

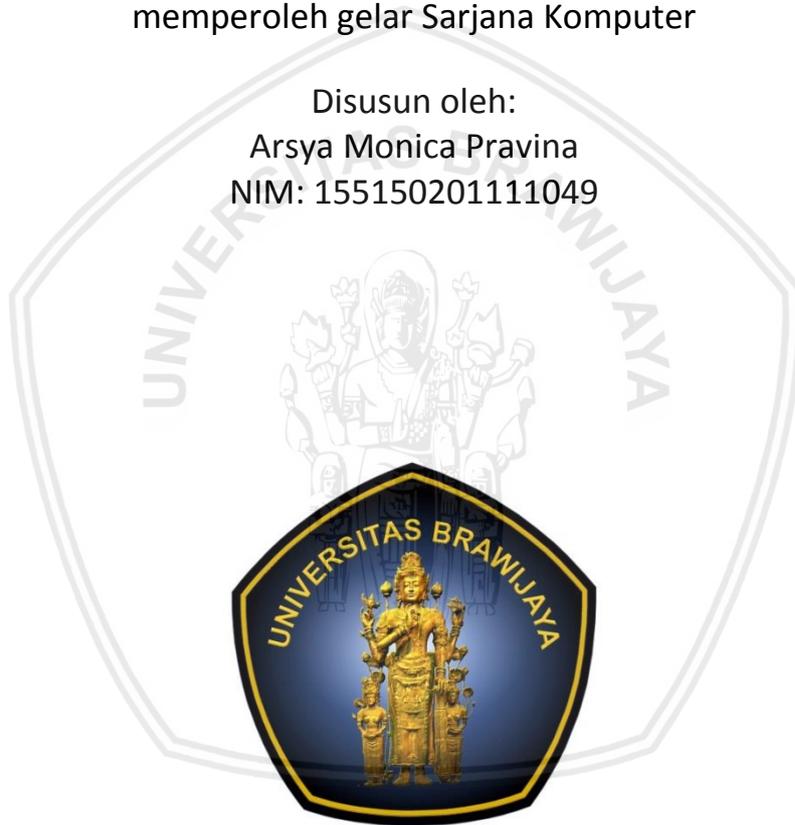
repository.ub.ac.id

**ANALISIS SENTIMEN TENTANG OPINI MASKAPAI
PENERBANGAN PADA DOKUMEN TWITTER
MENGUNAKAN ALGORITME *SUPPORT VECTOR MACHINE*
(SVM)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Arsya Monica Pravina
NIM: 155150201111049



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

ANALISIS SENTIMEN TENTANG OPINI MASKAPAI PENERBANGAN PADA
DOKUMEN TWITTER MENGGUNAKAN ALGORITME *SUPPORT VECTOR MACHINE*
(SVM)

SKRIPSI

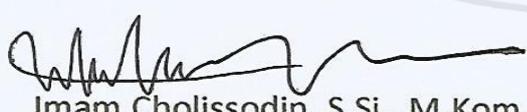
Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Arsya Monica Pravina
NIM: 155150201111049

Skripsi ini telah diuji dan dinyatakan lulus pada
03 Januari 2019
Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II



Imam Cholissodin, S.Si., M.Kom.

NIK: 201201 850719 1 001



Putra Pandu Adikara, S.Kom., M.Kom.

NIP: 19850725 200812 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Ni Astoto Kurniasari, S.T., M.T., Ph.D.

NIP: 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 di dalam 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, 28 Desember 2018



Arsya Monica Pravina

NIM: 155150201111049

PRAKATA

Alhamdulillahirrabil'alamin, puji syukur penulis panjatkan kehadiran Allah SWT karena dengan izin dan rida-Nya penulis dapat menyelesaikan laporan skripsi yang berjudul "Analisis Sentimen Tentang Opini Maskapai Penerbangan Pada Dokumen Twitter Menggunakan Algoritme *Support Vector Machine* (SVM)". Penulisan skripsi ini merupakan salah satu persyaratan untuk meraih gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa hasil dari penulisan skripsi ini belum sempurna, namun penulis telah berusaha membuat hasil penulisan yang mendekati sempurna dengan upaya maksimal. Tentunya penulis tidak mungkin dapat berusaha menyelesaikan skripsi tanpa dukungan dari pihak lain. Rasa hormat dan terima kasih yang tak terhingga atas bantuan, petunjuk, arahan, *support*, doa kepada:

1. Bapak Imam Cholissodin, S.Si, M.Kom dan Bapak Putra Pandu Adikara, S.Kom, M.Kom selaku pembimbing skripsi yang telah membimbing, memberikan arahan, dan motivasi kepada penulis hingga dapat menyelesaikan skripsi dengan baik dan tepat pada waktunya.
2. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Ayah H. Cecep Pramono, S.E dan Ibu dr. Hj. Eka Alfina N., orang tua penulis yang tidak pernah letih memberikan dukungan dari segala sisi dan doa yang tidak ada hentinya.
6. Alya Salsabila Prashafa dan Muhammad Adam Prawira, adik-adik tersayang yang telah membantu, memberi semangat, dorongan, dan doa.
7. Nenek Aliani, Eyang lin, Tante lin, Tante Maya, Tante Susi, Mang Alfin, Om Jimmy, om dan tante lainnya, saudara sepupu tersayang yaitu: Bella, Ayurei, Mba Puti, Ghina, Mba Sasa, Rommy, Mas Odi, Te Vanda dan lainnya, serta seluruh keluarga yang selalu memberikan semangat dan doa.
8. Teman-teman yang tidak bosan untuk membantu, menemani, memberi dorongan dalam menjalani perkuliahan dan keseharian penulis: Tania, Pima, Annisa, Krishna, Rama, Fatur, Fika, Jay, Mba Wanda, Kak Hafiz, Kak Irin, Rafi, Sule, Alfita, Afina, Bunga, Putri, Desy, Carlita, Biya, Enoy, Ofi, Ulfa, Kak Talitha, Mas Acung, Departemen Kominfo EMIF UB 2015/2016, Departemen Internal EMIF UB 2016/2017, tidak lupa teman-teman HDC

tersayang yaitu: Tiara, Fira, Ei, Indah, Kinas, Rifda, Anggi, Muti, Rina, Ipeh, Andin, Shige.

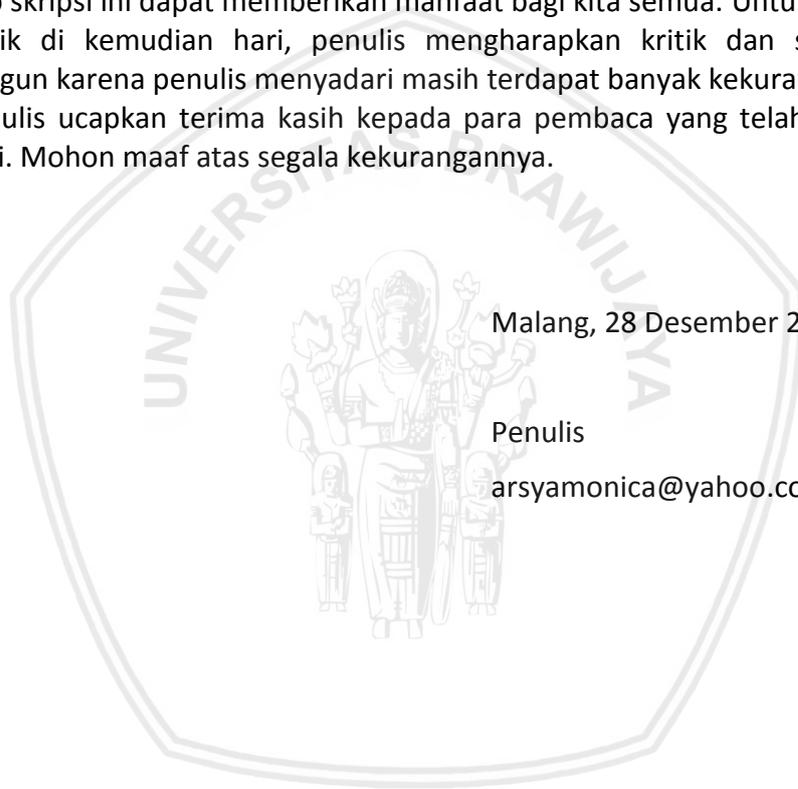
9. Seluruh dosen yang telah meluangkan waktu dan memberikan ilmunya dari awal masa perkuliahan hingga skripsi ini telah selesai.
10. Karyawan Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan arahan dan bantuan.
11. Seluruh pihak yang tidak bisa disebutkan satu per satu.

