini diketahui sebagai proses *stemming*. *Stemming* dilakukan berdasar pada pendekatan aturan. Hasil manualisasi dari proses *stemming* ditunjukkan pada Tabel 4.11 dan 4.12.

Tabel 0.11 Data Latih Proses *Stemming*

No	Data Asli	Data Setelah <i>STEMMING</i>
1	ciuman orang malu kecuali mabuk malu berteman pertemanan melibatkan ciuman bibir coba bayangin sanggup ciuman bibir teman teman dasar kelakuan anak zaman bego micin	cium orang malu kecuali mabuk malu teman teman libat cium bibir coba bayangin sanggup cium bibir teman teman dasar laku anak zaman bego micin
2	khusus undangan penggemar bunmai pernikahan ding diundang vip penggemar berat pinkan cewek terkeren maia bosan	khusus undang gemar bunmai nikah ding undang vip gemar berat pin cewek keren maia bosan
3	iya ariel ganteng ariel kharisma menarik tua keriput wajahnya kharisma lelaki keliatan terpancar buktinya tergila ariel sa	iya ariel ganteng ariel kharisma tarik tua keriput wajah kharisma lelaki liat pancar bukti gila ariel sa
4	emasnya salah fokus diemasnya sumpah lagunya musik keren konsep mv ambil handal keren denada taruh kutang diluar minim prestasi agnes wajar prestasi dunia fashion artis indonesia unggul berani cocok pakai	emas salah fokus emas sumpah lagu musik keren konsep mv ambil handal keren denada taruh kutang luar minim prestasi agnes wajar prestasi dunia fashion artis indonesia unggul berani cocok pakai
5	rakyat indonesia bodoh membedakan buruk perilaku buruk sopan santun dipilih	rakyat indonesia bodoh beda buruk perilaku buruk sopan santun pilih panutan idola murah kelas ber etika

Tabel 4.11 Data Latih Proses Stemming (Lanjutan)

No	Data Asli	Data Setelah STEMMING
	panutan mengidolakan murahan berkelas ber etika	
6	keren enak didengarnya bosan putar ulang lipsing joget gitu	keren enak dengar bosan putar ulang lipsing joget gitu

Tabel 0.12 Data Uji Proses Stemming

No	Data Asli	Data Setelah Stemming
1	tolol hubungan keguguran pakai hijab syar'i bilang bayi kepanasan didalam hubungan otak jempol singkron tulis komentar	tolol hubung gugur pakai hijab syar i bilang bayi panas dalam hubung otak jempol singkron tulis komentar
2	gila pelakor laki gatal alasan berpoligami mikir pakai otak pikir orang pintar agama buruk allah menguji kesetiaan nafsunya kaya gitu menyakiti istri menutup pintu rezekimu anak istri korbakan imam keluarga gagal mempertahankan rumah tangganya mencoreng nama orang suka benci	gila pelakor laki gatal alas poligam mikir pakai otak pikir orang pintar agama buruk allah uji setia nafsu kaya gitu sakit istri tutup pintu rezeki anak istri korban imam keluarga gagal tahan rumah tangga coreng nama orang suka benci
3	ratu pelakor sejagat raya dimana lonte malu ngerasa lonte tolong malu dikit	ratu pelakor jagat raya mana lonte malu ngerasa lonte tolong malu dikit

4.7.6 Manualisasi Tokenisasi

Langkah ketiga dalam proses *Pre-processing* adalah tokenisasi (tokenisasi). Proses manualisasi yang dilakukan pada tahapan ini adalah memecah dokumen teks yang berupa kalimat pada data latih dan data uji menjadi per kata. Proses manualisasi tokenisasi pada data latih dan data uji diperlihatkan pada Tabel 4.13 dan 4.14.



Tabel 0.13 Data latihan proses tokenisasi

No	Data Asli	Data Setelah Tokenisasi
1	cium orang malu kecuali mabuk malu teman teman libat cium bibir coba bayangin sanggup cium bibir teman teman dasar laku anak zaman bego micin	'cium', 'orang', 'malu', 'kecuali', 'mabuk', 'malu', 'teman', 'teman', 'libat', 'cium', 'bibir', 'coba', 'bayangin', 'sanggup', 'cium', 'bibir', 'teman', 'teman', 'dasar', 'laku', 'anak', 'zaman', 'bego', 'micin'
2	khusus undang gemar bunmai nikah ding undang vip gemar berat pin cewek keren maia bosan	'khusus', 'undang', 'gemar', 'bunmai', 'nikah', 'ding', 'undang', 'vip', 'gemar', 'berat', 'pin', 'cewek', 'keren', 'maia', 'bosan'
3	iya ariel ganteng ariel kharisma tarik tua keriput wajah kharisma lelaki liat pancar bukti gila ariel sa	'iya', 'ariel', 'ganteng', 'ariel', 'kharisma', 'tarik', 'tua', 'keriput', 'wajah', 'kharisma', 'lelaki', 'liat', 'pancar', 'bukti', 'gila', 'ariel', 'sa'
4	emas salah fokus emas sumpah lagu musik keren konsep mv ambil handal keren denada taruh kutang luar minim prestasi agnes wajar prestasi dunia fashion artis indonesia unggul berani cocok pakai	'emas', 'salah', 'fokus', 'emas', 'sumpah', 'lagu', 'musik', 'keren', 'konsep', 'mv', 'ambil', 'handal', 'keren', 'denada', 'taruh', 'kutang', 'luar', 'minim', 'prestasi', 'agnes', 'wajar', 'prestasi', 'dunia', 'fashion', 'artis', 'indonesia', 'unggul', 'berani', 'cocok', 'pakai'
5	rakyat indonesia bodoh beda buruk perilaku buruk sopan santun pilih panutan idola murah kelas ber etika	'rakyat', 'indonesia', 'bodoh', 'beda', 'buruk', 'perilaku', 'buruk', 'sopan', 'santun', 'pilih', 'panutan', 'idola', 'murah', 'kelas', 'ber', 'etika'
6	keren enak dengar bosan putar ulang lipsing joget gitu	'keren', 'enak', 'dengar', 'bosan', 'putar', 'ulang', 'lipsing', 'joget', 'gitu'

Tabel 0.14 Data uji proses tokenisasi

No	Data Asli	Data Setelah Tokenisasi
1	tolol hubung gugur pakai hijab syar i bilang bayi panas dalam hubung otak jempol singkron tulis komentar	'tolol', 'hubung', 'gugur', 'pakai', 'hijab', 'syar', 'i', 'bilang', 'bayi', 'panas', 'dalam', 'hubung', 'otak', 'jempol', 'singkron', 'tulis', 'komentar'



Tabel 4.14 Data uji proses tokenisasi (Lanjutan)

No	Data Asli	Data Setelah Tokenisasi
2	gila pelakor laki gatal alas poligam mikir pakai otak pikir orang pintar agama buruk allah uji setia nafsu kaya gitu sakit istri tutup pintu rezeki anak istri korban imam keluarga gagal tahan rumah tangga coreng nama orang suka benci	'gila', 'pelakor', 'laki', 'gatal', 'alas', 'poligam', 'mikir', 'pakai', 'otak', 'pikir', 'orang', 'pintar', 'agama', 'buruk', 'allah', 'uji', 'setia', 'nafsu', 'kaya', 'gitu', 'sakit', 'istri', 'tutup', 'pintu', 'rezeki', 'anak', 'istri', 'korban', 'imam', 'keluarga', 'gagal', 'tahan', 'rumah', 'tangga', 'coreng', 'nama', 'orang', 'suka', 'benci'
3	ratu pelakor jagat raya mana lonte malu ngerasa lonte tolong malu dikit	'ratu', 'pelakor', 'jagat', 'raya', 'mana', 'lonte', 'malu', 'ngerasa', 'lonte', 'tolong', 'malu', 'dikit'

4.7.7 Manualisasi Perhitungan tf , $Wtf_{t,d}$, df , dan idf

Setelah *Pre-processing* teks telah selesai, maka tahapan selanjutnya adalah pembobotan dari setiap kata (*term*) disetiap dokumen dengan melakukan perhitungan nilai tf , $Wtf_{t,d}$, df , dan idf . Awalnya dicari nilai tf didapatkan dari frekuensi kata yang sama pada satu dokumen. Sedangkan nilai df didapatkan dari banyaknya dokumen yang mengandung kata (*term*). Proses perhitungan manual pada tf dan df ditunjukkan pada Tabel 4.15 dan untuk hasil manualisasi yang lengkap dapat dilihat pada Lampiran F.

Tabel 0.15 Perhitungan nilai tf dan df pada data latih dan data uji

Term	TF									df
	Data Latih						Data Uji			
	D1	D2	D3	D4	D5	D6	U1	U2	U3	
tolol	0	0	0	0	0	0	1	0	0	1
hubung	0	0	0	0	0	0	2	0	0	1
gugur	0	0	0	0	0	0	1	0	0	1
pakai	0	0	0	1	0	0	1	1	0	3
hijab	0	0	0	0	0	0	1	0	0	1
syar	0	0	0	0	0	0	1	0	0	1
l	0	0	0	0	0	0	1	0	0	1
bilang	0	0	0	0	0	0	1	0	0	1
bayi	0	0	0	0	0	0	1	0	0	1
panas	0	0	0	0	0	0	1	0	0	1
dalam	0	0	0	0	0	0	1	0	0	1
otak	0	0	0	0	0	0	1	1	0	2
jempol	0	0	0	0	0	0	1	0	0	1
singkron	0	0	0	0	0	0	1	0	0	1



Tabel 4.15 Perhitungan nilai *tf* dan *df* pada data latih dan data uji (Lanjutan)

Term	TF									df
	Data Latih						Data Uji			
	D1	D2	D3	D4	D5	D6	U1	U2	U3	
tulis	0	0	0	0	0	0	1	0	0	1
komentar	0	0	0	0	0	0	1	0	0	1
cium	3	0	0	0	0	0	0	0	0	1
orang	1	0	0	0	0	0	0	2	0	2
malu	2	0	0	0	0	0	0	0	2	2
kecuali	1	0	0	0	0	0	0	0	0	1
mabuk	1	0	0	0	0	0	0	0	0	1
teman	4	0	0	0	0	0	0	0	0	1
libat	1	0	0	0	0	0	0	0	0	1
bibir	2	0	0	0	0	0	0	0	0	1
coba	1	0	0	0	0	0	0	0	0	1
bayangin	1	0	0	0	0	0	0	0	0	1
sanggup	1	0	0	0	0	0	0	0	0	1
dasar	1	0	0	0	0	0	0	0	0	1
laku	1	0	0	0	0	0	0	0	0	1
.
.
.
joget	0	0	0	0	0	1	0	0	0	1

Dari nilai *tf* dan *df* yang didapat pada tabel diatas, selanjutnya akan dicari nilai $W_{tf,d}$ dan nilai *idf* pada setiap term. Untuk mencari $W_{tf,d}$ akan dilakukan perhitungan sesuai dengan rumus pada persamaan 2.1. Setelah nilai $W_{tf,d}$ ditemukan, akan dilakukan manualisasi pada nilai *idf* dengan rumus yang ditunjukkan pada persamaan 2.2. Nilai yang diberikan dari kedua proses manualisasi tersebut ditunjukkan pada Tabel 4.16 dan untuk hasil manualisasi yang lengkap dapat dilihat pada Lampiran G.

Tabel 0.16 Perhitungan nilai $W_{tf,d}$ dan *idf* pada data latih dan data uji

Term	$W_{tf,d}$									idf
	Data Latih						Data Uji			
	D1	D2	D3	D4	D5	D6	U1	U2	U3	
tolol	0	0	0	0	0	0	1	0	0	0.954
hubung	0	0	0	0	0	0	1.3	0	0	0.954
gugur	0	0	0	0	0	0	1	0	0	0.954
pakai	0	0	0	1	0	0	1	1	0	0.477
hijab	0	0	0	0	0	0	1	0	0	0.954



Tabel 4.15 Perhitungan nilai $W_{tf,t,d}$ dan idf pada data latih dan data uji (Lanjutan)

Term	$W_{tf,t,d}$									idf
	Data Latih						Data Uji			
	D1	D2	D3	D4	D5	D6	U1	U2	U3	
syar	0	0	0	0	0	0	1	0	0	0.954
i	0	0	0	0	0	0	1	0	0	0.954
bilang	0	0	0	0	0	0	1	0	0	0.954
bayi	0	0	0	0	0	0	1	0	0	0.954
panas	0	0	0	0	0	0	1	0	0	0.954
dalam	0	0	0	0	0	0	1	0	0	0.954
otak	0	0	0	0	0	0	1	1	0	0.653
jempol	0	0	0	0	0	0	1	0	0	0.954
singkron	0	0	0	0	0	0	1	0	0	0.954
tulis	0	0	0	0	0	0	1	0	0	0.954
komentar	0	0	0	0	0	0	1	0	0	0.954
cium	1.48	0	0	0	0	0	0	0	0	0.954
orang	1	0	0	0	0	0	0	1.3	0	0.653
malu	1.3	0	0	0	0	0	0	0	1.3	0.653
kecuali	1	0	0	0	0	0	0	0	0	0.954
mabuk	1	0	0	0	0	0	0	0	0	0.954
teman	1.6	0	0	0	0	0	0	0	0	0.954
libat	1	0	0	0	0	0	0	0	0	0.954
bibir	1.3	0	0	0	0	0	0	0	0	0.954
coba	1	0	0	0	0	0	0	0	0	0.954
bayangin	1	0	0	0	0	0	0	0	0	0.954
sanggup	1	0	0	0	0	0	0	0	0	0.954
dasar	1	0	0	0	0	0	0	0	0	0.954
laku	1	0	0	0	0	0	0	0	0	0.954
.
.
.
joget	0	0	0	0	0	1	0	0	0	0.954

Contoh perhitungan untuk mengetahui nilai $W_{tf,t,d}$ pada term pertama untuk data uji kesatu, dapat dijabarkan sebagai berikut:

$$t(\text{term}) = 1$$

$$d(\text{dokumen}) = 7 (U1)$$

$$W_{tf,t,d} = 1 + \log_{10} (1)$$

$$W_{tf,t,d} = 1$$



Contoh perhitungan manual yang dilakukan pada nilai *idf* pada term pertama, dapat dijabarkan sebagai berikut:

$$t(\text{term}) = 1$$

$$idf_t = \log_{10} \frac{9}{1}$$

$$idf_t = 0,954$$

4.7.8 Manualisasi Perhitungan TF-IDF

Untuk menyelesaikan pembobotan kata pada data latih dan data uji, maka proses selanjutnya adalah menghitung nilai TF-IDF. Rumus persamaan untuk mendapatkan hasil pembobotan kata pada tiap kata (*term*) disetiap dokumen dapat diketahui seperti persamaan 2.3. Hasil dari proses manualisasi perhitungan TF-IDF ditunjukkan pada Tabel 4.17 dan untuk hasil manualisasi yang lengkap dapat dilihat pada Lampiran H.