Terima kasih atas segala bantuan dan doanya. Kelak Allah SWT yang akan membalas kebaikan dari seluruh pihak yang telah penulis sebutkan. Penulis berharap skripsi ini dapat memberikan manfaat bagi kita semua. Untuk hasil yang lebih baik di kemudian hari, penulis mengharapkan kritik dan saran yang membangun karena penulis menyadari masih terdapat banyak kekurangan. Tidak lupa penulis ucapkan terima kasih kepada para pembaca yang telah membaca skripsi ini. Mohon maaf atas segala kekurangannya.

Malang, 28 Desember 2018

Penulis

arsyamonica@yahoo.com



ABSTRAK

Arsya Monica Pravina, Analisis Sentimen tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme *Support Vector Machine* (SVM)

Pembimbing: Imam Cholissodin, S.Si., M.Kom. dan Putra Pandu Adikara, S.Kom., M.Kom.

Dengan meningkatnya penggunaan Twitter, media sosial yang bekerja secara real-time untuk masyarakat dapat menyampaikan keluhan maupun apresiasinya terhadap maskapai-maskapai penerbangan, perlu dibuat sebuah sistem yang dapat melakukan klasifikasi suatu tweet yang berisikan opini termasuk ke dalam kelas apa, dalam penelitian ini terdapat kelas positif dan negatif. Hal tersebut dilakukan agar dapat membantu perusahaan maskapai penerbangan dalam hal evaluasi peningkatan pelayanan serta dapat membantu masyarakat dalam memilih maskapai penerbangan dengan tepat. Sehingga dilakukan klasifikasi sentimen dengan fitur Lexicon Based yang dapat menerima opini berbahasa lain selain Bahasa Indonesia (dalam penelitian ini digunakan Bahasa Inggris) untuk melakukan analisis sentimen. Digunakan algoritme support vector machine untuk melakukan klasifikasi. Hasil dari penelitian ini menunjukkan parameter optimal dan pengaruh penggunaan Lexicon Based Features. Dengan digunakan parameter C bernilai 10 dan learning rate bernilai 0,03 serta digunakan Lexicon Based Features dengan iterasi sebanyak 50 kali memberikan hasil accuracy sebesar 40%, precision 40%, 100% recall, dan f-measure sebesar 57,14%.

Kata kunci: analisis sentimen, opini maskapai penerbangan, twitter, *support vector machine*, *lexicon based features*

ABSTRACT

Arsya Monica Pravina, Analisis Sentimen tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme *Support Vector Machine* (SVM)

Pembimbing: Imam Cholissodin, S.Si., M.Kom. dan Putra Pandu Adikara, S.Kom., M.Kom.

With the increasing use of Twitter, social media that works in real-time for the public can convey complaints and appreciation to airlines, it is necessary to create a system that can classify a tweet containing opinions including what is the best class, in this study there are positive and negative classes. This is done so that it can help airline companies in terms of evaluating service improvements and can help people choose the right airline. Thus a sentiment classification with Lexicon Based features which is able to receive information in languages other than Indonesian (in this study used in English) is done to conduct sentiment analysis. Use the support vector machine algorithm to classify. The results of this study show optimal parameters and the effect of using Lexicon Based Features. By using parameter C is 10 and the learning rate is 0.03 also used Lexicon Based Features with an iteration of 50 times giving accuracy 40%, precision 40%, recall 100%, and f-measure 57,14%.

Keywords: sentiment analysis, airlines opinion, twitter, support vector machine, lexicon based features

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Maskapai Penerbangan	9
2.3 Media Sosial	12
2.3.1 Twitter	12
2.4 Analisis Sentimen	13
2.5 <i>Text Mining</i>	13
2.5.1 <i>Pre-Processing</i>	14
2.5.2 <i>Pembobotan Term Frequency-Inverse Document Frequency</i>	15
2.6 <i>Lexicon Based Features</i>	16
2.6.1 <i>Normalisasi Min-Max</i>	17
2.7 Klasifikasi.....	17
2.7.1 <i>Support Vector Machine (SVM)</i>	17



2.8 Confusion Matrix.....	19
BAB 3 METODOLOGI PENELITIAN	21
3.1 Tipe Penelitian	21
3.2 Strategi Penelitian.....	21
3.3 Partisipan Penelitian	23
3.4 Lokasi Penelitian	23
3.5 Teknik Penelitian Data	23
3.6 Data Penelitian.....	23
3.7 Teknik Analisis Data	24
3.8 Implementasi Algoritme Penelitian	24
BAB 4 PERANCANGAN DAN IMPLEMENTASI	25
4.1 Deskripsi Permasalahan.....	25
4.2 Deskripsi Umum Sistem	26
4.3 <i>Pre-Processing</i>	27
4.3.1 <i>Case Folding</i>	27
4.3.2 <i>Cleaning</i>	28
4.3.3 <i>Translation</i>	30
4.3.4 <i>Stopword Removal</i>	30
4.3.5 <i>Stemming</i>	31
4.3.6 Tokenisasi.....	32
4.4 Pembobotan TF-IDF	33
4.4.1 Menentukan Fitur Kata	34
4.4.2 Menghitung Nilai TF	35
4.4.3 Menghitung Nilai Wtf.....	36
4.4.4 Menghitung Nilai DF dan IDF	37
4.4.5 Menghitung Nilai TF-IDF.....	39
4.5 <i>Lexicon Based Features</i>	40
4.5.1 Normalisasi <i>Min-Max</i>	40
4.5.2 Nilai Sentimen	42
4.6 <i>Support Vector Machine</i>	43
4.6.1 Perhitungan Kernel	44
4.6.2 Perhitungan Matriks <i>Hessian</i>	45



4.6.3 Perhitungan <i>Training SVM</i>	46
4.6.4 Perhitungan Bias	48
4.6.5 Perhitungan <i>Testing SVM</i>	50
4.7 Manualisasi dengan Algoritme <i>SVM</i>	51
4.7.1 Manualisasi <i>Case Folding</i>	52
4.7.2 Manualisasi <i>Data Cleaning</i>	53
4.7.3 Manualisasi <i>Translation</i>	55
4.7.4 Manualisasi <i>Stopword Removal</i>	56
4.7.5 Manualisasi <i>Stemming</i>	57
4.7.6 Manualisasi Tokenisasi.....	59
4.7.7 Manualisasi Perhitungan Tf, Wtf, DF, dan IDF	60
4.7.8 Manualisasi Perhitungan TF-IDF	62
4.7.9 Manualisasi Perhitungan Normalisasi <i>Min-Max</i>	63
4.7.10 Manualisasi Perhitungan Klasifikasi <i>SVM</i>	64
4.7.11 Manualisasi Perhitungan Evaluasi.....	66
4.8 Skenario Pengujian	67
4.8.1 Skenario Pengujian terhadap Parameter <i>Support Vector Machine</i>	67
4.8.2 Skenario Pengujian Implementasi <i>Lexicon Based Features</i>	68
4.9 Penarikan Kesimpulan	69
4.10 Implementasi	69
4.10.1 <i>Pre-Processing</i>	69
4.10.2 Pembobotan TF-IDF	72
4.10.3 <i>Lexicon Based Features</i>	73
4.10.4 Klasifikasi <i>Support Vector Machine</i>	74
4.10.5 Evaluasi.....	77
BAB 5 HASIL DAN PEMBAHASAN	79
5.1 Hasil.....	79
5.1.1 Hasil Pengujian terhadap Parameter <i>Support Vector Machine</i> ..	79
5.1.2 Hasil Pengujian Implementasi <i>Lexicon Based Features</i>	80
5.2 Pembahasan.....	81
BAB 6 PENUTUP	89



6.1 Kesimpulan.....	89
6.2 Saran	89
DAFTAR REFERENSI	91
LAMPIRAN A MANUALISASI PERHITUNGAN TF-IDF.....	93
LAMPIRAN B PELABELAN AWAL MANUAL OLEH PAKAR	102



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	8
Tabel 2.2 <i>Confusion Matrix</i>	20
Tabel 4.1 Data Latih Proses <i>Case Folding</i>	52
Tabel 4.2 Data Uji Proses <i>Case Folding</i>	52
Tabel 4.3 Data Latih Proses <i>Data Cleaning</i>	53
Tabel 4.4 Data Uji Proses <i>Data Cleaning</i>	54
Tabel 4.5 Data Latih Proses <i>Translation</i>	55
Tabel 4.6 Data Uji Proses <i>Translation</i>	55
Tabel 4.7 Data Latih Proses <i>Stopword Removal</i>	56
Tabel 4.8 Data Uji Proses <i>Stopword Removal</i>	57
Tabel 4.9 Data Latih Proses <i>Stemming</i>	58
Tabel 4.10 Data Uji Proses <i>Stemming</i>	58
Tabel 4.11 Data Latih Proses Tokenisasi	59
Tabel 4.12 Data Uji Proses Tokenisasi.....	60
Tabel 4.13 Perhitungan Nilai TF dan DF pada Data Latih dan Data Uji.....	60
Tabel 4.14 Perhitungan Nilai $w_{tf,d}$ dan idf pada Data Latih dan Data Uji	61
Tabel 4.15 Perhitungan TF-IDF pada Data Latih dan Data Uji.....	63
Tabel 4.16 Normalisasi Pembobotan <i>Lexicon</i>	63
Tabel 4.17 Normalisasi <i>Min-Max</i>	64
Tabel 4.18 Hasil Perhitungan Kernel pada Data Latih.....	64
Tabel 4.19 Hasil Perhitungan <i>Error Rate</i> pada Data Latih.....	64
Tabel 4.20 Hasil Perhitungan Delta Alfa pada Data Latih	65
Tabel 4.21 Hasil Perhitungan α Baru pada Data Latih	65
Tabel 4.22 Hasil Perhitungan Nilai Bias.....	65
Tabel 4.23 Hasil Perhitungan Klasifikasi pada Data Uji.....	65
Tabel 4.24 Hasil <i>Confusion Matrix</i>	66
Tabel 4.25 Skenario Pengujian Konstanta <i>Learning Rate</i>	67
Tabel 4.26 Skenario Pengujian Konstanta Nilai C	68
Tabel 4.27 Skenario Pengujian Implementasi <i>Lexicon Based Features</i>	68
Tabel 5.1 Hasil Pengujian Konstanta <i>Learning Rate</i>	79

Tabel 5.2 Hasil Pengujian Konstanta Nilai C..... 80
Tabel 5.3 Hasil Pengujian Implementasi *Lexicon Based Features*..... 81
Tabel 5.4 Hasil Analisis Data Uji dengan Parameter Optimal 88



DAFTAR GAMBAR

Gambar 2.1 <i>Tweet</i> Pengguna terhadap Maskapai Batik Air	12
Gambar 2.2 <i>Tweet</i> Pengguna terhadap Maskapai Citilink.....	12
Gambar 2.3 <i>Tweet</i> Pengguna terhadap Maskapai Garuda Indonesia	12
Gambar 2.4 <i>Tweet</i> Pengguna terhadap Maskapai Lion Air	12
Gambar 2.5 <i>Tweet</i> Pengguna terhadap Maskapai Sriwijaya Air.....	12
Gambar 4.1 Deskripsi Sistem	26
Gambar 4.2 Alur <i>Pre-Processing</i>	27
Gambar 4.3 Alur <i>Case Folding</i>	28
Gambar 4.4 Alur <i>Cleaning</i>	29
Gambar 4.5 Alur <i>Translation</i>	30
Gambar 4.6 Alur <i>Stopword Removal</i>	31
Gambar 4.7 Alur <i>Stemming</i>	32
Gambar 4.8 Alur Tokenisasi	33
Gambar 4.9 Alur Pembobotan TF-IDF.....	34
Gambar 4.10 Tentukan Fitur Kata.....	35
Gambar 4.11 Alur Hitung Nilai TF	36
Gambar 4.12 Alur Hitung Nilai Wtf	37
Gambar 4.13 Alur Hitung Nilai DF dan IDF.....	38
Gambar 4.14 Alur Hitung Nilai TF-IDF	39
Gambar 4.15 Alur <i>Lexicon Based Features</i> dengan Normalisasi <i>Min-Max</i>	41
Gambar 4.16 Alur <i>Lexicon Based Features</i> dengan Nilai Sentimen.....	42
Gambar 4.17 Alur Perhitungan SVM.....	43
Gambar 4.18 Alur Perhitungan Kernel.....	44
Gambar 4.19 Alur Perhitungan Matriks <i>Hessian</i>	45
Gambar 4.20 Alur Perhitungan <i>Training SVM</i>	47
Gambar 4.21 Alur Perhitungan Bias.....	49
Gambar 4.22 Alur Perhitungan <i>Testing SVM</i>	51
Gambar 5.1 Grafik Hasil Pengujian terhadap Parameter <i>Learning Rate</i> (Iterasi=50)	83



Gambar 5.2 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=100) 84

Gambar 5.3 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=150) 84

Gambar 5.4 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=200) 85

Gambar 5.5 Grafik Hasil Pengujian terhadap Parameter *Nilai C* (Iterasi=50)..... 85

Gambar 5.6 Grafik Hasil Pengujian terhadap Parameter *Nilai C* (Iterasi=100)..... 86

Gambar 5.7 Grafik Hasil Pengujian terhadap Parameter *Nilai C* (Iterasi=150)..... 86

Gambar 5.8 Grafik Hasil Pengujian terhadap Parameter *Nilai C* (Iterasi=200)..... 87

Gambar 5.9 Grafik Hasil Pengujian Implementasi *Lexicon Based Features*..... 87

Gambar 5.10 Hasil Analisis Data Uji dengan Parameter Optimal 88



DAFTAR LAMPIRAN

LAMPIRAN A MANUALISASI PERHITUNGAN TF-IDF.....	93
A.1 Tabel Perhitungan TF.....	93
A.2 Tabel Perhitungan IDF.....	95
A.3 Tabel Perhitungan TF-IDF.....	98
LAMPIRAN B PELABELAN AWAL MANUAL OLEH PAKAR.....	102
B.1 Tabel Pelabelan Awal.....	102



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada dasarnya, seseorang sering kali membutuhkan pendapat orang lain ketika ingin membeli maupun memilih sesuatu. Saat ini, tanpa perlu mengeluarkan banyak tenaga, seseorang dapat hanya dengan duduk manis dan membuka internet saja untuk mengecek pendapat atau review orang lain mengenai suatu hal yang sedang dicari atau dibutuhkan. Media sosial adalah salah satu media yang menyediakan kebutuhan tersebut. Isi media sosial tersebut berupa pesan-pesan pribadi, berita, gagasan, serta produk-produk budaya berbentuk digital yang diproduksi dan dikonsumsi oleh individu, organisasi, dan industri (Howard & Parks, 2012). Banyak sekali media sosial yang ada saat ini, salah satu media sosial yang populer yaitu Twitter. Twitter merupakan media sosial yang bekerja secara *real-time*, yang memungkinkan pengguna untuk mengekspresikan opini dan perasaan mereka mengenai banyak isu atau permasalahan (Hamdan, Bellot & Bechet, 2015).