Tabel 0.17 Perhitungan TF-IDF pada data latih dan data uji

TF-IDF									
Term	Data Latih						Data Uji		
	D1	D2	D3	D4	D5	D6	U1	U2	U3
tolol	0	0	0	0	0	0	0.954	0	0
hubung	0	0	0	0	0	0	1.24	0	0
gugur	0	0	0	0	0	0	0.954	0	0
pakai	0	0	0	0.477	0	0	0.477	0.477	0
hijab	0	0	0	0	0	0	0.954	0	0
syar	0	0	0	0	0	0	0.954	0	0
i	0	0	0	0	0	0	0.954	0	0
bilang	0	0	0	0	0	0	0.954	0	0
bayi	0	0	0	0	0	0	0.954	0	0
panas	0	0	0	0	0	0	0.954	0	0
dalam	0	0	0	0	0	0	0.954	0	0
otak	0	0	0	0	0	0	0.653	0.653	0
jempol	0	0	0	0	0	0	0.954	0	0
singkron	0	0	0	0	0	0	0.954	0	0
tulis	0	0	0	0	0	0	0.954	0	0
komentar	0	0	0	0	0	0	0.954	0	0
cium	1.412	0	0	0	0	0	0	0	0
orang	0.653	0	0	0	0	0	0	0.849	0
malu	0.849	0	0	0	0	0	0	0	0.849
kecuali	0.954	0	0	0	0	0	0	0	0
mabuk	0.954	0	0	0	0	0	0	0	0
teman	1.526	0	0	0	0	0	0	0	0
libat	0.954	0	0	0	0	0	0	0	0
bibir	1.24	0	0	0	0	0	0	0	0

Tabel 4.15 Perhitungan TF-IDF pada data latih dan data uji (Lanjutan)

coba	0.954	0	0	0	0	0	0	0	0
bayangin	0.954	0	0	0	0	0	0	0	0
sanggup	0.954	0	0	0	0	0	0	0	0
dasar	0.954	0	0	0	0	0	0	0	0
laku	0.954	0	0	0	0	0	0	0	0
.
.
.
joget	0	0	0	0	0	0.954	0	0	0

Perhitungan manualisasi nilai TF-IDF pada term pertama untuk data uji kesatu dijelaskan sebagai berikut:

$$t(\text{term}) = 1$$

$$d(\text{dokumen}) = 7 (U1)$$

$$W_{t,d} = W_{tf,t,d} \times idf_t$$

$$W_{t,d} = 1 \times 0,954$$

$$W_{t,d} = 0,954$$

4.7.9 Manualisasi Perhitungan *Lexicon Based Features* dengan Normalisasi *Min-Max*

Proses klasifikasi teks tidak hanya dapat dilakukan dengan pembobotan TF-IDF saja. Setiap kata yang mengandung sentimen dapat dibobotkan pula dengan menggunakan pembobotan lexicon. Kata yang menjadi data latih maupun data uji dapat diketahui sentimennya berdasarkan kamus sentimen yang ada. Proses pembobotan sentimen dilakukan dengan mencocokkan kata dengan kamus sentimen untuk diketahui jenis sentimennya. Dari kata yang telah diketahui sentimennya akan dijumlahkan berdasarkan dokumen. Proses dalam menghitung sentimen ditunjukkan pada manualisasi pada Tabel 4.18.

Tabel 0.18 Pembobotan *Lexicon* pada data latih dan data uji

Dokumen	Jumlah Kata Sentimen	
	Positif	Negatif
D1	1	1
D2	4	2
D3	2	1
D4	7	1
D5	1	3
D6	2	2
U1	0	3



Tabel 4.18 Pembobotan *Lexicon* pada data latih dan data uji (Lanjutan)

U2	6	9
U3	1	3

Dari hasil pembobotan *Lexicon* yang telah didapatkan terdapat kesenjangan nilai antar data sehingga diperlukan proses normalisasi data. Proses normalisasi dilakukan agar data dapat direpresentasikan dengan baik. Normalisasi yang dilakukan pada pembobotan *Lexicon* adalah proses normalisasi min-max. Tahapan manualisasi diuraikan sebagai berikut:

$$v'_i = \frac{v_i - \min_a}{\max_a - \min_a} (\text{newmax} - \text{newmin}) + \text{newmin}$$

$$v'_i = \frac{4 - 2}{4 - 2} (0.9 - (0.1)) + (0.1)$$

$$v'_i = \frac{2}{2} 2 - 0.9$$

$$v'_i = 1 * 0.9$$

$$v'_i = 0.9$$

Berdasarkan contoh manualisasi dalam menormalisasikan bobot *Lexicon* diatas, didapatkan hasil normalisasi bobot yang ditunjukkan pada Tabel 4.19.

Tabel 0.19 Normalisasi pembobotan *Lexicon*

Dokumen	Jumlah Kata Sentimen		Normalisasi	
	Positif	Negatif	Positif	Negatif
D1	1	1	0.1	0.1
D2	4	2	0.9	0.1
D3	2	1	0.9	0.1
D4	7	1	0.9	0.1
D5	1	3	0.1	0.9
D6	2	2	0.1	0.1
U1	0	3	0.1	0.9
U2	6	9	0.1	0.9
U3	1	3	0.1	0.9

4.7.10 Manualisasi Perhitungan Klasifikasi *Support Vector Machine*

Pada tahapan ini akan dimulai proses klasifikasi dokumen menggunakan metode *Support Vector Machine*. Untuk memulai proses klasifikasi data yang diambil merupakan data dari hasil proses pembobotan TF-IDF dan pembobotan *Lexicon* yang telah dilakukan sebelumnya. Dalam metode *Support Vector Machine* diperlukan suatu fitur untuk proses datanya. Fitur-fitur tersebut merupakan kata



atau term yang ada pada setiap dokumen data latih maupun data uji. Data yang digunakan untuk memulai proses klasifikasi dapat dilihat pada Tabel 4.20 dan untuk hasil manualisasi yang lengkap dapat dilihat pada Lampiran I.

Tabel 0.20 Fitur *term* hasil perhitungan pembobotan TF-IDF dan pembobotan Lexicon yang diproses dengan SVM

Term	TF-IDF								
	Data Latih						Data Uji		
	D1	D2	D3	D4	D5	D6	U1	U2	U3
tolol	0	0	0	0	0	0	0.954	0	0
hubung	0	0	0	0	0	0	1.24	0	0
gugur	0	0	0	0	0	0	0.954	0	0
pakai	0	0	0	0.477	0	0	0.477	0.477	0
hijab	0	0	0	0	0	0	0.954	0	0
syar	0	0	0	0	0	0	0.954	0	0
i	0	0	0	0	0	0	0.954	0	0
bilang	0	0	0	0	0	0	0.954	0	0
bayi	0	0	0	0	0	0	0.954	0	0
panas	0	0	0	0	0	0	0.954	0	0
dalam	0	0	0	0	0	0	0.954	0	0
otak	0	0	0	0	0	0	0.653	0.653	0
jempol	0	0	0	0	0	0	0.954	0	0
singkron	0	0	0	0	0	0	0.954	0	0
tulis	0	0	0	0	0	0	0.954	0	0
komentar	0	0	0	0	0	0	0.954	0	0
cium	1.412	0	0	0	0	0	0	0	0
orang	0.653	0	0	0	0	0	0	0.849	0
malu	0.849	0	0	0	0	0	0	0	0.849
kecuali	0.954	0	0	0	0	0	0	0	0
mabuk	0.954	0	0	0	0	0	0	0	0
teman	1.526	0	0	0	0	0	0	0	0
libat	0.954	0	0	0	0	0	0	0	0
bibir	1.24	0	0	0	0	0	0	0	0
coba	0.954	0	0	0	0	0	0	0	0
bayangin	0.954	0	0	0	0	0	0	0	0
sanggup	0.954	0	0	0	0	0	0	0	0
dasar	0.954	0	0	0	0	0	0	0	0
laku	0.954	0	0	0	0	0	0	0	0
.
.
.
Lex Positif	0.1	0.9	0.9	0.9	0.1	0.1	0.1	0.1	0.1
Lex Negatif	0.1	0.1	0.1	0.1	0.9	0.1	0.9	0.9	0.9



Tahapan awal yang dilakukan pada proses klasifikasi pada metode *Support Vector Machine* adalah mencari pembatas antar kelas data (*hyperplane*) dengan menggunakan *sequential training SVM*. Untuk mencari *hyperplane* data yang dibutuhkan hanyalah data latih yang telah diketahui kelas sentimennya. Tahap dari *sequential learning* yang akan dilakukan adalah sebagai berikut:

1. Menginisialisasi setiap variabel yang dibutuhkan untuk perhitungan SVM. Variable tersebut adalah a_i , λ , γ , C , ϵ , d , dan iterasi maksimum (i_{max}). Inisialisasi nilai awal pada setiap variabel adalah:

$$\begin{aligned}
 a_i &= 0 \\
 \lambda &= 0.5 \\
 \gamma &= 0.0001 \\
 C &= 1 \\
 \epsilon &= 0.0001 \\
 i_{max} &= 3 \\
 d &= 2
 \end{aligned}$$

2. Menghitung matriks hessian D_{ij} dapat dilakukan sesuai dengan rumus yang ditunjukkan pada persamaan 2.12. Kernel yang digunakan adalah jenis kernel polynomial dengan memperhitungkan nilai *degree*. Nilai *degree* yang dimasukkan pada kernel adalah 2. Dari perkalian kernel yang dilakukan dengan rumus $K(x_i, x_d) = (X_i^T X_j + C)^d$. Contoh proses perhitungan kernel polynomial pada dokumen ke-1 dapat dijabarkan sebagai berikut:

$$\begin{aligned}
 K(x_i, x_d) &= (X_i^T X_j + C)^d \\
 K(x_i, x_d) &= ((0 \cdot 0) + \dots + (1,412 \cdot 1,412) + (0,653 \cdot 0,653) \\
 &\quad + (0,849 \cdot 0,849) + \dots + (0,1 \cdot 0,1) + (0,1 \cdot 0,1)) + 1^2 \\
 K(x_i, x_d) &= 340,953
 \end{aligned}$$

Hasil perhitungan kernel polynomial pada seluruh data latih ditunjukkan pada Tabel 4.21.

Tabel 0.21 Hasil perhitungan kernel pada data latih

K(xi,xi)	D1	D2	D3	D4	D5	D6
D1	340.953	1.210	1.210	1.210	1.210	1.040
D2	1.210	188.793	3.312	4.476	1.392	3.076
D3	1.210	3.312	221.382	3.312	1.392	1.210
D4	1.210	4.476	3.312	673.719	2.581	1.948
D5	1.210	1.392	1.392	2.581	219.002	1.210
D6	1.040	3.076	1.210	1.948	1.210	57.169

Kemudian contoh hasil perhitungan matriks Hessian pada dokumen ke-1 dapat dijabarkan sebagai berikut:

$$\begin{aligned}
 D_{ij} &= y_i y_j (K(x_i, x_j) + \lambda^2) \\
 D_{11} &= y_1 y_1 (K(x_1, x_1) + \lambda^2) \\
 D_{11} &= 1.1 (340,953 + 0.5^2) \\
 D_{11} &= (340,953 + 0,25)
 \end{aligned}$$

$$D_{11} = 341,203$$

Hasil lengkap dalam perhitungan matriks Hessian pada seluruh dokumen data latih dapat diketahui pada Tabel 4.22.

Tabel 0.22 Hasil perhitungan matriks Hessian

$D_{i,j}$	D1	D2	D3	D4	D5	D6
D1	341.203	-1.460	-1.460	-1.460	-1.460	1.290
D2	-1.460	189.043	3.562	4.726	1.642	-3.326
D3	-1.460	3.562	221.632	3.562	1.642	-1.460
D4	-1.460	4.726	3.562	673.969	2.831	-2.198
D5	-1.460	1.642	1.642	2.831	219.252	-1.460
D6	1.290	-3.326	-1.460	-2.198	-1.460	57.419

3. Langkah ketiga pada sequential learning adalah menghitung nilai E_i , $\delta\alpha_i$, dan α_i . Langkah-langkah perhitungan pada setiap tahapnya dijabarkan sebagai berikut:

- 4.7.10.1.1.3.1 Langkah awal yang dilakukan adalah menghitung nilai *error rate* yang ditunjukkan pada rumus Persamaan 2.14. Contoh perhitungan manual pada tahap ini dijelaskan sebagai berikut:

$$E_i = \sum_{j=1}^i a_j D_{ij}$$

$$E_1 = (341,203 \times 0) + (-1,460 \times 0) + (-1,460 \times 0) + (-1,460 \times 0) + (-1,460 \times 0) + (1,290 \times 0)$$

$$E_1 = 0$$

Hasil perhitungan error rate hingga iterasi ketiga ditunjukkan pada Tabel 4.23.

Tabel 0.23 Hasil perhitungan Error rate pada data latih

E_i	D1	D2	D3	D4	D5	D6
E_1	0	0	0	0	0	0
E_2	0.0337	0.0194	0.0227	0.0681	0.0222	0.0050
E_3	0.0662	0.0384	0.0450	0.1317	0.0440	0.0100



- b. Nilai *error rate* yang telah diketahui hasilnya akan digunakan pada perhitungan nilai delta alfa yang disimbolkan dengan $\delta\alpha_i$. Perhitungan manualisasi pada $\delta\alpha_i$ diiterasi kesatu dijelaskan sebagai berikut:

$$\delta\alpha_i = \min(\max[\gamma(1 - E_i), \alpha_i], C - \alpha_i)$$

$$\delta\alpha_1 = \min(\max[0.0001 \times (1 - 0), 0], 1 - 0)$$

$$\delta\alpha_1 = \min(\max[0.0001, 0], 1)$$

$$\delta\alpha_1 = \min(0.0001, 1)$$

$$\delta\alpha_1 = 0.0001$$

Nilai $\delta\alpha_i$ yang dilakukan pada iterasi maksimum pada manualisasi ditunjukkan pada Tabel 4.24.

Tabel 0.24 Hasil perhitungan delta alfa pada data latih

$\delta\alpha_i$	D1	D2	D3	D4	D5	D6
$\delta\alpha_1$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
$\delta\alpha_2$	0.00009 663	0.00009 806	0.00009 773	0.00009 319	0.00009 778	0.00009 950
$\delta\alpha_3$	0.00009 338	0.00009 615	0.00009 550	0.00008 680	0.00009 560	0.00009 900

- c. Setelah didapatkan nilai $\delta\alpha_i$, maka nilai α_i perlu diperbaharui untuk digunakan pada iterasi selanjutnya. Contoh manualisasi dari perhitungan α_i dijabarkan sebagai berikut:

$$\alpha_i = \alpha_i + \delta\alpha_i$$

$$\alpha_1 = 0 + 0.0001$$

$$\alpha_1 = 0.0001$$

Nilai α_i yang didapatkan hingga iterasi maksimum pada tahap manualisasi ditunjukkan pada Tabel 4.25.

Tabel 0.25 Hasil perhitungan α_i pada data latih

α_i	D1	D2	D3	D4	D5	D6
α_0	0	0	0	0	0	0
α_1	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
α_2	0.00019 7	0.00019 8	0.00019 8	0.00019 3	0.00019 8	0.00019 9
α_3	0.00029 0	0.00029 4209	0.00029 3222	0.00028 0	0.00029 3	0.00029 8

- d. Langkah akhir dari proses sequential learning adalah memeriksa batas maksimum iterasi yang telah ditentukan atau melihat nilai ($\delta\alpha_i \leq \epsilon$) untuk mengakhiri iterasi data. Pada proses manualisasi diinisialkan iterasi maksimum adalah 3. Maka hasil proses perhitungan akhir dari nilai E_i , $\delta\alpha$, dan α_i ditunjukkan pada Tabel 4.26, 4.27, dan 4.28.