Twitter memberikan banyak keuntungan bagi masyarakat. Misalnya dalam dunia bisnis, masyarakat dapat mengetahui apakah sebuah layanan, produk, atau lain sebagainya dinilai baik atau tidak melalui opini masyarakat yang dituliskan di Twitter. Selain itu, keuntungan bagi perusahaan penyedia layanan atau produk tersebut juga dapat menggunakan opini masyarakat tersebut sebagai bahan evaluasi agar dapat meningkatkan kualitas maupun pelayanannya. Dalam dunia perkuliahan, mahasiswa tidak hanya berkontribusi dalam hal akademik. Bentuk kontribusi biasanya dalam bentuk keikutsertaan para mahasiswa dalam pengadaan acara-acara yang akan diadakan untuk keperluan tertentu dari jurusan, fakultas, maupun universitas. Tidak hanya di dalam kampus, saat ini sudah banyak komunitas, organisasi, perkumpulan yang ada di luar kampus yang menarik perhatian para mahasiswa untuk ikutserta di dalamnya. Biasanya, mereka sering mengadakan acara-acara yang cukup besar, tidak hanya untuk anggotanya namun untuk umum. Mereka sering kali melibatkan artis atau penyanyi (untuk acara musik) atau seorang pembicara terkenal (untuk acara seminar) dan lain-lain untuk mengisi acara mereka. Tetapi kebanyakan orang-orang terkenal tersebut berasal dari luar tempat diadakannya acara tersebut (luar kota atau negeri). Tentunya akan dibutuhkan akomodasi seperti pesawat terbang, sehingga para mahasiswa biasanya melakukan survey di internet terutama di media sosial berdasarkan opini orang lain untuk mengetahui maskapai apa yang dikenal paling baik dan ekonomis dibanding maskapai lainnya, yang mana di Indonesia terdapat banyak maskapai penerbangan.

Maskapai penerbangan merupakan organisasi atau jasa layanan transportasi yang tentunya sangat dibutuhkan masyarakat untuk berpindah tempat dari satu kota ke kota yang lain maupun satu negara ke negara lain dengan waktu yang cepat. Terdapat beberapa maskapai penerbangan yang terkenal di Indonesia

antara lain seperti Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air dan Citilink Indonesia. Terkadang masyarakat masih sering bingung ketika memilih maskapai penerbangan untuk membawa mereka berpergian, ada maskapai yang memiliki biaya yang sangat mahal tetapi pelayanannya buruk dan ada pula maskapai dengan biaya murah namun pelayanannya sangat baik. Banyak orang yang sering melampirkan kesenangan, kepuasan, maupun keluhannya mengenai maskapai-maskapai penerbangan di media sosial Twitter. Ditemukan bahwa penggunaan media sosial Twitter terus meningkat setiap tahunnya, peningkatan tersebut juga mencakup pemanfaatan media sosial Twitter yang terus berkembang dalam hal sarana komunikasi dan informasi antara pelanggan dengan sebuah perusahaan penyedia layanan atau produk tertentu, dalam penelitian ini yaitu penggunaan media sosial Twitter bagi pelanggan dan perusahaan maskapai penerbangan.

Berdasarkan data yang ada di Twitter Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air maupun Citilink Indonesia merupakan beberapa perusahaan atau maskapai penerbangan yang cukup aktif di media sosial. Tidak hanya untuk melakukan publikasi, namun untuk merespon para pelanggan yang bertanya maupun melampirkan keluhan kesahnya. Masyarakat juga cukup antusias dengan adanya media sosial maskapai-maskapai tersebut dengan memposting *tweets* melalui akun pribadinya. Karena saat ini, banyak perusahaan yang sudah tidak aktif lagi dalam menggunakan Twitter sebagai layanan komunikasi dan informasi sebuah perusahaan tersebut, padahal nama akun Twitter perusahaan tersebut tertera pada website maupun brosur milik perusahaannya. Namun, ditemukan beberapa permasalahan antara lain seperti terdapat banyak kalimat yang tidak terstruktur bermakna rancu dengan ejaan gaul yang disertakan URL sehingga dapat terjadi penurunan akurasi ketika menganalisis sentimen apalagi dengan tidak ditambahkan teknik-teknik ekstraksi fitur tertentu. Padahal, berdasarkan fakta-fakta yang telah dijelaskan sebelumnya, dibutuhkan sebuah sistem otomatis untuk menganalisis sentimen dari opini masyarakat mengenai maskapai penerbangan pada dokumen Twitter untuk memudahkan perusahaan dalam bahan evaluasi kinerja serta kualitas perusahaan serta untuk memudahkan masyarakat dalam membantu memilih maskapai penerbangan yang paling tepat sesuai kebutuhan melalui opini masyarakat yang tentunya berbeda-beda. Oleh karena itu, dipilihlah opini tentang Garuda Indonesia (@IndonesiaGaruda), Lion Air (@lionairgroup), Batik Air (@BatikAirINA), Sriwijaya Air (@SriwijayaAir) dan Citilink Indonesia (@Citilink) pada dokumen Twitter sebagai objek penelitian.

Pada penelitian ini, setiap kalimat akan dianalisis secara terpisah dan diklasifikasi sebagai positif dan negatif. Namun sebelum itu, dianjurkan untuk melakukan pemrosesan teks atau yang disebut *pre-processing text*, karena proses tersebut sangat membantu meningkatkan hasil yang akurat pada analisis sentimen (Zhao, 2015). Selain pemrosesan teks, teknik ekstraksi fitur juga dapat membantu menghasilkan nilai akurasi yang lebih baik ketika disatukan dengan metode klasifikasi seperti *Naïve Bayes* atau *Support Vector Machine (SVM)*, namun hasil akurasi lebih unggul ekstraksi fitur dengan SVM (Hidayatullah, 2014). Teknik ekstraksi fitur TF-IDF adalah salah satu jenis teknik ekstraksi fitur yang membantu

menghasilkan nilai akurasi yang lebih baik ketika disatukan dengan metode klasifikasi *Naïve Bayes* atau *Support Vector Machine* (SVM), namun hasil akurasi lebih unggul TF-IDF dengan SVM dibandingkan TF-IDF dengan *Naïve Bayes* (Windasari, Uzzi & Satoto, 2017). Sama halnya dengan pengujian yang bertujuan untuk membandingkan metode *Naïve Bayes* dengan *Support Vector Machine*, ditemukan bahwa analisis sentimen pada data yang digunakan dengan menggunakan metode *Naïve Bayes* menghasilkan tingkat akurasi yang lebih kecil 10% dibandingkan ketika menggunakan metode *Support Vector Machine*, namun dalam penelitian tersebut tanpa menggunakan teknik ekstraksi fitur seperti *Term Frequency-Inverse Document Frequency* (Susanti, 2017). Selain Teknik ekstraksi fitur, untuk menentukan persentase kelas sentimen dengan fitur *Lexicon Based* cukup membantu meningkatkan akurasi. Berdasarkan penelitian sebelumnya, didapatkan nilai akurasi yang tinggi yaitu 84% ketika menambahkan fitur *Lexicon Based* dalam penyelesaian analisis sentimen yang dipadukan dengan penggunaan metode *Support Vector Machine* (Tanesab, 2017). Selain itu, fitur *Lexicon Based* apabila dikolaborasikan dengan metode *Support Vector Machine* yang juga menggunakan pembobotan *Term Frequency-Inverse Document Frequency* juga menghasilkan akurasi yang cukup tinggi sebesar 79% (Rofiqoh, 2017). Penggunaan teknik ekstraksi fitur TF-IDF sempat dibandingkan dengan penggunaan *Ratio*, dan *N-Gram* pada klasifikasi sentimen dan klasifikasi filter *tweet* pada Twitter. Ditemukan bahwa penggunaan teknik ekstraksi fitur *Term Frequency-Inverse Document Frequency* menghasilkan tingkat akurasi tertinggi yaitu 74% pada klasifikasi filter *tweet* dan 80% pada klasifikasi sentimen (Juniawan, 2017). Dalam penelitian ini, akan dilakukan ekstraksi fitur dengan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) dan menggunakan fitur *Lexicon Based* setelah melakukan pemrosesan teks dan kemudian dapat dilakukan klasifikasi menggunakan algoritme *Support Vector Machine* (SVM). Opini terhadap Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air, dan Citilink pada dokumen Twitter akan diolah dan diklasifikasikan ke dalam kelas positif dan negatif.

Dengan adanya penelitian ini, diharapkan dapat dengan mudah dan dalam waktu yang singkat untuk sebuah sistem otomatis menunjukkan apakah sebuah dokumen pada Twitter yang dipilih merupakan opini yang positif atau negatif. Dengan tujuan untuk membantu masyarakat dalam memilih maskapai penerbangan sesuai kebutuhan maupun untuk menjadikan opini sebagai bahan evaluasi perusahaan agar sebuah maskapai penerbangan dapat terus mengembangkan pelayanannya. Berdasarkan latar belakang yang sudah dijelaskan sebelumnya, penulis mengajukan sebuah penelitian dengan judul “Analisis Sentimen tentang Opini Maskapai Penerbangan pada Dokumen Twitter menggunakan Algoritme *Support Vector Machine*”.

1.2 Rumusan masalah

Masalah yang akan dibahas pada penelitian ini berdasarkan latar belakang di atas adalah sebagai berikut.

1. Bagaimana mengetahui fitur yang ideal serta parameter yang optimal dalam melakukan analisis sentimen opini masyarakat pada maskapai penerbangan Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air, dan Citilink Indonesia dari media sosial Twitter?
2. Bagaimana tingkat akurasi yang diperoleh algoritme *Support Vector Machine* (SVM) dalam melakukan analisis sentimen opini masyarakat pada maskapai penerbangan Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air, dan Citilink Indonesia dari media sosial Twitter?

1.3 Tujuan

Dari rumusan masalah di atas, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Mengetahui fitur yang ideal serta parameter yang optimal dalam melakukan analisis sentimen opini masyarakat pada maskapai penerbangan Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air, dan Citilink Indonesia dari media sosial Twitter.
2. Menguji tingkat akurasi yang diperoleh algoritme *Support Vector Machine* (SVM) dalam melakukan analisis sentimen opini masyarakat pada maskapai penerbangan Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air, dan Citilink Indonesia dari media sosial Twitter.

1.4 Manfaat

Manfaat yang akan didapatkan dari penelitian ini antara lain adalah untuk memberikan keuntungan bagi maskapai penerbangan dalam pertimbangan peningkatan kualitas dan pelayanan dari perusahaannya berdasarkan opini masyarakat mengenai maskapai penerbangan tersebut, serta memberikan bantuan bagi masyarakat yang sedang mencari tahu atau mempertimbangkan memilih maskapai penerbangan.

1.5 Batasan masalah

Berikut adalah beberapa batasan yang diberikan untuk memfokuskan ruang lingkup pada penelitian ini agar dapat lebih mudah dipahami.

1. Sumber data berasal dari opini masyarakat yang diposting pada media sosial Twitter untuk Garuda Indonesia (@IndonesiaGaruda), Lion Air (@lionairgroup), Batik Air (BatikAirINA), Sriwijaya Air (@SriwijayaAir), dan Citilink Indonesia (@Citilink).
2. Jumlah data sebanyak 250 opini tentang maskapai penerbangan pada dokumen Twitter.
3. Data yang digunakan sebanyak 125 opini positif dan 125 opini negatif.
4. Data berbahasa Inggris akan diterjemahkan terlebih dahulu menjadi bahasa Indonesia.

5. Teknik ekstraksi fitur yang digunakan adalah *Term Frequency-Inverse Document Frequency*.
6. Maskapai-maskapai penerbangan yang digunakan merupakan maskapai penerbangan yang beroperasi di Bandara Abdul Rachman Saleh (Malang).

1.6 Sistematika pembahasan

Pada bagian ini, berisi struktur skripsi penelitian yang akan dimulai dari Bab Pendahuluan sampai dengan Bab Penutup beserta deskripsi dari tiap bab secara ringkas. Bagian ini diperuntukkan bagi pembaca agar lebih mudah memahami skripsi ini.

BAB 1 PENDAHULUAN

Bab ini berisi tentang bagaimana kondisi terkini terkait topik analisis sentimen. Disampaikan juga mengenai penelitian sebelumnya yang menjadi dasar untuk penelitian ini. Kemudian disampaikan tentang permasalahan dan hasil yang ingin didapatkan dari penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai sumber penelitian yang mengawal penelitian tentang analisis sentimen. Kemudian terdapat penjelasan terkait sumber pengambilan data dan objek penelitian yaitu maskapai penerbangan Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air, dan Citilink. Dijelaskan juga tentang algoritme yang akan digunakan.

BAB 3 METODOLOGI PENELITIAN

Bab ini menggambarkan bagaimana tahapan yang akan dilakukan untuk menyelesaikan penelitian dari awal hingga akhir. Mulai dari bagaimana pengumpulan data dilakukan, kemudian tahapan persiapan atau *pre-processing* untuk selanjutnya di proses ke ekstraksi fitur *Term Frequency-Inverse Document Frequency* sebelum mengklasifikasi hasil opini masyarakat ke beberapa kelas dan mendapatkan hasil.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini memberikan gambaran untuk sistem yang akan dibuat dengan memberikan rancangan sistem secara keseluruhan, kemudian rancangan penggunaan metode, hingga rancangan untuk bagaimana pengujian dilakukan. Menjelaskan proses implementasi metode yang digunakan ke dalam sistem perangkat lunak, beserta penjelasan komponen-komponen yang ada yang kemudian dijelaskan seperti apa hasil implementasi tersebut.

BAB 5 HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil pengujian yang dilakukan terhadap penggunaan metode *Support Vector Machine* dan ekstraksi fitur *Term Frequency-Inverse Document Frequency* dalam memberikan hasil klasifikasi yang baik terhadap opini

masyarakat yang memiliki dua kelas. Akan disebutkan dan dijelaskan hasil dari beberapa macam pengujian juga.

BAB 6 PENUTUP

Bab ini berisi kesimpulan dari hasil akhir pada penelitian analisis sentimen dengan menggunakan metode dan ekstraksi fitur yang telah disebutkan sebelumnya kemudian juga diberikan saran untuk penelitian-penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Ditemukan sebuah artikel yang mengatakan bahwa sangat sering terdapat data dengan berbagai macam variasi dan tidak terstruktur (Ariwibowo, 2015). Sering sekali bermakna rancu dan membuat bingung. Kata-kata pada dokumen biasanya singkat dan bahasa yang digunakan sering kali tidak formal dengan ejaan yang salah, berbahasa slang/gaul, dan sering terdapat *link*/URL di dalam sebuah dokumen serta kesalahan-kesalahan lain yang muncul (Han & Guo, 2013). Sehingga menyebabkan penurunan akurasi dalam penentuan klasifikasi ke kelas yang tepat (Bhadane, Dalal & Doshi, 2015). Permasalahan-permasalahan tersebut merupakan beberapa faktor diadakannya penelitian yang berhubungan dengan sebuah data (dokumen) yang terdiri dari banyak kata dan kalimat. Biasanya, para peneliti melakukan penelitian mengenai sebuah dokumen yang berisikan suara masyarakat mengenai suatu hal.