Tabel 0.26 Hasil E_3 iterasi maksimum pada data latih

E_3	D1	D2	D3	D4	D5	D6
	0.0662	0.0384	0.0450	0.1317	0.0440	0.0100

Tabel 0.27 Hasil $\delta\alpha_3$ iterasi maksimum pada data latih

$\delta\alpha_3$	D1	D2	D3	D4	D5	D6
	0.00009 338	0.00009 615	0.00009 550	0.00008 680	0.00009 560	0.00009 900

Tabel 0.28 Hasil α_3 iterasi maksimum pada data latih

α_3	D1	D2	D3	D4	D5	D6
	0.00029 0	0.00029 4209	0.00029 3222	0.00028 0	0.00029 3	0.00029 8

- e. Pada tahap perhitungan sequential learning akan didapatkan nilai akhir α_i yang berfungsi sebagai nilai Support Vector (SV). Memasuki tahapan klasifikasi maka perlu dicari nilai bias yang ditunjukkan pada persamaan 2.16. Untuk mencari nilai bias, harus diketahui nilai x^- dan x^+ . Kedua nilai tersebut didapatkan dari nilai maksimum α_i dari kelas positif (x^+) dan nilai maksimum α_i dari kelas negatif (x^-). Pada perhitungan manual didapatkan nilai maksimum α_i kelas x^+ pada dokumen D1 dan nilai maksimum kelas x^- pada dokumen D6. Nilai bias yang didapatkan pada proses manualisasi ditunjukkan pada Tabel 4.29.

Tabel 0.29 Hasil perhitungan nilai bias

x_i	$K(x_i, x^+)$	$K(x_i, x^-)$	$\alpha_i y_i K(x_i, x^+)$	$\alpha_i y_i K(x_i, x^-)$
D1	1.04	1.21	0.0003016	0.0003509
D2	3.076	188.793	-0.0009050	-0.0555447
D3	1.21	3.312	-0.0003548	-0.0009712
D4	1.948	4.476	-0.0005454	-0.0012532
D5	1.21	1.392	-0.0003550	-0.0004084
D6	57.169	3.076	0.0170645	0.0009182
Total			0.0152060	-0.0569083
Bias			0.0169022	

- f. Langkah akhir dalam proses klasifikasi data dengan metode *Support Vector Machine* adalah memasukkan data uji yang ingin diketahui kelas sentimennya ke *hyperplane* yang telah ditemukan pada perhitungan sebelumnya. Proses klasifikasi data dilakukan dengan



menggunakan Persamaan 2.18. Nilai positif atau negatif yang ditunjukkan pada hasil akhir perhitungan $Sign(f(x))$ akan menentukan kelas data uji. Jika nilai perhitungan menunjukkan -1, maka dokumen data uji termasuk sebagai kelas negatif. Namun jika hasil perhitungan menunjukkan +1, maka dokumen data uji termasuk sebagai kelas positif. Hasil perhitungan data uji pada proses manualisasi ditunjukkan pada Tabel 4.30.

Tabel 0.30 Hasil perhitungan klasifikasi pada data uji

$\alpha_i y_i K(x, x_i)$	U1	U2	U3
D1	0.00035092	0.00125569	0.00125569
D2	-0.0004097	-0.00040966	-0.0004097
D3	-0.0004083	-0.00075667	-0.0007567
D4	-0.0005547	-0.00055469	-0.0005547
D5	-0.0009718	-0.00165396	-0.001654
D6	0.00036118	0.00067571	0.00069547
$\sum_{i=0}^m \alpha_i y_i K(x_i, x)$	-0.0002941	-0.00071059	-0.0007106
$\sum_{i=0}^m \alpha_i y_i K(x_i, x) + b$	0.00029847	0.00059007	0.00060728
Sentimen	Positif	Positif	Positif

4.7.11 Manualisasi Perhitungan Evaluasi

Tahapan ini akan dilakukan perhitungan secara manual terhadap hasil evaluasi metode klasifikasi SVM yang telah dilakukan. Mulanya untuk melakukan tahapan evaluasi, perlu mengisi tabel dalam *confusion matrix*. Hal ini dilakukan dalam upaya mempermudah perhitungan nilai akurasi, presisi, *recall*, dan *f-measure*. Hasil yang diberikan dalam bentuk tabel *confusion matrix* ditunjukkan oleh Tabel 4.31.

Tabel 0.31 Hasil Confusion Matrix

Classification	Predicted Positives	Predicted Negatives
<i>Actual Positive Cases</i>	2	0
<i>Actual Negative Cases</i>	1	0

Berdasarkan Tabel 4.31, didapatkan hasil perhitungan manualisasi terhadap tiga data uji. Pada tabel ditunjukkan bahwa dua diantara tiga data uji telah diklasifikasikan dengan benar, namun satu data uji masih diprediksikan salah dengan kelas aktualnya adalah negatif dan hasil klasifikasi yang diberikan adalah kelas positif. Dari hasil *confusion matrix* dapat dihitung nilai akurasi, presisi, *recall*, dan *f-measure* sebagai berikut:



$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} = \frac{0 + 2}{0 + 2 + 1 + 0} = 0,6667$$

$$Precision = \frac{TP}{TP + FP} = \frac{2}{2 + 1} = 0,6667$$

$$Recall = \frac{TP}{TP + FN} = \frac{2}{2 + 0} = 1$$

$$F - Measure = \frac{2 \times 0,6667 \times 1}{0,6667 + 1} = 0,8$$

4.8 Perancangan Pengujian

Tahap pengujian akan diberlakukan dengan tujuan untuk mengetahui tingkat keakuratan sistem yang telah dibuat. Tingkat akurasi yang digunakan adalah kesesuaian antara hasil sistem yang berjalan dengan rancangan kebutuhan. Terdapat tiga macam pengujian yang diterapkan, yaitu pengujian terhadap parameter *Support Vector Machine*, pengaruh implementasi normalisasi kata baku dan *Lexicon Based Features*.

4.8.1 Perancangan Pengujian Terhadap Parameter *Support Vector Machine*

Pada pengujian parameter *Support Vector Machine*, dilakukan pengujian terhadap parameter *degree* pada kernel polynomial dengan komposisi data latih dan data uji yang berbeda, pengaruh dari nilai learning rate (γ) serta jumlah iterasi maksimum yang digunakan. Selain itu proses pengujian pada tahapan ini akan menghitung waktu komputasi yang dibutuhkan dalam menjalankan sistem. Waktu komputasi akan didapatkan berdasarkan jumlah iterasi maksimum yang dilakukan pada tahapan klasifikasi dengan metode *Support Vector Machine*. Pengujian ini dilakukan untuk mengetahui jumlah iterasi maksimum terbaik untuk mendapatkan nilai waktu komputasi terendah yang digunakan oleh sistem. Perancangan pengujian pada parameter *Support Vector Machine* ditunjukkan pada Tabel 4.32 dan 4.33.

Tabel 0.32 Perancangan Pengujian Nilai *Degree*

Evaluasi	Kernel Polynomial (Nilai <i>degree</i>)				
	2	3	4	5	6
<i>Accuracy</i>					
<i>Precision</i>					
<i>Recall</i>					
<i>F-Measure</i>					



Tabel 0.33 Perancangan Pengujian Konstanta Learning Rate γ

Iterasi		Learning rate γ					Rata-Rata	Waktu Komputasi (s)
		0,0001	0,0005	0,001	0,0025	0,05		
50	Accuracy							
	Precision							
	Recall							
	F-Measure							
100	Accuracy							
	Precision							
	Recall							
	F-Measure							
150	Accuracy							
	Precision							
	Recall							
	F-Measure							
200	Accuracy							
	Precision							
	Recall							
	F-Measure							
Rata-rata	Accuracy							
	Precision							
	Recall							
	F-Measure							

4.8.2 Perancangan Pengujian Implementasi *Lexicon Based Features*

Pengujian yang diimplementasikan pada sistem dilakukan dengan membandingkan tingkat akurasi ketika sistem menggunakan *Lexicon Based Features* dengan normalisasi *min-max* dan sistem yang tidak menerapkan *Lexicon Based features* selain itu terdapat perancangan pengujian tingkat akurasi pada sistem yang menggunakan *Lexicon Based Features* dengan perhitungan skor sentimen. Dari tahap pengujian ini akan diketahui apakah fitur kata yang mengandung sentimen positif ataupun negatif memiliki pengaruh yang besar atau



tidak terhadap jalannya sistem klasifikasi. Perancangan pengujian ditunjukkan pada Tabel 4.34.

Tabel 0.34 Perancangan pengujian pengaruh implementasi *Lexicon Based Features* dengan Normalisasi *Min-Max* dan *Lexicon Based Features* dengan skor sentimen

Perbandingan data latih:data uji		Tanpa <i>Lexicon Based Features</i>	<i>Lexicon Based Features</i> dengan normalisasi <i>min-max</i>	<i>Lexicon Based Features</i> dengan skor sentimen
50:50	<i>Accuracy</i>			
	<i>Precision</i>			
	<i>Recall</i>			
	<i>F-Measure</i>			
60:40	<i>Accuracy</i>			
	<i>Precision</i>			
	<i>Recall</i>			
	<i>F-Measure</i>			
70:30	<i>Accuracy</i>			
	<i>Precision</i>			
	<i>Recall</i>			
	<i>F-Measure</i>			
80:20	<i>Accuracy</i>			
	<i>Precision</i>			
	<i>Recall</i>			
	<i>F-Measure</i>			
90:10	<i>Accuracy</i>			
	<i>Precision</i>			
	<i>Recall</i>			
	<i>F-Measure</i>			
Rata - rata	<i>Accuracy</i>			
	<i>Precision</i>			
	<i>Recall</i>			
	<i>F-Measure</i>			



4.8.3 Penarikan Kesimpulan

Setelah seluruh tahap pengujian yang dilakukan selesai, maka akan dilakukan penarikan kesimpulan. Hal ini dilakukan dengan menganalisis setiap hasil tahap pengujian yang ditunjukkan dengan tabel maupun grafik hasil pengujian sistem analisis. Kesimpulan dapat memuat kekurangan maupun kelebihan dari pengimplementasian sistem maupun dari penulisan dokumen yang telah dilakukan. Dari kesimpulan yang diambil, penulis dapat memberikan saran yang dibuat berdasarkan kekurangan-kekurangan pada sistem. Dari kekurangan tersebut dapat dibenahi maupun dikembangkan oleh peneliti selanjutnya.



BAB 5 IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi sistem analisis sentimen *cyberbullying* pada komentar Instagram dengan metode klasifikasi *Support Vector Machine* berdasarkan perancangan yang telah dijelaskan pada bab sebelumnya.

5.1 Implementasi Sistem

Pada implementasi sistem akan menguraikan tahapan dalam analisis sentimen *cyberbullying* pada komentar instagram dengan metode klasifikasi *Support Vector Machine*. Tahapan ini diuraikan sesuai dengan perancangan sistem yang telah dibuat dan dijelaskan pada bab sebelumnya.

Dalam melakukan analisis sentimen perlu melalui beberapa tahapan yaitu meliputi tahapan *Pre-processing* teks yang dilanjutkan dengan pembobotan kata hingga proses klasifikasi data. Pada tahapan *Pre-processing* ini dilakukan beberapa sub-proses yaitu *casefolding*, *data cleaning*, normalisasi bahasa, *stopword removal*, *stemming*, dan tokenisasi. Hasil dari tahapan *Pre-processing* akan masuk kedalam tahapan perhitungan TF-IDF yang memiliki beberapa sub-proses yaitu perhitungan *tf*, *wtf*, *df*, *idf*, dan *tfidf*. Kemudian menjalankan proses klasifikasi dengan metode *Support Vector Machine*, dimana data akan diproses dan diklasifikasikan sebagai komentar positif *cyberbullying* atau negatif *cyberbullying*.

5.2 Pre-processing

Pre-processing merupakan tahapan awal dalam memproses teks yang dimasukkan pada sistem. Tahapan ini akan memproses teks data input dengan beberapa sub-proses yang akan dilalui untuk diketahui fitur-fitur yang membantu dalam tahapan klasifikasi. Tahapan *Pre-processing* meliputi *case folding*, *data cleaning*, normalisasi bahasa, *stopword removal*, *stemming*, dan tokenisasi.

5.2.1 Case Folding

Tahapan awal dari *Pre-processing* adalah menginputkan data yang diproses yang berbentuk file csv, yang kemudian dilakukan proses *case folding*. Proses ini berfungsi untuk mengubah seluruh teks menjadi *lower case* (huruf kecil). Implementasi dari tahapan *case folding* ditunjukkan pada Kode Program 5.1.

```

1 def CaseFolding(self, term):
2     for i in range(len(term)):
3         term[i][0] = term[i][0].lower()
4     return term

```

Kode Program 0.1 Implementasi Case Folding

Potongan program yang diimplementasikan merupakan kode program dari *case folding* yang dapat dijelaskan sebagai berikut:

1. Baris 2-4 menunjukkan proses perulangan pada panjang data dan merubah data yang berupa komentar menjadi huruf kecil.

5.2.2 Data Cleaning

Setelah mendapatkan hasil dari *case folding*, proses yang dilalui adalah *data cleaning*. Pada tahapan ini teks yang diperoleh dari hasil *case folding* akan dibersihkan dengan menghapus seluruh link url, username, tanda baca, dan karakter huruf. Implementasi dari *data cleaning* ditunjukkan oleh Kode Program 5.2.

```

1 def DataCleaning(self, term):
2
3     for i in range(len(term)):
4         term[i][0] = re.sub(r'(\A\s)@(\w+) | (^https?:\/\/.*[\r\n]*)
5 | (\A\s)[0-9](\w+) | (\d)', " ", term[i][0])
6         term[i][0] = re.sub(r'\d', " ", term[i][0])
7         term[i][0] = re.sub(r'[\.\?&^*!()/\,`~;:<>#]', " ",
8 term[i][0])
9
10    return term

```

Kode Program 0.2 Implementasi Data cleaning

Proses implementasi pada kode program diatas dapat diuraikan pada keterangan berikut:

1. Baris 3-6 adalah proses penghapusan link url, username, angka, dan tanda baca dengan menggunakan regex (*regular expression*).

5.2.3 Normalisasi Bahasa

Normalisasi Bahasa merupakan tahapan yang bertujuan untuk melakukan pemeriksaan pada setiap term pada data yang dimasukkan. Jika pada term merupakan kata tidak baku, maka akan dilakukan proses pernormalisasi bahasa. Term yang diketahui sebagai kata tidak baku diubah menjadi kata yang baku, sesuai dengan kamus normalisasi. Proses normalisasi bahasa pada tahapan *pre-processing* ditunjukkan pada Kode Program 5.3.

```

1 def Normalisasi(self, term):
2     with open('normalisasi.csv', 'r') as kamusAlay:
3         reader = csv.reader(kamusAlay, delimiter=',')
4         fileAlay = list(reader)
5         for i in range(len(term)):
6             split = term[i][0].split()
7             for x in range(len(split)):
8                 for j in range(len(fileAlay)):
9                     if split[x] == fileAlay[j][0]:
10                        split[x] = fileAlay[j][1]
11                        term[i][0] = " ".join(split)
12
13    return term

```

Kode Program 0.3 Implementasi Normalisasi Bahasa

Berdasarkan kode program yang diimplementasikan pada sistem dalam melakukan proses normalisasi bahasa, dijelaskan pada keterangan berikut:

1. Baris 2-4 adalah proses membaca kamus normalisasi bahasa yang berekstensi csv. File kamus tersebut disimpan dalam bentuk list pada program.