Untuk topik mengenai analisis sentimen, sudah banyak penelitian sebelumnya dengan berbagai metode dan fitur ekstraksinya yang tentu berbeda-beda. Ada sebuah penelitian dari Ike Pertiwi Windasari, Fajar Nurul Uzzi, Kodrat Iman Satoto (2017) yang berjudul "*Sentiment Analysis on Twitter Posts: An Analysis of Positive or Negative Opinion on Gojek*". Pada penelitian tersebut menghasilkan 2 kelas klasifikasi dari *tweets* mengenai opini masyarakat tentang Gojek, yaitu kelas positif dan kelas negatif. Data diambil dari proses "*tweets Crawling*" pada Twitter. Data yang digunakan sebanyak 1000 positif dan 1000 negatif. Pada penelitian tersebut, peneliti mengusulkan penggunaan metode *Support Vector Machine* (SVM) untuk klasifikasi *tweets* dan dibantu dengan ekstraksi fitur yaitu pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Ternyata, hasil dari penelitian tersebut menghasilkan hasil yang baik dengan tingkat akurasi sebesar 86%. Hampir sama dengan penelitian pada tahun selain itu, pada penelitian yang dilakukan oleh Novita Dewi Susanti (2017), diambil 350 data berupa opini mengenai Candi Prambanan dari aplikasi Foursquare. Peneliti melakukan perbandingan antara penggunaan metode *Support Vector Machine* (SVM) dengan penggunaan metode *Naïve Bayes* yang juga menggunakan *Rapid Miner* di keduanya. Hasil penelitian menunjukkan bahwa hasil akurasi SVM lebih besar 10,96% dibandingkan dengan *Naïve Bayes* yang juga menggunakan *Rapid Miner* tersebut. SVM memberikan hasil akurasi sebesar 76,47% dan *Naïve Bayes* memberikan hasil akurasi sebesar 65,78%. Sama halnya dengan penelitian yang dilakukan pada tahun 2017 oleh Dede Juniawan yang melakukan perbandingan antara beberapa teknik ekstraksi fitur seperti *Term Frequency-Inverse Document Frequency* (TF-IDF), *Ratio*, dan *N-Gram* dikolaborasikan dengan penggunaan algoritme *Support Vector Machine*, didapatkan hasil sebagai berikut. Metode TF-IDF menghasilkan akurasi tertinggi yaitu 74% pada klasifikasi filter *tweet* dan 80% pada klasifikasi sentimen, *Ratio* menghasilkan akurasi 72% pada klasifikasi filter *tweet* dan 74% pada klasifikasi sentimen, dan *N-Gram* menghasilkan akurasi 64% pada klasifikasi filter *tweet* dan 78% pada klasifikasi sentimen. Berdasarkan

penelitian sebelumnya yang dilakukan oleh Fiktor Immanuel Tanesab pada tahun 2017, selain Teknik ekstraksi fitur, untuk menentukan persentase kelas sentimen dengan fitur *Lexicon Based* cukup membantu meningkatkan akurasi. Didapatkan nilai akurasi yang tinggi yaitu 84% ketika menambahkan fitur *Lexicon Based* dalam penyelesaian analisis sentimen yang dikolaborasikan dengan penggunaan metode *Support Vector Machine*. Selain itu, pada penelitian yang dilakukan oleh Umi Rofiqoh pada tahun 2017, menunjukkan bahwa fitur *Lexicon Based* apabila dikolaborasikan dengan metode *Support Vector Machine* yang juga menggunakan pembobotan *Term Frequency-Inverse Document Frequency* juga menghasilkan akurasi yang cukup tinggi yaitu sebesar 79%.

Tabel 2.1 Kajian Pustaka

No.	Pustaka	Objek	Metode	Hasil
1.	(Windasari, Uzzi & Satoto, 2017)	Opini tentang Gojek	<i>Support Vector Machine (SVM)</i> + TF-IDF	Apabila dibandingkan dengan <i>Naïve Bayes</i> + TF-IDF, maka SVM + TF-IDF akan lebih unggul. Akurasi yang didapat ketika SVM dengan TF-IDF sebesar 86 %.
2.	(Rofiqoh, 2017)	Opini tentang penyedia layanan telekomunikasi seluler Indonesia	<i>Support Vector Machine (SVM)</i> + <i>Lexicon Based Features</i>	Hasil akurasi yang didapat sebesar 79%, <i>precision</i> sebesar 65%, <i>recall</i> sebesar 97%, dan <i>f-measure</i> sebesar 78%.
3.	(Tanesab, 2017)	Opini tentang kinerja Ahok	<i>Support Vector Machine (SVM)</i> + <i>Lexicon Based Features</i>	Hasil akurasi sebesar 84%, <i>precision</i> 91%, <i>recall</i> 80%, <i>TP rate</i> 91.1 dan <i>TN rate</i> 44.8%.
4.	(Susanti, 2016)	Data mengenai Candi Prambanan	<i>Naïve Bayes</i> dan <i>Support Vector Machine (SVM)</i>	Analisis Sentimen pada data di foursquare mengenai Candi Prambanan dengan

No.	Pustaka	Objek	Metode	Hasil
				metode <i>Naive Bayes</i> dengan akurasi 65,78% sedangkan pada SVM dengan nilai akurasi 76,48%.
5.	(Juniawan, 2017)	Seleksi fitur untuk klasifikasi SVM	<i>Support Vector Machine (SVM)</i>	Metode TF-IDF menghasilkan akurasi tertinggi yaitu 74% pada klasifikasi filter <i>tweet</i> dan 80% pada klasifikasi sentimen, <i>Ratio</i> menghasilkan akurasi 72% pada klasifikasi filter <i>tweet</i> dan 74% pada klasifikasi sentimen, dan <i>N-Gram</i> menghasilkan akurasi 64% pada klasifikasi filter <i>tweet</i> dan 78% pada klasifikasi sentimen.

2.2 Maskapai Penerbangan

Maskapai penerbangan merupakan organisasi atau jasa layanan transportasi yang tentunya sangat dibutuhkan masyarakat untuk berpindah tempat dari satu kota ke kota yang lain maupun satu negara ke negara lain dengan waktu yang cepat. Ada beberapa maskapai penerbangan terkenal yang berpusat di Jakarta, Indonesia antara lain seperti Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air dan Citilink Indonesia. Maskapai-maskapai penerbangan tersebut merupakan maskapai penerbangan yang telah beroperasi di kota-kota besar di Indonesia.

Berawal dari banyaknya permintaan untuk adanya jasa transportasi udara berdasarkan kebutuhan bangsa pada jaman ketika Indonesia sedang mempertahankan kemerdekaannya, muncullah "Indonesian Airways" pada tahun

1949 yang kemudian berubah menjadi “Garuda Indonesian Airways” yang mana nama “Garuda” diberikan pertama kali oleh Presiden Soekarno. Saat ini, Garuda Indonesia berkantor pusat di Gedung Garuda Indonesia Jl. Kebon Sirih No. 44, Jakarta 10110, Indonesia. Dari dahulu sampai saat ini, perusahaan terus menerus mengembangkan jaringan penerbangan dari Barat sampai Timur Indonesia dan saat ini Garuda Indonesia melayani 83 tujuan penerbangan di seluruh dunia. Tidak hanya jaringan penerbangan, namun kualitas pelayanan juga selalu diperhatikan. Garuda Indonesia merupakan maskapai penerbangan berbintang lima yang sudah diakui kualitasnya oleh banyak orang. Garuda Indonesia dikenal memberikan pelayanan terbaik yang mengedepankan keramahtamahan dan kekayaan budaya Indonesia. Bukti-buktinya dapat dilihat dalam pencapaian berskala internasional oleh Garuda Indonesia, diantaranya seperti “*The World’s Best Cabin Crew*” pada tahun 2014 sampai 2017, “*The World’s Most Loved Airline 2016*” dan “*The World’s Best Economy Class 2013*” dari Skytrax. Garuda Indonesia juga merupakan salah satu maskapai yang menyediakan beberapa media sosial untuk melakukan promosi maupun berkomunikasi dengan pelanggan, antara lain seperti Facebook, Twitter, Youtube, Instagram, dan Line (sumber: <https://www.garuda-indonesia.com/>).