- Baris 5-12 merupakan proses pengecekan pada setiap data. Dimana data teks akan dipecah terlebih dahulu dengan menggunakan split sehingga dapat diperiksa setiap kata. Kata atau term yang cocok dengan kamus tidak baku maka akan diganti dengan kata yang baku. Kata baku yang dimaksud ialah kata yang tidak disingkat atau tidak berlebihan (*alay*).

5.2.4 Stopword Removal

Tahapan *stopword removal* merupakan tahapan filterisasi dari data yang telah melalui proses normalisasi. Kata yang akan dihapus merupakan kata yang tidak memiliki arti, sehingga dapat mengurangi fitur term dalam proses pembobotan. Tahapan *stopword removal* pada *pre-processing* penelitian ini menggunakan *stoplist* tala. Proses *stopword removal* ini ditunjukkan pada Kode Program 5.4.

```

1  def Stopword(self, term):
2      stoplist =self.stoplist()
3      for i in range(len(term)):
4          split = term[i][0].split()
5          term[i][0] = (" ".join(c for c in split if c not in
6 stoplist))
7      return term

```

Kode Program 0.4 Implementasi Stopword Removal

Potongan kode program pada implementasi *stopword removal* dapat diuraikan pada keterangan berikut:

- Baris 2 merupakan pemanggilan method *stoplist* yang berfungsi dalam membaca file yang berisi daftar *stopword*.
- Baris 3-7 proses perulangan dimana komentar akan dipecah menjadi term terpisah. Dalam proses perulangan dijalankan pula proses penghapusan term yang sesuai dengan daftar *stoplist*. Namun setelah beberapa term yang sesuai pada *stoplist* dihapus, pemecahan kata akan digabungkan kembali menjadi kalimat.

5.2.5 Stemming

Pada tahapan *stemming* adalah mengubah data menjadi *root* atau kata dasar. Pada tahapan ini menggunakan *stemming* sastrawi dalam mengubah kata yang memiliki imbuhan *prefix*, *infix*, maupun *suffix* menjadi kata dasar. Tahapan *stemming* dalam *Pre-processing* ditunjukkan pada Kode Program 5.5.

```

1  def Stopword(self, term):
2      stoplist =self.stoplist()
3      for i in range(len(term)):
4          split = term[i][0].split()
5          term[i][0] = (" ".join(c for c in split if c not in
6 stoplist))
7      return term

```

Kode Program 0.5 Implementasi Stemming

Potongan kode program 5.5 menunjukkan implementasi pada tahapan *stemming*. Potongan kode tersebut dapat diuraikan pada keterangan berikut:

1. Baris 2 berfungsi dalam memanggil method `stoplist` untuk membuka file `stoplist` yang digunakan dalam *stopword removal*
2. Baris 3-7 proses perulangan dalam memeriksa setiap kata pada data yang dimasukkan. Data yang telah diperiksa akan diperiksa satu-persatu dengan file `stoplist`. Jika terdapat kecocokan antara kata yang terdapat pada file data dengan `stoplist`, maka kata yang terdapat pada data akan dihapuskan.

5.2.6 Tokenisasi

Tahapan selanjutnya merupakan tokenisasi yang merupakan tahapan akhir dari *Pre-processing*. Pada tahapan ini dilakukan pemecahan kalimat menjadi term. Potongan dari proses tokenisasi ditunjukkan pada Kode Program 5.5.

```

1 def Tokenize(self, term):
2     for i in range(len(term)):
3         term[i][0] = term[i][0].split()
4     return term

```

Kode Program 0.6 Implementasi Tokenisasi

Proses tokenisasi yang telah ditunjukkan pada Kode Program 5.5 diuraikan dengan penjelasan berikut:

1. Baris 2-4 merupakan proses perulangan pada komentar yang dimasukkan dimana pada proses tersebut, komentar dipecah menjadi term yang terpisah.

5.3 Perhitungan Kata

Tahapan selanjutnya setelah dilakukan *Pre-processing* pada data latih adalah proses pembobotan kata. Agar dapat membobotkan setiap kata, maka perlu diketahui fitur kata yang akan digunakan, sehingga untuk memulai pembobotan akan dilakukan proses pencarian fitur. Setelah fitur telah ditemukan akan dijalankan proses pembobotan kata. Proses pembobotan diperlukan ketika melakukan analisis sentimen, dimana proses ini berfungsi dalam menentukan bobot pada setiap fitur kata berdasarkan tingkat kemunculan kata pada dokumen teks tersebut maupun dokumen teks lainnya. Untuk memperoleh bobot pada setiap fitur kata perlu dilakukan beberapa tahap perhitungan yang meliputi perhitungan nilai `tf` dan `wtf`, perhitungan `df` dan `idf`, dan proses perhitungan nilai `TF-IDF`.

5.3.1 Pencarian Fitur Kata

Sebelum memasuki proses pembobotan kata, maka perlu diketahui fitur-fitur kata yang akan digunakan. Pada proses ini akan diketahui fitur-fitur kata setelah proses tokenisasi pada tahapan *Pre-processing* teks telah selesai dilakukan. Tahapan pada pencarian kata akan ditunjukkan oleh Kode Program 5.7.

```

1 def fiturkata(self, hasilPre-processing):
2     fitur = []
3     for i in range(len(hasilPre-processing)):
4         for word in hasilPre-processing[i][0]:
5             if word not in fitur:

```

```

5         fitur.append(word)
6     return fitur
    
```

Kode Program 0.7 Implementasi Pencarian Fitur Kata

Berdasarkan kode program diatas dalam mencari fitur kata, akan diuraikan mengenai tahapan pencarian fitur dalam penjelasan berikut:

1. Baris 2 mendeklarasikan list yang diberi nama unik yang berfungsi sebagai penyimpan data yang berupa fitur kata.
2. Baris 4-6 proses pencarian fitur kata pada hasil *Pre-processing* teks. Jika terdapat kata yang sama dengan list yang menampung fitur, maka kata yang sama tersebut tidak akan dimasukkan pada list unik.

5.3.2 Perhitungan Nilai TF dan $Wtf_{t,d}$

Tahapan awal yang dilakukan ketika melakukan proses pembobotan kata ialah dengan menghitung nilai tf dan $wtf_{t,d}$. Pada proses ini bertujuan untuk mengetahui bobot frekuensi kemunculan kata pada hanya pada dokumen teks tersebut. Proses perhitungan nilai tf dan $Wtf_{t,d}$ ditunjukkan pada Kode Program 5.8.

```

1  def tf(self, hasilPre-processing, fitur):
2      tf = []
3      for i in range(len(fitur)):
4          temp= []
5          for j in range(len(hasilPre-processing)):
6              temp.append(hasilPre-
7 processing[j][0].count(fitur[i]))
8          tf.append(temp)
9      return tf
10
11 def wtf(self, tf):
12     wtf = []
13     for i in range(len(tf)):
14
15         temp= []
16         for j in range(len(tf[i])):
17             hasil = 0
18             if tf[i][j] > 0:
19                 hasil = 1 + math.log10(tf[i][j])
20             temp.append(hasil)
21         wtf.append(temp)
22     return wtf
    
```

Kode Program 0.8 Implementasi perhitungan nilai tf dan wtf

Dalam implementasi kode program diatas bertujuan untuk menghitung nilai tf dan $wtf_{t,d}$ pada fungsi *def* yang berbeda. Fungsi *def* yang telah diikuti oleh parameter akan diuraikan sebagai berikut:

1. Baris 2 merupakan deklarasi list yang berfungsi untuk menyimpan hasil perhitungan tf.
2. Baris 3-7 adalah proses perulangan yang berjalan pada hasil *pre-processing* yang berupa tokenisasi. Hasil *pre-processing* tersebut dibandingkan dengan fitur kata yang telah ditentukan. Jika terdapat kesamaan antara

fitur dan token kata pada hasil *pre-processing*, maka nilai *tf* akan bertambah satu.

3. Baris 11 adalah deklarasi variabel yang berbentuk list untuk menyimpan hasil perhitungan $wf_{t,d}$.
4. Baris 12-19 perulangan pada hasil *tf*, ketika pada *tf* memiliki nilai lebih dari nol maka akan dihitung nilai $wf_{t,d}$.

5.3.3 Perhitungan Nilai DF dan IDF

Setelah diketahui frekuensi tingkat kemunculan kata tersebut dilakukan proses selanjutnya yaitu dengan mencari nilai *df* dan *idf*. Proses yang dilakukan bertujuan untuk menghitung bobot dari frekuensi kata yang muncul dari dokumen-dokumen yang ada. Implementasi perhitungan *df* dan *idf* ditunjukkan pada kode program 5.9.

```

1  def dfidf(self, tf):
2      df = []
3      idf = []
4      for i in range(len(tf)):
5          hasil_df = 0
6          for j in range(len(tf[i])):
7              if tf[i][j] > 0:
8                  hasil_df = tf[i][j]+hasil_df
9              hasil_idf =
10     math.log10(float((len(tf[i])))/float(hasil_df))
11     df.append(hasil_df)
12     idf.append(hasil_idf)
13     return df, idf

```

Kode Program 0.9 Implementasi perhitungan nilai *df* dan *idf*

Kode program yang telah ditunjukkan merupakan langkah program untuk menghitung nilai *df* dan *idf*. Uraian pada baris program akan dijelaskan pada keterangan berikut:

1. Baris 2-3 deklarasi variabel yang berfungsi dalam menyimpan nilai *df* dan *idf*.
2. Baris 4-12 merupakan perulangan pada hasil perhitungan *tf* dimana jika nilai dari kolom *tf* lebih dari nol maka akan menghitung nilai *df* dan diikuti menghitung nilai *idf*.

5.3.4 Perhitungan Nilai TF-IDF

Tahapan akhir dari proses pembobotan kata adalah menghitung nilai TF-IDF dengan cara menghitung dari setiap proses yang telah dilakukan sebelumnya. Proses perhitungan TF-IDF dilakukan dengan tujuan untuk mengetahui bobot dari kata yang banyak muncul pada suatu dokumen namun jarang muncul pada dokumen lainnya. Kejadian tersebut merupakan kondisi dimana kata tersebut dapat menjadi kata yang penting dan memiliki bobot yang lebih. Pada tahapan ini akan mengkalikan nilai *wf* dan *idf* yang telah didapatkan. Implementasi dari proses ini ditunjukkan pada Kode Program 5.10.


```

1 def tfidf(self,wtf, idf):
2     tfidf = []
3     for i in range(len(idf)):
4         temp = []
5         for j in range(len(wtf[i])):
6             if wtf[i][j]> 0:
7                 hasil = wtf[i][j] * idf[i]
8                 temp.append(hasil)
9             else:
10                temp.append(0)
11        tfidf.append(temp)
12    return tfidf

```

Kode Program 0.10 Implementasi perhitungan TF-IDF

Implementasi perhitungan TF-IDF yang telah ditunjukkan pada kode program diatas diuraikan pada keterangan berikut:

1. Baris 2 adalah deklrasi variable yang berbentuk list untuk menyimpan nilai TF-IDF yang telah ditemukan.
2. Baris 3-11 merupakan bentuk perulangan yang dilakukan pada hasil perhitungan wtd dan nilai idf. Jika nilai wtd lebih dari nol, maka akan dihitung untuk mencari nilai TF-IDF.

5.4 Pembobotan Lexicon Based Features

Dalam membantu pembobotan TF-IDF untuk mencari hasil klasifikasi, dilakukan pula tahapan pembobotan menggunakan *Lexicon Based Features*. Tahapan ini dilakukan dengan mebobot kata yang memiliki sentimen positif ataupun sentimen negatif yang sesuai dengan daftar *lexicon* yang digunakan. Untuk menghindari perbedaan nilai pada bobot *lexicon* yang signifikan, maka dilakukan proses normalisasi min-max. Implementasi dari perhitungan pembobotan *lexicon* ditunjukkan oleh Kode Program 5.11.

```

1  ## normalisasi min max
2  def bobotlex(self, hasilPre-processing):
3      with open('lexicon.csv', 'r') as lexicon:
4          file = csv.reader(lexicon, delimiter=',')
5          filelex = list(file)
6          newmax = 0.9
7          newmin = 0.1
8          bobot_lex = []
9          norm_lex= []
10
11     for i in range(len(hasilPre-processing)):
12         hp = hasilPre-processing[i][0]
13         lexpos = lexneg = 0
14         temp_bobot = []
15         temp_norm = []
16         for x in range(len(hp)):
17             for j in range(len(filelex)):
18                 if hp[x] == filelex[j][0] and filelex[j][1]
19 == 'positif':
20                     lexpos += 1
21                 elif hp[x] == filelex[j][0] and filelex[j][1]
22 == 'negatif':

```

```

23         lexneg += 1
24     maks = max (lexpos, lexneg)
25     mins = min(lexpos, lexneg)
26     if (maks - mins) > 0:
27         normpos = (lexpos - mins) / (maks - mins) *
28 (newmax - newmin) + newmin
29         normneg = (lexneg - mins) / (maks - mins) *
30 (newmax - newmin) + newmin
31     else:
32         normpos = newmin
33         normneg = newmin
34     temp_bobot.append(lexpos)
35     temp_bobot.append(lexneg)
36     bobot_lex.append(temp_bobot)
37     temp_norm.append(normpos)
38     temp_norm.append(normneg)
39     norm_lex.append(temp_norm)
40
41     return bobot_lex, norm_lex
42 ## skor sentimen
43 def skorlex(self, hasilPre-processing):
44     with open('lexicon.csv', 'r') as lexicon:
45         file = csv.reader(lexicon, delimiter=',')
46         filelex = list(file)
47         skor_lex = []
48         for i in range(len(hasilPre-processing)):
49             hp = hasilPre-processing[i][0]
50             lexpos = lexneg = 0
51             for x in range(len(hp)):
52                 for j in range(len(filelex)):
53                     if hp[x] == filelex[j][0] and filelex[j][1]
54 == 'positif':
55                         lexpos += 1
56                     elif hp[x] == filelex[j][0] and filelex[j][1]
57 == 'negatif':
58                         lexneg += 1
59             len_comm = len(hp)
60             pol_pos = lexpos / len_comm
61             pol_neg = lexneg / len_comm
62             temp_skor = pol_pos - pol_neg
63             skor_lex.append(temp_skor)
64     return skor_lex

```

Kode Program 0.11 Implementasi perhitungan pembobotan *Lexicon Based Features*

Implementasi dari pembobotan *Lexicon Based Features* yang ditunjukkan pada kode program diatas dapat diuraikan langkahnya pada penjelasan berikut:

1. Baris 1 memberikan keterangan pada kode program yang digunakan untuk pembobotan *Lexicon Based Features* dengan normalisasi *min-max*.
2. Baris 2 adalah kode program yang berfungsi dalam membaca file *lexicon*.
3. Baris 6-7 inialisasi variabel yang memiliki nilai maksiman dan nilai minimum untuk digunakan pada langkah normalisasi.