Lion Air adalah bagian dari Lion Group yang terdiri dari Wings Air, Batik Air, Lion Bizjet, Malindo Air, dan Thai Lion. Lion Air merupakan maskapai penerbangan yang mulai beroperasi pada tahun 2000. Lion Air memiliki kantor pusat di Jalan Gajah Mada No.7, Kota Jakarta Pusat, 10130. Dengan memiliki sebanyak 112 armada dan telah mengoperasikan pesawat ke 183 rute perjalanan gabungan antara rute nasional dan internasional, PT. Lion Mentari Airlines atau disebut Lion Air memiliki biaya penerbangan yang rendah dengan slogan “We Make People Fly” yang bermakna bahwa dengan harga yang murah siapapun bisa terbang bersama Lion Air dengan mengutamakan keselamatan, keamanan, dan kualitas penerbangan. Sampai saat ini, Lion Air telah memiliki beberapa penghargaan maupun sertifikasi internasional sebagai motivasi untuk selalu meningkatkan kualitas pelayanannya, seperti ISSA, sebuah standar keselamatan dan keamanan berskala internasional yang diberikan oleh IATA pada tahun 2016. Selanjutnya, sertifikasi ISO 9001:2015 mengenai delay management. Selain *call center*, Lion Air memiliki akun resmi pada media sosial Facebook dan Twitter untuk promosi serta menerima kritik saran dari pelanggannya (sumber: <http://www.lionair.co.id/id>).

Berawal dari perencanaan didirikannya maskapai penerbangan dengan layanan penuh, untuk mengalihkan jumlah pesanan yang banyak dipesan oleh Lion Air, Rusdi Kirana mendirikan maskapai penerbangan Batik Air pada tahun 2013 lalu. Masih terhitung baru, maskapai penerbangan Batik Air didirikan pada tanggal 1 Maret 2013 dan mulai beroperasi pada 3 Maret 2013. Maskapai penerbangan Batik Air berkantor pusat di Jakarta, di bawah perusahaan induk Lion Air Group. Berikut adalah lokasi-lokasi kantor perwakilan Batik Air. Ambon, Banda Aceh, Bandung, Balikpapan, Banjarmasin, Batam, Bekasi, Bengkulu, Bengkulu. Meski terhitung baru, Batik Air sudah sangat dikenal dan sering dipilih pelanggan untuk digunakan berpergian. Batik Air cukup aktif dalam media sosial untuk

berkomunikasi dengan pelanggan, antara lain aktif dalam Facebook, Twitter, dan sebagainya (sumber: https://id.wikipedia.org/wiki/Batik_Air).

Selanjutnya, PT. Sriwijaya Air adalah perusahaan swasta murni yang didirikan oleh Chandra Lie, Hendry Lie, Johannes Bunjamin, dan Andy Halim. Diawali dengan adanya sebuah pesawat Boeing 737-200 sebagai modal utamanya, Sriwijaya Air mulai mengoperasikan penerbangan pertamanya pada tanggal 10 November 2003. Kantor pusat Sriwijaya Air terletak di Sriwijaya Air Tower Jalan Atang Sanjaya No. 21 Soekarno Hatta Airport, Tangerang, Banten, Indonesia. Sriwijaya Air, maskapai penerbangan yang berpusat di Jakarta yang memiliki tujuan-tujuan yang dikemas di dalam visi dan misinya menjadikan Sriwijaya Air menjadi maskapai penerbangan yang dicari dan dibutuhkan masyarakat. Visi misi Sriwijaya Air antara lain mengutamakan kualitas pelayanan, menjadi maskapai penerbangan yang dapat bersaing secara nasional serta regional, mampu melaksanakan bisnis di level dunia, tidak tertinggal teknologi saat ini dan memiliki manajemen perusahaan yang efektif serta efisien, memfasilitasi turis ke destinasi-destinasi yang diinginkan, serta menghasilkan keuntungan dalam bisnis. Dalam hal pelayanan, Sriwijaya cukup dikenal baik. Hal tersebut dapat dibuktikan dengan penghargaan-penghargaan yang telah didapatkan. Salah satunya dalam pelayanan komunikasi dengan *Customer*, Sriwijaya Air mendapat penghargaan "*Good Service Performance*" untuk *call center award* pada tahun 2012 lalu. Dapat dilihat bahwa Sriwijaya Air merupakan perusahaan yang menyediakan fasilitas komunikasi yang baik, selain *call center* melalui telepon, terdapat beberapa media sosial seperti Facebook, Twitter, Instagram, Youtube, dan lain sebagainya (sumber: <https://sriwijayaair.co.id>).

Terakhir, PT. Citilink Indonesia merupakan anak perusahaan Garuda Indonesia. Citilink adalah maskapai penerbangan yang sangat cepat berkembang di Indonesia mulai tahun 2011. Citilink berkantor pusat di Menara Citicon, lantai 16, Jl. Letjend. S. Parman Kav. 72, Jakarta 11410. Visi dari Citilink yaitu menjadi maskapai penerbangan dengan biaya yang murah, menyediakan jasa angkutan udara terjadwal, dan mengutamakan keselamatan. Citilink juga mengutamakan kepuasan pelanggan. Berikut adalah beberapa bukti keberhasilan Citilink dalam komitmen mengutamakan dan meningkatkan pelayanan pada pelanggan.

-Memenangkan kategori *Best Overall Marketing Campaign* di *The Budgeties & Travel Awards 2012*

-Memenangkan *Service To Care Award* dua tahun berturut-turut 2012 dan 2013 untuk *Airlines category* dari *Markplus Insight*.

-Memenangkan *Top IT Innovation Award on Transportation* dari Kemenkominfo. dan masih ada lagi.

Kemudian, inilah beberapa kantor cabang Citilink yang sudah menyebar di Indonesia: Jakarta, Medan, Surabaya, Makassar, Bali, Semarang, Yogyakarta, Balikpapan, Banjarmasin, Kupang, Palembang, Padang, Lombok, Batam, Pangkalpinang, Bengkulu, Tanjung Pandan, Pekanbaru, Malang, Solo, Palangkarya, Dili, Jambi, Manado, Pontianak, Aceh. Citilink juga merupakan salah satu maskapai

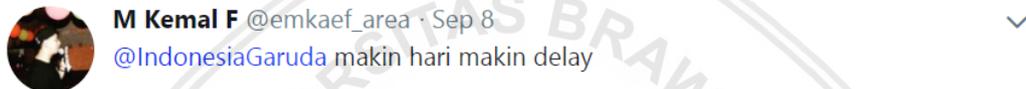
penerbangan yang aktif dan memiliki komunikasi yang baik antara maskapai dan pelanggan. Selain *call center*, Citilink juga menyediakan layanan untuk pelanggan di media sosial seperti Twitter, Facebook, Instagram, Youtube, dan Linkers Magazine (sumber: <https://www.citilink.co.id/>).