4. Baris 11-25 proses perulangan pada hasil *pre-processing* dimana akan memeriksa token kata dengan kamus *lexicon*. Jika token kata terdapat pada kamus *lexicon* maka akan diperiksa pula sentimennya. Setiap kata yang memiliki sentimen maka *counter* akan bertambah satu pada setiap sentimennya.
5. Baris 27- 41 merupakan proses normalisasi nilai bobot *lexicon* yang telah ditemukan. Nilai bobot maksimal diubah bernilai 0,9 dan nilai bobot minimal diubah menjadi 0,1. Namun ketika bobot nilai *lexicon* sama, secara otomatis bobot nilai dinormalisasikan menjadi nilai minimal yaitu 0,1.
6. Baris 42 merupakan keterangan bahwa implementasi dibawahnya merupakan kode program untuk *Lexicon Based Features* dengan skor sentimen.
7. Baris 43 adalah kode program yang berfungsi dalam membaca file *lexicon*.
8. Baris 47 merupakan inisiliasi variabel list untuk menyimpan hasil skor sentimen.
9. Baris 48-58 proses perulangan pada hasil *pre-processing* dimana akan memeriksa token kata dengan kamus *lexicon*. Jika token kata terdapat pada kamus *lexicon* maka akan diperiksa pula sentimennya. Setiap kata yang memiliki sentimen maka *counter* akan bertambah satu pada setiap sentimennya.
10. Baris 59-64 merupakan langkah perhitungan untuk mencari nilai polaritas sentimen positif dan sentimen negatif pada setiap dokumen komentar. Yang kemudian hasil dari polaritas tersebut akan dicari selisihnya untuk mendapatkan skor sentimen.

5.5 Klasifikasi dengan *Support Vector Machine*

Tahapan klasifikasi menggunakan *Support Vector Machine* merupakan tahapan inti dalam melakukan analisis sentimen. Tahapan ini melakukan pencarian garis pemisah atau *hyperplane* antara data berkelas positif dan data yang berkelas negatif. Proses klasifikasi dilakukan dengan menghitung kernel, matriks hessian, *sequential learning*, penentuan *support vector*, *bias*, dan penentuan kelas positif dan kelas negatif pada data uji.

5.5.1 Pembentukan Matriks dan Transposisi Matriks

Proses awal yang dilalui adalah mengubah data menjadi bentuk matriks dan bentuk transposisi. Hal ini dilakukan karena pada tahapan selanjutnya diperlukan perkalian data yang berbentuk matriks. Implementasi untuk membentuk matrik ditunjukkan pada Kode Program 5.12.

```

1  ## normalisasi min-max
2  def matrixfitur(self, tfidf, normlexicon):
3      temp_normpos = []
4      temp_normneg = []
5      for i in range(len(normlexicon)):
6          temp_normpos.append(normlexicon[i][0])
7          temp_normneg.append(normlexicon[i][1])
8          tfidf.append(temp_normpos)
9          tfidf.append(temp_normneg)
10
11     return tfidf
12
13     ## skor sentimen
14     def matrixskor(self, tfidf, skorlex):
15         tfidf.append(skorlex)
16         return tfidf
17
18     def transpose(self, tfidf):
19         transpose = []
20
21         for i in range(len(tfidf[0])):
22             temp = []
23             for j in range(len(tfidf)):
24                 temp.append(tfidf[j][i])
25             transpose.append(temp)
26
27         return transpose

```

Kode Program 0.12 Implementasi pembentukan matriks dan transposisi

Kode program diatas merupakan langkah-langkah dalam membentuk matriks dan matriks tranpos. Kode Program 5.12 dapat diuraikan keterangannya pada penjelasan berikut:

1. Baris 2-8 merupakan langkah dalam menyisipkan bobot lexicon pada perhitungan TF-IDF yang telah dinormalisasikan dengan min-max
2. Baris 14-15 merupakan langkah dalam menyisipkan bobot lexicon pada perhitungan TF-IDF yang dihitung dengan cara mencari skor sentimen.
3. Baris 18-27 merupakan langkah untuk transposisi dari matriks yang telah dibentuk pada langkah sebelumnya.

5.5.2 Perhitungan Kernel

Perhitungan kernel adalah proses pertama dalam melakukan perhitungan klasifikasi dengan metode *Support Vector Machine*. Nilai yang didapatkan dari hasil perhitungan kernel berguna ketika dilakukan proses selanjutnya, yaitu mencari nilai dari matriks *Hessian*. Pada tahapan ini akan diterapkan perhitungan dengan kernel polynomial yang diimplementasikan pada Kode Program 5.13.

```

1  def kernel(self, mat, trans, d):
2      c = 1
3      mat_kernel = []
4
5      for i in range(0, len(trans)):
6          temp = []
7          for j in range(0, len(trans)):

```

```

8         total = 0
9         for x in range(0, len(mat[0])):
10            hasil = trans[i][x]*mat[x][j]
11            total = total + (hasil)
12            tambah_c = total +c
13            hasil_pangkat = math.pow(tambah_c, d)
14            kernel = hasil_pangkat
15            temp.append(kernel)
16            mat_kernel.append(temp)
17     return mat_kernel
18

```

Kode Program 0.13 Implementasi perhitungan kernel

Perhitungan kernel dilakukan dengan menjalankan kode program yang telah ditunjukkan. Langkah-langkah dalam menghitung kernel pada kode program diuraikan pada keterangan berikut:

3. Baris 2 adalah menginisialisasikan nilai c yaitu 1.
4. Baris 5-18 merupakan langkah dalam menghitung matriks yang dilakukan dengan menggunakan perulangan. Untuk menghitung kernel polynomial diperlukan matriks hasil perhitungan TF-IDF yang dikalikan dengan transposisinya. Hasil perkalian matriks tersebut kemudian akan dijumlahkan dengan nilai c yang telah diinisialisasikan sebelumnya dan kemudian dipangkatkan dengan nilai *degree*. Nilai *degree* merupakan parameter yang dapat diubah.

5.5.3 Perhitungan Matriks Hessian

Tahapan selanjutnya adalah menghitung nilai matriks hessian dengan cara mengkalikan hasil perhitungan kernel yang didapatkan dengan kelas pada setiap data latih dan kemudian akan ditambah dengan nilai lambda yang telah ditentukan. Proses implementasi pada tahapan ini ditunjukkan pada Kode Program 5.14.

```

1  def hessian(self, kelas, kernel):
2      lamda = 0.5
3      hessian = []
4      for i in range (len(kernel)):
5          temp = []
6          for j in range(0, len(kernel[0])):
7              hess = kelas[i]*kelas[j] * kernel[i][j]+
8              (math.pow(lamda , 2))
9              temp.append(hess)
10             hessian.append(temp)
11
12     return hessian

```

Kode Program 0.14 Implementasi perhitungan matriks hessian

Kode implementasi diatas merupakan langkah yang dilakukan oleh sistem untuk menghitung Matriks Hessian. Langkah-langkah perhitungan sistem dapat diuraikan dalam penjelasan berikut:

1. Baris 2 adalah inisialisasi variabel yang merupakan nilai lambda.

- Baris 4-10 langkah perhitungan matriks hessian dengan mengkalikan kelas pada setiap dokumen kelas dengan hasil kernel yang telah didapatkan. Untuk mendapatkan nilai hessian hasil perkalian tersebut dijumlahkan dengan nilai lambda yang telah dipangkatkan. Hasil perhitungan matriks Hessian disimpan dalam list hessian.

5.5.4 Perhitungan Sequential Learning

Setelah dilakukan perhitungan matriks hessian, kemudian dilanjutkan dengan menghitung *sequential learning*. Proses ini bertujuan untuk mencari nilai *support vector* yang merupakan titik terdekat dengan *hyperplane* yaitu pembatas antar dua kelas. Implementasi perhitungan dari tahapan *sequential learning* ditunjukkan pada Kode Program 5.15.

```

1  def sequential(self, hessian, itermax, lr ):
2      epsilon = 0.0001
3      c = 1
4      alfa = []
5      E = []
6      temp_a = []
7      delta_a = []
8
9      for i in range(len(hessian)):
10         temp_a.append(0)
11         alfa.append(temp_a)
12
13         for i in range(itermax):
14             temp_e = []
15             temp_d = []
16             temp_a = []
17             for j in range(len(hessian)):
18                 total = 0
19                 for x in range(len(hessian[0])):
20                     total = total + hessian[j][x] * alfa[-1][x]
21                 temp_e.append(total)
22             E.append(temp_e)
23             for x in range(len(E[-1])):
24                 hasil = min(max(lr* (1-E[-1][x]), -alfa[-1][x]),
25 c -alfa[-1][x])
26                 temp_d.append(hasil)
27                 delta_a.append(temp_d)
28             for x in range(len(temp_d)):
29                 hasil = (alfa[-1][x]) + (delta_a[-1][x])
30                 temp_a.append(hasil)
31                 alfa.append(temp_a)
32
33         return E[-1], delta a[-1], alfa[-1]

```

Kode Program 0.15 Implementasi perhitungan *sequential learning*

Kode program diatas menunjukkan implementasi perhitungan *sequential learning* yang dilakukan oleh algoritme *Support Vector Machine*. Hasil yang didapatkan pada kode program diatas adalah nilai *error rate*, *delta alfa*, dan nilai *alfa*. Penjelasan dari Kode Program 5.15 diuraikan pada keterangan berikut:

1. Baris 9-10 adalah langkah untuk memberikan nilai awal alfa pada setiap dokumen.
2. Baris 13-33 menjelaskan mengenai perhitungann *sequential learning* yang dilakukan hingga iterasi maksimum yang diinginkan. Dalam satu kali iterasi akan menghitung nilai *error rate*, *delta alfa*, dan nilai *alfa*. Untuk menghitung nilai *error rate* dilakukan operasi perkalian antara matriks hessian dengan nilai alfa. Kemudian dilakukan perhitungan delta alfa dengan mencari nilai minimum dan maksimum seperti yang ditunjukkan pada persamaan 2.14. Nilai delta alfa yang telah didapatkan berfungsi dalam memperbarui nilai alfa. Pencarian nilai error rate, delta alfa dan nilai alfa dilakukan hingga iterasi maksimum yang telah ditentukan.

5.5.5 Perhitungan Bias

Perhitungan bias merupakan tahapan yang dilakukan setelah nilai *support vector* telah didapatkan. Proses ini bertujuan dalam pembentukan *hyperplane* berdasarkan nilai *support vector* dari masing-masing sentimen. Proses implementasi dari perhitungan bias ditunjukkan pada Kode Program 5.16.

```

1  def bias(self, matrix, sv, kelas, d):
2      c = 1
3      xpos = xneg = 0
4
5      for i in range(len(kelas)):
6          if kelas[i] == -1:
7              xneg = max(xneg, sv[i])
8          elif kelas[i] == 1:
9              xpos = max(xpos, sv[i])
10         alpos = sv.index(xpos)
11         alneg = sv.index(xneg)
12         sigmapos = sigmaneg = 0
13
14         for i in range(len(kelas)):
15             totpos = totneg = 0
16             for j in range(len(matrix[i])):
17                 totpos = totpos + matrix[i][j] * matrix[alpos][j]
18                 totneg = totneg + matrix[i][j] * matrix[alneg][j]
19             kernelpos = math.pow((totpos + c), d)
20             kernelneg = math.pow((totneg + c), d)
21             h_pos = sv[i]*kelas[i]*kernelpos
22             h_neg = sv[i]*kelas[i]*kernelneg
23             sigmapos+= h_pos
24             sigmaneg+= h_neg
25         bias = -0.5 * (sigmapos+sigmaneg)
26         return bias

```

Kode Program 0.16 Implementasi perhitungan bias

Perhitungan bias yang ditunjukkan oleh Kode Program 5.16 dapat diuraikan pada penjelasan berikut:

1. Baris 5-12 adalah langkah dalam menentukan dokumen yang memiliki nilai support vector tertinggi pada kelas positif dan kelas negatif.

- Baris 14-25 merupakan langkah dalam mencari nilai bias. Pencarian nilai bias dilakukan dengan mendapatkan nilai alfa terbaik pada setiap kelas yang kemudian kernel dokumen yang memiliki nilai terbaik akan dikalikan dengan setiap kelas dan nilai alfa. Pencarian nilai bias terdapat pada Persamaan 2.17. Hal ini dilakukan pada setiap kelas positif dan negatif, yang kemudian hasilnya akan dijumlahkan. Untuk mendapatkan nilai bias, total nilai dari kelas positif dan negatif dikalikan dengan 0.5.

5.5.6 Perhitungan *Testing*

Tahapan akhir dalam klasifikasi dengan *Support Vector Machine* adalah melakukan data uji untuk menentukan kelas sentimen dari setiap dokumen. Implementasi pada perhitungan testing ditunjukkan pada Kode Program 5.17.

```

1  def datatesting(self, kelas, matrix, trans, d, alfa, bias):
2      matrix_tes = matrix
3      c = 1
4      hasiltes = []
5      hasilkelas = []
6      for i in range(len(matrix_tes[0])):
7          temp = []
8          for j in range(len(kelas)):
9              total = 0
10             for x in range(len(matrix_tes[i])):
11                 total = total + ( trans[i][x]
12 *matrix_tes[x][j])
13                 kernel = math.pow((total+c), d)
14                 hasil = alfa[j] * kelas[j] * (kernel)
15                 temp.append(hasil)
16             jumlah = sum(temp)+bias
17             hasiltes.append(jumlah)
18             if jumlah > 0:
19                 klas = 1
20             else:
21                 klas = -1
22             hasilkelas.append(klas)
23         return hasiltes, hasilkelas

```

Kode Program 0.17 Implementasi perhitungan *testing*

Kode Program 5.17 merupakan langkah untuk menentukan perhitungan data uji. Langkah dalam menjalankan program diuraikan pada penjelasan berikut:

- Baris 6-22 merupakan langkah perhitungan data uji dimana telah dibentuk dalam matriks dan telah terdapat transposisi matriks. Untuk menghitung data uji akan dicari kernel dari data uji tersebut yang kemudian dikalikan dengan kelas dari setiap dokumen data latih dan dikalikan dengan nilai alfa dari setiap dokumen data latih.

5.6 Evaluasi

Tahapan akhir yang dilakukan oleh sistem yaitu proses evaluasi atau pengujian. Dilakukannya evaluasi bertujuan untuk mengukur keakuratan sistem dengan metode yang diterapkan dalam menyelesaikan masalah. Proses dari evaluasi ialah

dengan menghitung *confusion matrix*, nilai akurasi, *precision*, *recall*, dan *f-measure*.

5.6.1 Perhitungan *Confusion Matrix*

Tahapan evaluasi dimulai dengan mencari hasil *confusion matrix*. Tahapan ini dilakukan untuk menghitung jumlah data uji yang benar maupun data yang salah dalam proses klasifikasi. Implementasi pada tahapan ini ditunjukkan pada Kode Program 5.18.

```

1  def conf_matrix(self, kelas, testing_kelas):
2      conf = [[0,0],[0,0]]
3
4      for i in range(len(kelas)):
5          if kelas[i] == 1 and testing_kelas[i] == 1:
6              conf[0][0] += 1
7          elif kelas[i] == 1 and testing_kelas[i] == -1:
8              conf[0][1] += 1
9          elif kelas[i] == -1 and testing_kelas[i] == 1:
10             conf[1][0] += 1
11             elif kelas[i] == -1 and testing_kelas[i] == -1:
12                 conf[1][1] += 1
13
14         return conf

```

Kode Program 0.18 Implementasi *confusion matrix*

Kode Program 5.18 menunjukkan proses perhitungan *confusion matrix* yang digunakan sistem. Sesuai dengan kode program diatas dapat diuraikan pada penjelasan berikut:

1. Baris 2 menunjukkan nilai list awal pada *confusion matrix*.
2. Baris 4-12 merupakan perulangan dimana akan memeriksa setiap kelas latih dengan hasil kelas uji. Jika pada data latih menunjukkan kelas latihnya adalah satu dan pada hasil pengujian, kelas uji menunjukkan nilai satu, maka counter pada list baris satu, kolom satu akan bertambah satu. Karena pada list tersebut merupakan penempatan untuk nilai *true positive*. Pada baris kesatu, kolom kedua menunjukkan nilai *false negatif*. Pada baris kedua, kolom kesatu menunjukkan nilai *false positive*. Pada baris kedua, kolom kedua menunjukkan nilai *true negatif*.

5.6.2 Perhitungan Akurasi, Precision, Recall, dan F-Measure

Perhitungan akurasi, *precision*, *recall*, dan *F-Measure* merupakan perhitungan yang dilakukan untuk mengetahui kinerja dari sistem. Dimana perhitungan akurasi untuk mengetahui tingkat ketepatan sistem dalam menerapkan algoritme *Support Vector Machine*. Fungsi penerapan perhitungan *precision* dilakukan untuk mengetahui ketepatan sistem dalam melakukan proses klasifikasi. Perhitungan *recall* berfungsi untuk mengetahui tingkat keberhasilan sistem. Perhitungan *f-measure* merupakan pengujian kombinasi dari tingkat *precision* dan tingkat *recall*. Tahapan ini dilakukan setelah dilakukan tahapan perhitungan *confusion matrix*. Implementasi pada tahapan ini ditunjukkan pada Kode Program 5.19.


```

1  def pengujian(self, conf):
2      tp = conf[0][0]
3      fn = conf[0][1]
4      fp = conf[1][0]
5      tn = conf[1][1]
6      accuracy = (tn + tp) / (tn+tp+fn+fp)
7      precision = tp / (tp+fp)
8      recall = tp / (tp+fn)
9      fmeas = (2*precision*recall) / (precision+recall)
10
11     return accuracy, precision, recall, fmeas

```

Kode Program 0.19 Implementasi perhitungan akurasi

Kode program diatas menunjukkan perhitungan akurasi dengan menggunakan nilai *confusion matrix* yang didapatkan pada method sebelumnya. Penjelasan kode program diatas, diuraikan pada keterangan berikut:

1. Baris 2-5 mendeklarasikan nilai tp, fn, fp, dan tn pada *confusion matrix*.
2. Baris 6 menghitung nilai akurasi dengan menjumlahkan hasil tn dan tp dan dibagi dengan nilai keseluruhan pada *confusion matrix*.
3. Baris 7 merupakan perhitungan nilai *precision* dengan nilai tp sebagai pembilang dan menjumlahkan nilai tp dan fp sebagai penyebut
4. Baris 8 adalah baris program dalam menghitung nilai *recall* dengan nilai tp sebagai pembilang dan jumlah tp dan fn sebagai penyebut
5. Baris 9 menunjukkan perhitungan akurasi *f-measure* dengan mengkalikan nilai *precision*, *recall*, dan bilangan 2 sebagai pembilang dan jumlah nilai dari *precision* dan *recall* sebagai penyebut.

5.6.3 Perhitungan Waktu Komputasi

```

1  def start_time(self):
2
3      start = time.clock()
4      return start
5
6  def komputasi(self, start):
7
8      komputasi= time.clock()-start
9      return komputasi

```

Kode Program 0.20 Implementasi perhitungan waktu komputasi

Kode Program 5.20 menunjukkan waktu komputasi yang dibutuhkan sistem untuk menyelesaikan algoritme. Penjelasan dalam potongan program diatas, diuraikan pada penjelasan berikut:

1. Baris 1-4 adalah kode program untuk mendefinisikan waktu awal yang dicatat ketika sistem dijalankan
2. Baris 6-9 merupakan kode program yang mendefinisikan waktu akhir ketika sistem telah menyelesaikan algoritme. Untuk menyimpan durasi sistem dalam menyelesaikan algoritme, disimpan pada variabel komputasi.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang proses pengujian yang dilakukan pada sistem yang telah dibuat beserta analisis yang dilakukan berdasarkan hasil pengujian. Terdapat beberapa pengujian yang dilakukan pada sistem yaitu pengujian terhadap pengaruh parameter *Support Vector Machine* dan pengujian ketika diterapkannya *Lexicon Based Features*.

6.1 Pengujian Pengaruh Parameter *Support Vector Machine*

Pengujian yang dilakukan merupakan pengujian untuk mengetahui pengaruh parameter yang ada pada metode *Support Vector Machine*. Parameter yang akan diuji adalah nilai *degree* dan *learning rate* yang ada pada kernel polynomial. Pada pengujian parameter nilai *degree* pada kernel polynomial akan dicari nilai akurasi paling tertinggi dari setiap nilai *degree* yang diuji coba. Kemudian pengujian pada parameter *learning rate* akan dicari nilai akurasi tertinggi pada nilai *learning rate* tertentu namun dilihat pula dari iterasi yang dilakukan. Pengujian lainnya adalah perhitungan waktu komputasi yang diperlukan oleh sistem berdasarkan iterasi maksimum. Proses pengujian pada pengaruh parameter *Support Vector Machine* diuraikan pada skenario ujian berikut.

6.1.1 Skenario Pengujian Pengaruh Parameter *Support Vector Machine*

Pengujian yang dilakukan pada parameter *Support Vector Machine* dilakukan dengan menguji nilai *degree* yang ada pada perhitungan kernel polynomial dan menguji nilai *learning rate*. Pengujian yang dilakukan dengan mempertimbangkan jumlah iterasi maksimum yang dilakukan oleh sistem pada proses *sequential learning Support Vector Machine*. Dalam memulai pengujian digunakan 400 data komentar Instagram dengan perbandingan untuk data latih dan data uji yang digunakan adalah 70% sebagai data latih dan 30% sebagai data uji.

Untuk permulaan dalam melakukan pengujian, dilakukan pengujian pada pengaruh nilai *degree* yang digunakan pada perhitungan kernel polynomial. Nilai *learning rate* yang digunakan pada pengujian ini adalah 0,0001 dan iterasi maksimum yang dilakukan adalah 200 kali. Pengujian dilakukan sebanyak lima kali dengan dengan lima nilai *degree* yang berbeda.

Pengujian waktu komputasi sistem dilakukan ketika proses pengujian nilai *degree* dan *learning rate* telah selesai dilakukan. Parameter nilai *degree* dan *learning rate* yang digunakan pada proses waktu komputasi, merupakan nilai parameter terbaik. Hasil pengujian yang telah dilakukan ditunjukkan pada Tabel 6.1 dan 6.2.

Tabel 0.1 Hasil pengujian pengaruh nilai *degree*

Evaluasi	Kernel Polynomial (Nilai <i>degree</i>)				
	2	3	4	5	6
<i>Accuracy</i>	80%	52%	50%	50%	50%
<i>Precision</i>	86%	51%	50%	50%	50%
<i>Recall</i>	71.67%	100%	100%	100%	100%
<i>F-Measure</i>	78.18%	67%	67%	67%	67%

Tabel 0.2 Hasil pengujian konstanta *learning rate*

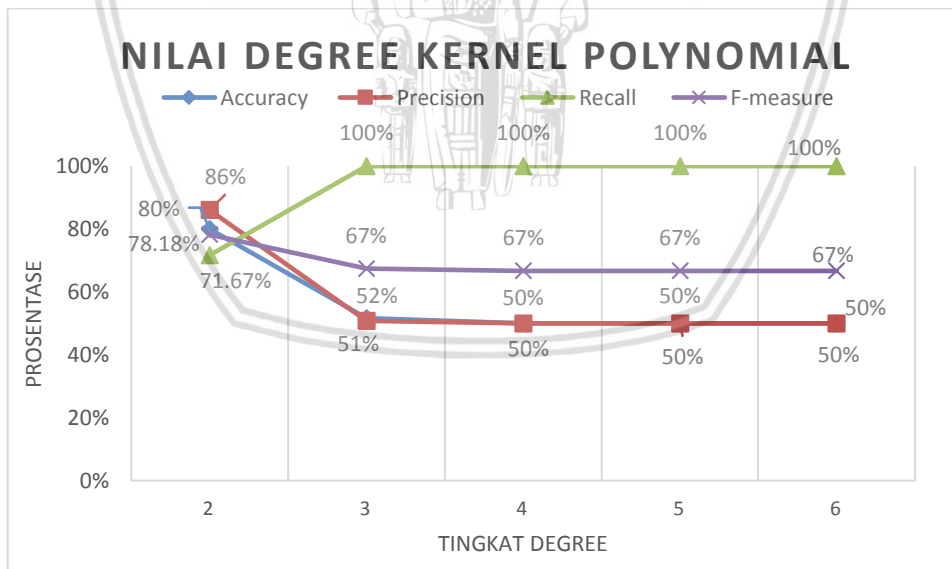
Iterasi		Learning rate γ					Rata-Rata	Waktu Komputasi (d)
		0,0001	0,0005	0,001	0,0025	0,05		
100	<i>Accuracy</i>	82.50%	75.00%	70.00%	50%	50%	65.50%	254.61
	<i>Precision</i>	95.35%	89.47%	96.15%	50%	50%	76.20%	
	<i>Recall</i>	68.33%	56.67%	41.67%	100%	100%	73.33%	
	<i>F-Measure</i>	79.61%	69.39%	58.14%	66.67%	66.67%	68.09%	
200	<i>Accuracy</i>	82.50%	79.17%	71.67%	50%	50%	66.67%	267.02
	<i>Precision</i>	86.79%	76.92%	75.00%	50%	50%	67.74%	
	<i>Recall</i>	76.67%	83.33%	65.00%	100%	100%	85.00%	
	<i>F-Measure</i>	81.42%	80.00%	69.64%	66.67%	66.67%	72.88%	
300	<i>Accuracy</i>	76.67%	65.00%	70.83%	50%	50%	62.50%	293.54
	<i>Precision</i>	97%	95%	93.10%	50%	50%	77.03%	
	<i>Recall</i>	55%	31.67%	45%	100%	100%	66.33%	
	<i>F-Measure</i>	70.21%	47.50%	60.67%	66.67%	66.67%	62.34%	
400	<i>Accuracy</i>	75.83%	67.50%	71.67%	50%	50%	63.00%	310.18
	<i>Precision</i>	77.19%	76.92%	70.31%	50%	50%	64.89%	
	<i>Recall</i>	73.33%	50.00%	75.00%	100%	100%	79.67%	
	<i>F-Measure</i>	75.21%	60.61%	72.58%	66.67%	66.67%	68.35%	
Rata-rata	<i>Accuracy</i>	79.38%	71.67%	71.04%	50%	50%		
	<i>Precision</i>	89.10%	85%	83.64%	50%	50%		
	<i>Recall</i>	68.33%	55.42%	57%	100%	100%		
	<i>F-Measure</i>	76.61%	64.37%	65.26%	67%	66.67%		



6.1.2 Analisis Hasil Pengujian Pengaruh Parameter *Support Vector Machine*

Pengujian yang telah dilakukan pada parameter *Support Vector Machine* yang menggunakan komposisi data berupa 70% data latih dan 30% data uji terhadap 400 data komentar Instagram ditunjukkan pada Tabel 6.1. Pada tabel tersebut menunjukkan bahwa mulanya yang diuji pada parameter *Support Vector Machine* adalah nilai *degree*, dimana yang digunakan ialah nilai *degree* 2, 3, 4, 5, dan 6.

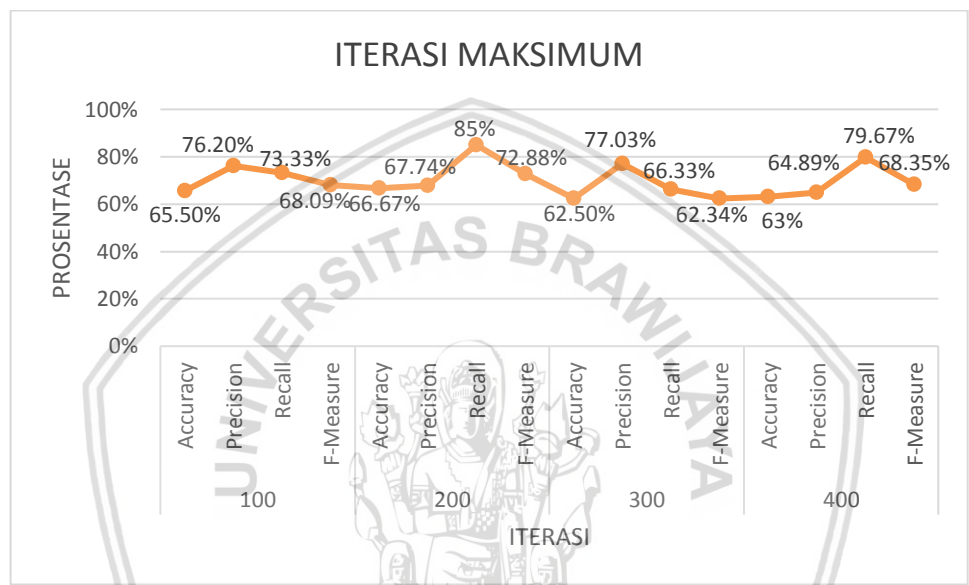
Pada pengujian tersebut digunakan data latih dan data uji yang sama untuk setiap nilai *degree* pada kernel. Didapatkan nilai pengujian yang setara pada setiap nilai *degree*. Pada Gambar 6.1 ditunjukkan nilai akurasi terbaik terletak pada nilai *degree* = 2, yaitu 80% dan diikuti oleh nilai *precision*, *recall*, dan *f-measure*. Nilai akurasi pada *degree* 3 hingga 6 cenderung konstan, yang mana nilai akurasinya menunjukkan pada prosentase 50%. Nilai *recall* yang didapatkan pada *degree* 3 hingga 6 dapat mencapai 100% menunjukkan bahwa sistem telah berjalan efektif, dikarenakan hasil yang diberikan sistem terhadap data yang relevan lebih besar atau seimbang. Namun berdasarkan hasil pengujian dapat disimpulkan bahwa nilai *degree* yang paling optimal ketika pada *degree* 2 dengan memperoleh tingkat akurasi tertinggi sebesar 80%. Meningkatnya nilai *degree* pada kernel polynomial berpengaruh terhadap hasil perhitungan matriks Hessian yang berfungsi dalam mencari nilai optimum pada setiap dokumen data, dimana hasil dari matriks Hessian digunakan untuk menghitung besar nilai *error rate* pada setiap dokumen dan berpengaruh terhadap pembentukan support vector.



Gambar 0.1 Grafik hasil pengujian pengaruh nilai *degree*

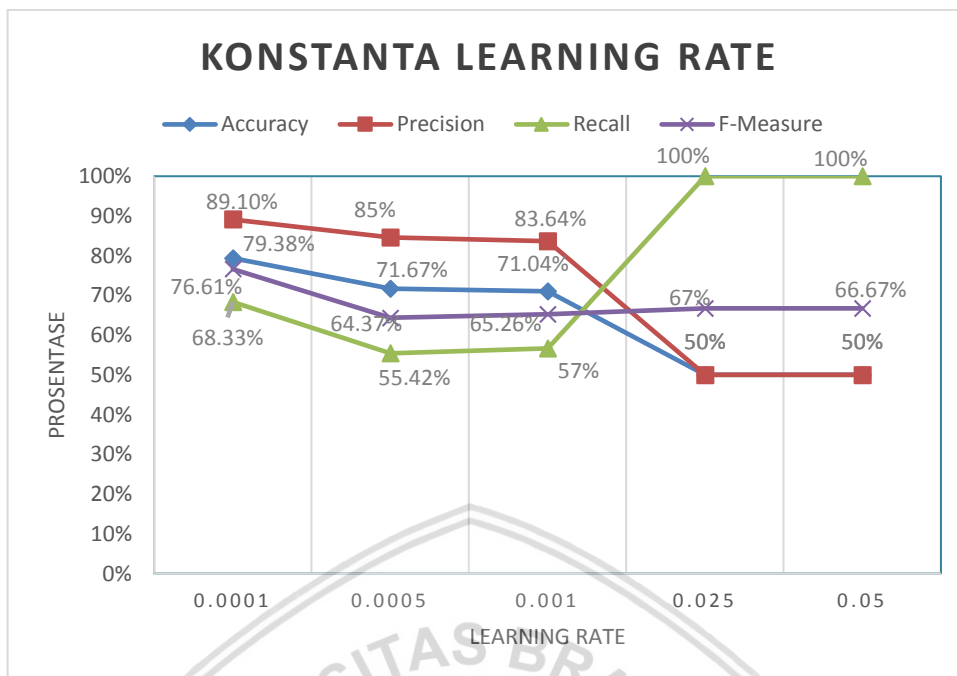
Dari pengujian iterasi maksimum didapatkan bahwa nilai akurasi paling baik ketika iterasi maksimum sebanyak 200 kali. Hal ini ditunjukkan oleh Gambar 6.2, dimana iterasi maksimum 200 memiliki tingkat akurasi sebesar 66,67% dan akurasi mengalami penurunan ketika iterasi maksimumnya adalah 300 kali dengan tingkat akurasi 62,50%. Namun peningkatan jumlah iterasi secara terus-menerus tidak

menandakan perbaikan akurasi, melainkan menurunkan tingkat akurasi. Penurunan tingkat akurasi terjadi karena pada tahapan *sequential learning* akan mengalami perubahan nilai α_i . Perubahan nilai α_i yang berpengaruh pada penurunan tingkat akurasi ini karena nilai α_i yang menjadi tidak konvergen yang dapat dibuktikan dengan perubahan nilai α_i . Nilai dari α_i tersebut menjadi pembentukan *support vector*. Nilai *support vector* berfungsi sebagai titik optimum data terdekat dengan garis *hyperplane* dalam menentukan setiap kelas data. Penentuan nilai *support vector* dipengaruhi oleh jumlah iterasi maksimum yang dilakukan oleh sistem. Dengan peningkatan jumlah iterasi, tidak menjamin nilai *support vector* yang diberikan merupakan batas terbaik antara kelas data.



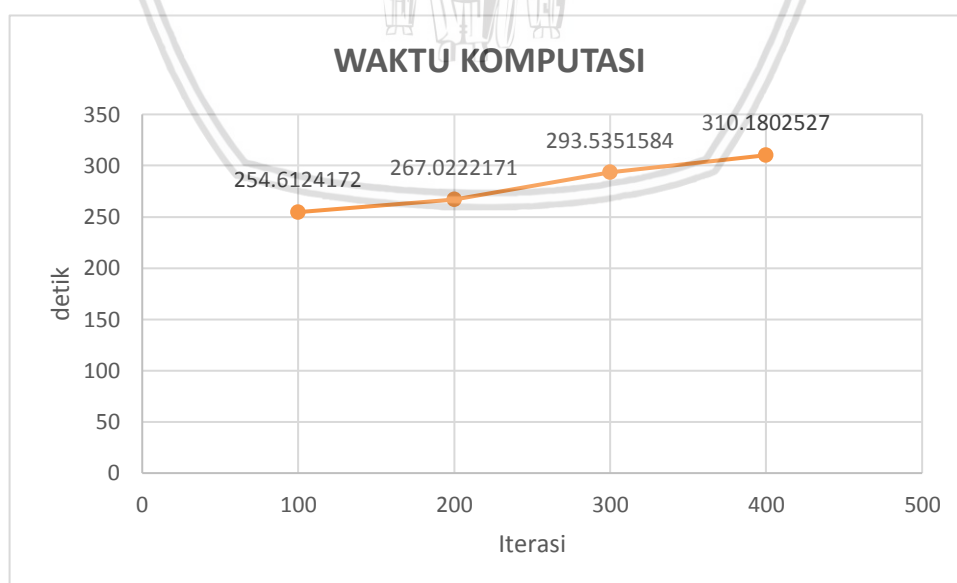
Gambar 0.2 Grafik hasil pengujian iterasi maksimum

Pengujian yang telah dilakukan setelahnya adalah pengujian terhadap nilai konstanta *learning rate*. Pada pengujian yang telah dilakukan dan ditunjukkan pada Tabel 6.2 didapatkan grafik hasil yang dapat dilihat pada Gambar 6.3. Hasil pengujian menunjukkan nilai konstanta *learning rate* terbaik ketika 0,0001. Hal ini ditunjukkan dengan tingkat akurasi yang mencapai 79,38%. Konstanta *learning rate* berfungsi untuk mengontrol kecepatan pada proses *training* dan bergantung pada jumlah iterasi untuk mencapai konvergensi. Dari hasil pengujian ditunjukkan bahwa nilai optimal konstanta *learning rate* adalah 0,0001 dan terjadi ketika iterasi maksimum mencapai 200 kali. Perubahan nilai *learning rate* pada sistem memengaruhi nilai $\delta\alpha$. Karena *learning rate* menjadi salah satu kandidat nilai yang berpengaruh dalam pembentukan nilai alfa dan pembentukan himpunan *support vector*. Semakin meningkatnya nilai *learning rate* berdampak pada proses pelatihan yang semakin cepat, sehingga tingkat ketelitian menjadi berkurang.



Gambar 0.3 Grafik hasil pengujian konstanta *learning rate*

Pengujian waktu komputasi (*running time*) pada sistem dilakukan terhadap iterasi maksimum yang dilakukan. Pada tahapan ini digunakan komposisi data sebesar 70% data latih dan 30% data uji. Seiring dengan bertambahnya iterasi maksimum, maka akan meningkatkan waktu komputasi yang direpresentasikan dalam satuan waktu detik. Ditunjukkan pada Gambar 6.4 bahwa waktu komputasi terbaik sistem dalam menyelesaikan klasifikasi dengan menggunakan 100 iterasi maksimum. Waktu yang dibutuhkan yaitu selama 254,6124172 detik.



Gambar 0.4 Grafik hasil pengujian pengaruh iterasi maksimum terhadap waktu komputasi sistem



6.2 Pengujian Pengaruh Implementasi *Lexicon Based Features*

Pada tahapan pengujian pengaruh implementasi *Lexicon Based Features* bertujuan untuk mencari pengaruh implementasi *Lexicon Based Features* terhadap tingkat akurasi sistem. Tahapan pengujian ini akan membandingkan sistem ketika *Lexicon Based Features* diimplementasikan dan ketika *Lexicon Based Features* tidak diimplementasikan. Pengujian dari implementasi *Lexicon Based Features* akan mempertimbangkan perbandingan data latih dan data uji untuk diketahui nilai akurasi, *precision*, *recall*, dan *f-measure*. Untuk memperjelas pengujian pada implementasi *Lexicon Based Features*, dapat dilihat pada skenario pengujian yang telah diuraikan.

6.2.1 Skenario Pengujian Pengaruh Implementasi *Lexicon Based Features*

Dalam melakukan pengujian pada tahapan ini, akan mempertimbangkan komposisi antara data latih dan data uji. Komposisi dari perbandingan antara data latih dan data uji dimulai dengan 50% data latih dan 50% data uji. Pengujian untuk perbandingan data latih dan data uji dilakukan hingga komposisi data mencapai 90% data latih dan 10% data uji.

Untuk memulai pengujian pada tahapan ini, nilai *degree* yang digunakan adalah 2 dan nilai dari *learning rate* adalah 0.0001. Hasil pengujian dapat dilihat pada tabel 6.3.

Tabel 0.3 Hasil pengujian pengaruh implementasi *Lexicon Based Features* dengan Normalisasi *Min-Max* dan *Lexicon Based Features* dengan skor sentimen

Perbandingan data latih : data uji		Tanpa <i>Lexicon Based Features</i>	<i>Lexicon Based Features</i> dengan normalisasi <i>min-max</i>	<i>Lexicon Based Features</i> dengan skor sentimen
50:50	<i>Accuracy</i>	90%	87.0%	85.5%
	<i>Precision</i>	94.44%	91.11%	84.47%
	<i>Recall</i>	85%	82%	87%
	<i>F-Measure</i>	89.47%	86.32%	85.71%
60:40	<i>Accuracy</i>	82.5%	81.25%	81.25%
	<i>Precision</i>	79.55%	89.06%	84.72%
	<i>Recall</i>	87.5%	71.25%	76.25%
	<i>F-Measure</i>	83.33%	79.17%	80.26%
70:30	<i>Accuracy</i>	82.5%	79.17%	75.83%
	<i>Precision</i>	86.79%	92.68%	75.41%

Tabel 6.3 Hasil pengujian pengaruh implementasi *Lexicon Based Features* dengan Normalisasi *Min-Max* dan *Lexicon Based Features* dengan skor sentimen (Lanjutan)

Perbandingan data latih : data uji		Tanpa <i>Lexicon Based Features</i>	<i>Lexicon Based Features</i> dengan normalisasi <i>min-max</i>	<i>Lexicon Based Features</i> dengan skor sentimen
	<i>Recall</i>	76.67%	63.33%	76.67%
	<i>F-Measure</i>	81.42%	75.25%	76.03%
80:20	<i>Accuracy</i>	75%	70%	71.25%
	<i>Precision</i>	88.46%	71.05%	68.09%
	<i>Recall</i>	57.5%	68%	80%
	<i>F-Measure</i>	69.7%	69%	73.56%
90:10	<i>Accuracy</i>	65%	60%	55%
	<i>Precision</i>	100%	66.67%	100%
	<i>Recall</i>	30%	40%	100%
	<i>F-Measure</i>	46.15%	50%	18.18%
Rata-rata	<i>Accuracy</i>	79%	75.5%	73.8%
	<i>Precision</i>	89.85%	82.12%	82.54%
	<i>Recall</i>	67.33%	64.82%	83.98%
	<i>F-Measure</i>	74.01%	72.0%	66.8%

6.2.2 Analisis Hasil Pengujian Pengaruh Penerapan *Lexicon Based Features*

Hasil pengujian perbandingan implementasi *Lexicon Based Features* dengan normalisasi *min-max*, *Lexicon Based Features* dengan perhitungan skor sentimen, dan tanpa mengimplementasikan *Lexicon Based Features* pada sistem telah ditunjukkan pada Tabel 6.4 dan direpresentasikan pada bentuk grafik pada Gambar 6.5. Dari hasil pengujian diketahui bahwa proses klasifikasi yang dilakukan tanpa mengimplementasikan metode *Lexicon Based Features* memiliki tingkat akurasi yang lebih baik dibandingkan dengan proses klasifikasi yang mengimplementasikan metode *Lexicon Based Features*. Secara umum pengaruh implementasi dari metode *Lexicon Based Features* yaitu pada penggunaan kamus lexicon yang masih umum (tidak terfokus pada *cyberbullying*) dan data yang digunakan merupakan data yang bersifat variatif dan kompleks. Kamus lexicon yang digunakan hanyalah berbentuk kata, namun dalam mengenali sifat sentimen

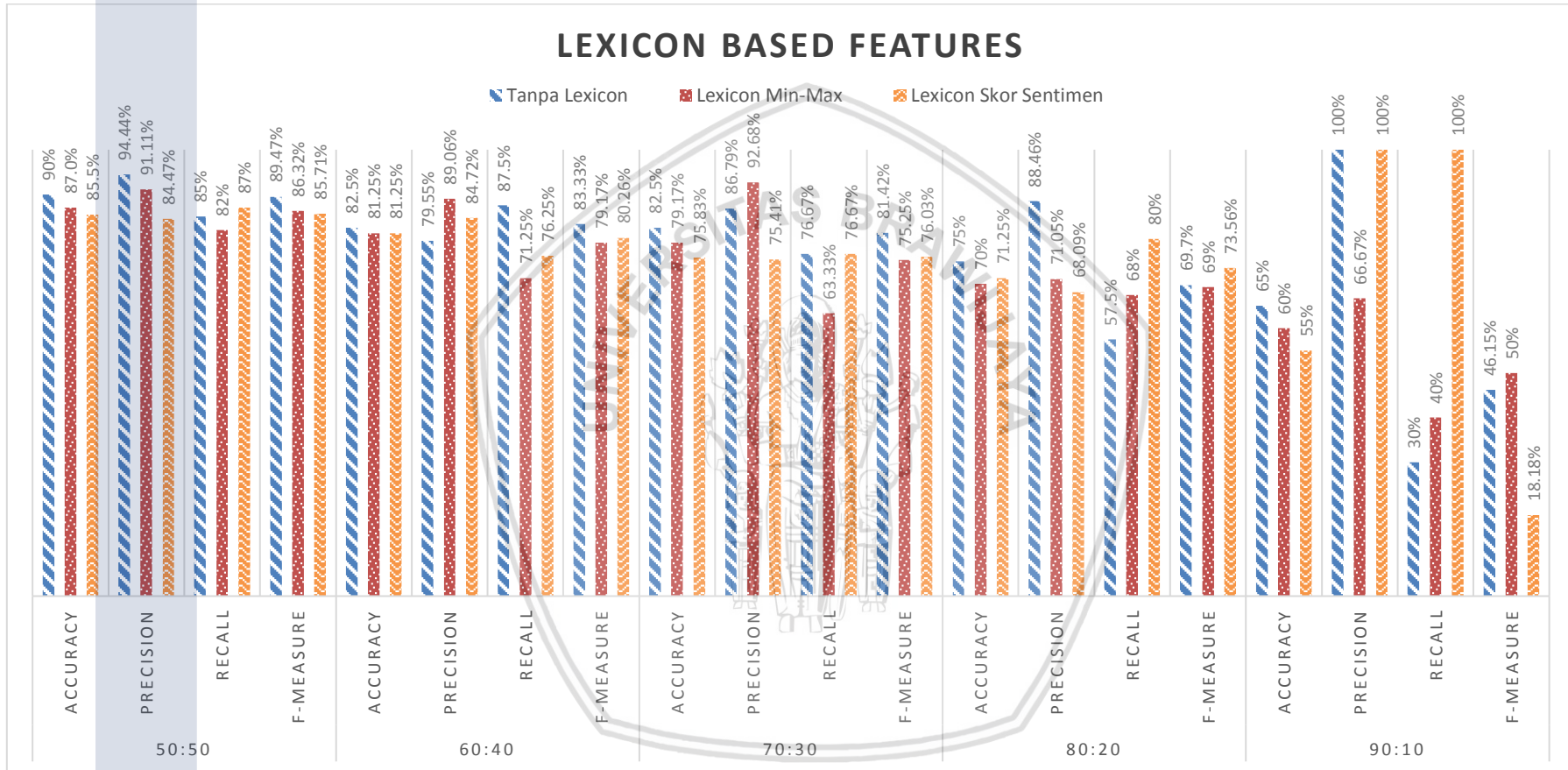


dari kalimat *cyberbullying* dibutuhkan kamus yang lebih spesifik yaitu kamus yang terdapat bentuk frase. Prosentase akurasi terbaik mencapai 90% pada sistem yang tidak mengimplementasikan *Lexicon Based Features*.

Didapatkan bahwa rata-rata akurasi *Lexicon Based Features* dengan normalisasi *min-max* lebih baik dibandingkan dengan *Lexicon Based Features* dengan perhitungan skor sentimen. Pada satu komposisi data (60% data latih :40% data uji), akurasi yang didapatkan pada kedua cara implementasi *Lexicon Based Features* tidak terdapat perbedaan. Namun ketika pengujian yang dilakukan pada komposisi data latih dan data uji lainnya (50% data latih: 50% data uji , 70% data latih: 30% data uji, 80% data latih: 20% data uji, 90% data latih: 10% data uji) memberikan tingkat akurasi yang cukup berbeda. Penyebab perbedaan yang terjadi dikarenakan hasil yang didapatkan ketika menghitung matriks Hessian. Pengaruh perbedaan hasil yang signifikan dari perhitungan matriks Hessian dapat memberikan dampak dalam pembentukan *support vector*.

Selain pengujian implementasi *Lexicon Based Features* dan tanpa *Lexicon Based Features*, dapat dilihat bahwa terdapat pengujian pada komposisi data latih dan data uji. Tingkat akurasi terbaik sebesar 90% yang didapatkan pada 50% komposisi data latih dan 50% komposisi data uji pada sistem yang tidak mengimplementasikan algoritme *Lexicon Based Features*. Namun seiring dengan bertambahnya komposisi data latih, tingkat akurasi semakin menurun. Hal ini terjadi karena adanya *over-fitting*, yaitu pada proses *training* (pelatihan) data telah dimodelkan dengan sangat baik, sehingga memungkinkan *noise* data telah dipelajari dan menyebabkan data uji tidak dapat diprediksi dengan baik.

Selain itu terdapat perubahan nilai *precision* dan *recall* pada pengujian komposisi data latih dan data uji. Perubahan nilai *precision* disebabkan oleh berubahnya jumlah dokumen yang diprediksi sebagai kelas positif namun memiliki kelas aktual negatif, sehingga sistem tidak memanggil kembali dokumen yang dianggap tidak sesuai dengan keinginan. Meningkatnya nilai *precision* dikarenakan rendahnya jumlah prediksi kelas data yang tidak sesuai dengan kelas aktual. Dan meningkatnya nilai *recall* dikarenakan jumlah dokumen yang ditemukan oleh sistem dianggap telah sesuai dengan yang diinginkan. Peningkatan nilai *recall* menunjukkan bahwa sistem telah berjalan dengan efektif.



Gambar 0.5 Grafik hasil pengujian perbandingan pengaruh penerapan *lexicon based features* dan tanpa *lexicon based features*

BAB 7 PENUTUP

Pada bab penutup akan menguraikan mengenai kesimpulan yang didapatkan ketika penelitian dilakukan hingga didapatkan suatu hasil penelitian serta memberikan saran-saran yang dapat dikembangkan pada penelitian selanjutnya.

7.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan dalam analisis sentimen *cyberbullying* pada komentar Instagram yang menerapkan metode *Support Vector Machine*. Hal yang disimpulkan dari penelitian ini adalah sebagai berikut:

1. Metode klasifikasi *Support Vector Machine* dapat digunakan dalam menganalisis sentimen *cyberbullying* pada kolom komentar Instagram. Hasil klasifikasi berupa kelas positif dan negatif yang dibedakan menjadi sentimen positif *cyberbullying* dan sentimen negatif *cyberbullying*. Dalam melakukan penelitian digunakan 400 data komentar Instagram, yang terdiri dari 200 komentar positif *cyberbullying* dan 200 negatif Instagram. Dalam penelitian digunakan komposisi data sebesar 70% pada data latih dan 30% data uji, yaitu 280 data sebagai data latih dan 120 data sebagai data uji. Dalam 280 data latih tersebut masih dikomposisikan 140 data dari kelas bersentimen positif *cyberbullying* dan 140 data dari kelas negatif *cyberbullying*. Proses yang dilakukan untuk melakukan analisis sentimen pada penelitian ini mulanya melakukan ekstraksi teks dengan melalui tahapan *Pre-processing*, kemudian pembobotan TF-IDF, dan perhitungan klasifikasi teks dengan metode *Support Vector Machine*. Hasil klasifikasi yang diberikan dapat berupa kelas sentimen positif *cyberbullying* atau negatif *cyberbullying*.
2. Berdasarkan hasil pengujian yang dilakukan didapatkan tingkat akurasi terbaik sebesar 90%, *precision* sebesar 94,44%, 85% *recall* sebesar dan *f-measure* sebesar 89,47%. Tingkat akurasi tersebut didapatkan dengan komposisi data 50% data latih dan 50% data uji dan tanpa mengimplementasikan algoritme *Lexicon Based Features*. Selain itu parameter-parameter yang digunakan pada algoritme *Support Vector Machine* memengaruhi tingkat akurasi yang didapatkan. Nilai *degree* yang digunakan pada proses perhitungan kernel polynomial memiliki titik optimal ketika bernilai 2. Selain itu konstanta *learning rate* yang digunakan sebesar 0,0001 dan iterasi maksimum yang baik yaitu sebesar 200 kali. Perubahan parameter-parameter Support Vector tentunya memengaruhi hasil akurasi klasifikasi, seperti halnya dengan peningkatan nilai *degree* akan menurunkan tingkat akurasi.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan, masih terdapat beberapa kekurangan yang perlu diperbaiki maupun dikembangkan dari penelitian ini. Saran yang diberikan untuk dilakukan pada penelitian berikutnya adalah:

1. Data yang digunakan untuk proses klasifikasi didapatkan secara *real time* yang kemudian dapat dimasukkan sebagai tambahan data untuk data latih baik yang bersentimen positif *cyberbullying* maupun yang bersentimen negatif *cyberbullying*.
2. Data diklasifikasikan menjadi tiga kelas sentimen, yaitu sentimen positif *cyberbullying*, netral, dan negatif *cyberbullying*. Hasil klasifikasi dari ketiga kelas tersebut dapat diketahui polaritas dari setiap sentimennya.
3. Dapat diimplementasikan suatu metode optimasi dalam ekstraksi fitur yang berguna dalam mengidentifikasi teks berdasarkan makna per kata, frase, dan kalimat.
4. Tahapan evaluasi sistem dapat mempertimbangkan konsep *macro average* dan *micro average* untuk mengevaluasi kinerja sistem, sehingga dapat meningkatkan kinerja dalam mengklasifikasi multi-label.
5. Sistem dapat dikembangkan menjadi suatu produk yang bersifat preventif bagi pengguna baik yang membaca kolom komentar maupun yang mengunggah suatu komentar.

DAFTAR PUSTAKA

- Adi, A. & Hidayat, A., 2017. *45 Juta Pengguna Instagram, Indonesia Pasar Terbesar di Asia*. [Online] Tersedia di: <<https://bisnis.tempo.co/read/news/2017/07/26/090894605/45-juta-pengguna-instagram-indonesia-pasar-terbesar-di-asia>> [Diakses 24 Agustus 2017].
- Akbari, M. I. H. A. D., Astri Novianty S.T., M. & Casi Setianingsih S.T., M., 2012. Analisis Sentimen Menggunakan Metode Learning Vector Quantization. *Telkom University*.
- Ardianto, 2011. *Komunikasi 2.0*. s.l.:s.n.
- Asian, J., 2007. Effective Techniques for Indonesian Text Retrieval.
- Basari, 2013. Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization. *Procedia Engineering*, Volume 53, pp. 453-462.
- Berry, M. & Kogan, J., 2010. Text Mining Application and Theory. In: Wiley: United Kingdom .
- Bohang, F. K., 2017. *Instagram Jadi Media "Cyber-Bullying" Nomor 1*. [Online] Tersedia di: <<http://tekno.kompas.com/read/2017/07/21/12520067/instagram-jadi-media-cyber-bullying-nomor-1>> [Diakses 16 Februari 2018].
- Browniee, J., 2016. *Overfitting and Underfitting With Machine Learning Algorithms*. [Online] Tersedia di: <<https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>> [Diakses 19 Februari 2018].
- Buntoro, G., Adji, T. & Purnamasari, A. E., 2014. Sentiment Analysis Twitter dengan Kombinasi Lexicon Based dan Double Propagation.
- Chrismanto, A. R. & Lukito, Y., 2017. Identifikasi Komentar Spam Pada Instagram. *LONTAR KOMPUTER*, Volume 8.
- Darma, I. M. B. S., 2017. Penerapan Sentimen Analisis Acara Televisi Pada Twitter Menggunakan Support Vector Machine dan Algoritma Genetika sebagai Metode Seleksi Fitur.
- Desai, M. & Mehta, M., 2016. Techniques for Sentiment Analysis of Twitter Data: A Comprehensive Survey. International Conference on Computing, Communication and Automation,.
- Han, J. & Kamber, M., 2006. *Data Mining Concept and Techniques*. San Fransisco: Morgan Kauffman.

Hasanah, U., 2016. Klasifikasi Kondisi Detak Jantung Berdasarkan Rekam Elektrokardiografi (EKG) Menggunakan Algoritma Support Vector Machine.

Hidayat, A. N., 2015. Analisis Sentimen Terhadap Wacana Politik Pada Media Masa Online Menggunakan Algoritma Support Vector Machine Dan Naive Bayes. *Jurnal Elektronik Sistim Informasi Dan Komputer*, Volume 1.

Hinduja, S. & J.Patchin, 2010. Bullying, Cyberbullying, and Suicide. *Archives of Suicides Research*, Volume 14, pp. 206-221.

Jiawei, H., Kamber, M. & Pei, J., 2012. *Data Mining: Concepts and Techniques Third Edition*. MA: Morgan Kaufmann.

Junaedi, H., Budiarto, H., Maryati, J. & Melani, Y., 2011. Data Transformation Pada Data Mining. *Prosiding Konferensi Nasional "Inovasi dalam Desain dan Teknologi" - IDEaTech*.

Kowalski, R. & Limber, S., 2007. Electronic bullying among middle schools students. *Journal of Adolescent Health*, Volume 41, pp. 22-30.

K, S. T. & Shetty, J., 2017. Sentiment Analysis of Product Reviews: A Review. *International Conference on Inventive Communication and Computational Technologies*.

Liu, B., 2012. Sentiment Analysis and Opinion Mining. In: Chicago: Morgan & Claypool Publisher.

Manalu, B. U., 2014. Analisis Sentimen Pada Twitter Menggunakan Text Mining.

Manning, C., Raghavan, P. & Schütze, H., 2009. *An Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Melville, P., Gryc, W. & D, R. L., 2011. Sentiment analysis of blogs by combining lexical knowledge with text classification. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1275-1284.

Muttya, A., 2017. *Kompas.com*. [Online] Tersedia di: <<http://entertainment.kompas.com/read/2017/04/28/060000410/gara-gara.tiket.bts.putri.uya.kuya.dapat.ancaman>> [Diakses 23 Agustus 2017].

Nahar, V., Unankard, S., Li, X. & Pang, C., 2012. Sentiment Analysis for Effective Detection of Cyber Bullying. *The Australian E-Health Research Center*, Volume 767-744.

Nasrullah, R., 2015. Perundungan Siber (Cyber-Bullying) di Status Facebook Divisi Humas Mabes Polri. *Jurnal Sositologi*, Volume 14.

Nugroho, G. A. P., 2016. Analisis Sentimen Data Twitter Menggunakan K-Means Clustering.

Nurfalah, A., Adiwijaya & Suryani, A. A., 2017. Analisis Sentimen Berbahasa Indonesia dengan Pendekatan Lexicon Based pada Media Sosial Twitter. *Jurnal Masyarakat Informatika Indonesia*, Volume 2, pp. 1-8.

Paramita, 2008. Penerapan Support Vector Machine untuk Ekstraksi Informasi dari Dokumen Teks. *Laporan Tugas Akhir , Program Studi Teknik Informatika, STEI Institut Teknologi Bandung*.

Parikh, R. & M.M, 2009. Sentiment Analysis of User Generated Twitter Updates using Various Classification Techniques.

Peng, W., 2011. *Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization*. s.l.:s.n.

Pratiwi, A., 2017. *Cyberbullying* [Interview] (18 Oktober 2017).

Putranti, N. D. & Winarko, E., 2014. Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine. *IJCCS*, Volume 8, pp. 91-100.

Rofiqoh, U., 2017. Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine Dan Lexicon Based Features.

Santosa, B., 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.

Stauffer, S., Heath, M. A., Coyne, S. M. & Ferrin, S., 2012. High School Teachers Perceptions of Cyberbullying Prevention and Intervention Strategies. *Psychology in the Schools*, Volume 49.

Tiara, Sabariah, M. K. & Effendy, V., 2015. Sentiment Analysis on Twitter Using the Combination of Lexicon-Based and Support Vector Machine for Assessing the Performance of a Television Program. *International Conference on Information and Communication Technology*, Volume 3rd.

Utomo, M. S., 2013. Implementasi Stemmer Tala pada Aplikasi Berbasis Web. *Jurnal Teknologi Informasi DINAMIK*, Volume 18, pp. 41-45.

Vijayakumar, W. S., 1999. Sequential Support Vector Classifiers and Regression. *International Conference on Soft Computing*, Issue SOCO'99, pp. 610-619.

Willard, N., 2007. Educator's Guide to Cyberbullying and Cyberthreats.

Wirawan, I. N. T. & Eksistyanto, I., 2015. Penerapan Naive Bayes Pada Intrusion Detection System Dengan Diskritisasi Variabel. *JUTI*, Volume 13, pp. 182-189.

Yunita, N., 2016. Analisis Sentimen Berita Artis Dengan Menggunakan Algoritma Support Vector Machine dan Particle Swarm Optimization. *Jurnal Sistem Informasi STMIK Antar Bangsa*, Volume V.

Yusuf, O., 2017. *Naik 100 Juta, Berapa Jumlah Pengguna Instagram Sekarang?*. [Online]

Tersedia di: <<http://tekno.kompas.com/read/2017/09/29/06304447/naik-100>

juta-berapa-jumlah-pengguna-instagram-sekarang>
[Diakses 11 Februari2018].