Gambar 2.2.1 Tweet Pengguna terhadap Maskapai Batik Air



Gambar 2.2 Tweet Pengguna terhadap Maskapai Citilink



Gambar 2.3 Tweet Pengguna terhadap Maskapai Garuda Indonesia



Gambar 2.4 Tweet Pengguna terhadap Maskapai Lion Air



Mhn bantuannya : Cc @alvinlie21 @OmbudsmanRI137 @YLKI_ID

Gambar 2.5 Tweet Pengguna terhadap Maskapai Sriwijaya Air

2.3 Media Sosial

Media sosial adalah media yang terdiri atas tiga bagian, yaitu infrastruktur informasi beserta alat yang digunakan untuk memproduksi dan mendistribusikan isi media, isi media yang berupa pesan-pesan pribadi, berita, dan, gagasan, serta yang terakhir produk-produk budaya yang berbentuk digital. Kemudian yang memproduksi dan mengkonsumsi isi media dalam bentuk digital adalah individu, organisasi, dan industri (Howard & Parks, 2012).

2.3.1 Twitter

Twitter adalah salah satu media sosial yang terkenal di kalangan masyarakat Indonesia maupun luar Indonesia. Twitter memungkinkan pengguna untuk

mengekspresikan opini dan perasaan mereka mengenai banyak isu atau permasalahan (Hamdan, Bellot & Bechet, 2015). Berbeda dengan media sosial yang lain yang harus menjadi teman terlebih dahulu baru dapat berinteraksi, Twitter memungkinkan antarpengguna tetap terhubung walaupun mereka tidak saling berteman (Windasari, Uzzi & Satoto, 2017). Bagi pengguna yang tidak melakukan registrasi akun maka hanya mendapat fasilitas untuk membaca *tweet* saja, namun bagi pengguna yang mendaftarkan akunnya mereka dapat membaca serta menulis *tweet* (<https://id.wikipedia.org>).

Terdapat pula beberapa istilah maupun fitur yang terdapat pada twitter antara lain sebagai berikut. Beranda, profil, pengikut, mengikuti, *mention*, *hashtag*, *list*, dan *trending topics*. Twitter juga merupakan media sosial yang bekerja secara *real-time* yaitu waktu terjadinya aktual, saat itu juga.

2.4 Analisis Sentimen

Analisis sentimen biasanya melakukan penelitian dalam bentuk analisis, salah satunya mengenai opini dan emosi banyak orang terhadap suatu entitas sebagai contoh permasalahan, topik atau layanan. Dengan penjelasan lain, analisis sentimen merupakan proses ekstraksi emosi atau opini dari sebuah teks atau bacaan (Kaur & Mangat, 2015). Saat ini analisis sentimen sedang marak dibahas dan digunakan dalam penelitian, terutama dalam bidang *text mining* (Kaur & Mangat, 2015). Analisis sentimen sangat berguna untuk menganalisis komentar orang-orang di media sosial, salah satunya di Twitter. Selanjutnya, akan diterjemahkan menjadi lebih bermakna, yang dapat menentukan apakah suatu hal dianggap sosial, positif, atau netral. Dalam dunia yang kompetitif ini, analisis sentimen terhadap opini orang-orang sangat penting untuk meraih kesuksesan sebuah perusahaan, organisasi, atau individu tertentu. Seperti untuk mengetahui jalannya kinerja perusahaan tersebut, atau bagi banyak orang dapat mempermudah untuk menentukan atau memilih sesuatu dengan mengetahui mana yang lebih baik digunakan. Karena saat ini tidak hanya satu perusahaan, organisasi, atau individu yang bergerak dalam satu bidang melainkan banyak perusahaan, organisasi, atau individu yang menjalankan suatu hal dalam satu bidang tersebut. Tidak hanya dalam bidang bisnis, namun juga berguna dalam bidang politik dan psikologi (Chalothorn & Ellman, 2013).

2.5 Text Mining

Text mining merupakan subyek penelitian yang sangat baru dan mulai diminati banyak orang. *Text mining* dapat menyelesaikan masalah-masalah dalam hal teks tidak terstruktur. Dalam penyelesaian masalah, *text mining* biasa digabungkan dengan beberapa subyek lain seperti *Data Mining*, *Natural Language Processing*, dan lain-lain. Dalam *text mining*, terdapat tahap seperti ekstraksi teks menggunakan teknik tertentu, pemrosesan teks atau yang biasa disebut *pre-processing text*, pembobotan atau pemberian indeks pada teks, maupun analisis suatu teks.

Text mining merupakan sebuah proses penemuan informasi, relasi, dan fakta yang tersembunyi di dalam teks ketika dilakukan pemrosesan dan analisis data dalam jumlah besar, struktur teks yang kompleks dan tidak lengkap, dimensi tinggi, serta data yang *noise*. *Text mining* fokus pada relasi dan keberadaan sebuah dokumen dengan yang lainnya. *Text mining* lebih dari sekedar *Information Retrieval*, metode-metode statistik juga digunakan dalam *text mining* untuk meningkatkan akurasi. Menurut para ahli, mengkategorisasikan dokumen adalah termasuk *text mining*. Sebenarnya, dalam kategorisasi dokumen tidak menghasilkan relasi atau fakta yang baru, namun jika dianalisis dan dikorelasikan lebih lanjut maka akan menghasilkan fakta baru tersebut pada tiap dokumen.

Berikut adalah beberapa teks tidak terstruktur. *e-mail, instant message, blog*, dan lain-lain. Apabila sumber teks adalah berita, *text mining* dapat digunakan untuk membandingkan beberapa berita. *Text mining* juga dapat membantu dalam mendeteksi sebuah tren di bidang tertentu, tidak hanya itu proses *deduplication* juga dapat dibantu. Misalnya lain untuk *technical paper*, dokumentasi, maupun spesifikasi perangkat lunak, *text mining* dapat digunakan untuk mengekstrak secara otomatis atau mendeteksi kekurangan antara *source code* dan juga dokumentasinya secara otomatis. Contoh lain untuk halaman web, *text mining* dapat digunakan untuk menganalisis *website* perusahaan, struktur halaman web tersebut, dan lain-lain. Masih banyak lagi kegunaan *text mining* yang dibutuhkan dalam kehidupan sehari-hari (Adiwijaya, 2006).

2.5.1 Pre-Processing

Pre-processing atau pemrosesan teks merupakan langkah awal untuk data yang akan diolah masuk pada proses klasifikasi terutama berfokus pada pembersihan data yang bertujuan untuk menghilangkan dan mengatasi *noisy data*, termasuk mengatasi informasi yang hilang atau tidak lengkap (Adiwijaya, 2006). Tahap ini bertujuan agar nantinya hasil perhitungan akan optimal. Berikut adalah langkah-langkah yang akan dilakukan dalam pemrosesan teks. *case folding, cleaning, translation, stemming, stopword removal* dan tokenisasi. Dengan dilakukannya proses tersebut, akan dihasilkan kata-kata dasar yang baku secara terpisah untuk mempermudah pada proses selanjutnya (terutama sata perhitungan). Tahap ini sangat membantu meningkatkan hasil yang akurat pada analisis sentimen (Zhao, 2015).

2.5.1.1 Case Folding

Pada tahap ini, dilakukan perubahan seluruh huruf kapital atau huruf besar menjadi huruf kecil (Indraloka & Santosa, 2017). Sebagai contoh: "Nama Saya Budi" dapat diubah menjadi "nama saya budi"

2.5.1.2 Cleaning

Pada tahap ini, dilakukan penghapusan karakter-karakter selain yang ditentukan seperti huruf atau karakter di luar dari alfabet a-z (termasuk tanda baca), menghapuskan URL atau *link*, menghapuskan *hashtag*, menghapuskan *username*. Biasanya, aturan dalam tiap penelitian berbeda-beda pada tahap

cleaning. Pada penelitian ini, dilakukan juga pergantian kalimat “batik air” menjadi “batikair”, “garuda Indonesia” menjadi “garudaindonesia”, “sriwijaya air” menjadi “sriwijayaair”. Sebagai contoh: “budi makan jengkol sebanyak 2 porsi” dapat diubah menjadi “budi makan jengkol sebanyak porsi” karena angka 2 dihapus.

2.5.1.3 Translation

Pada tahap ini, akan diterjemahkan suatu kalimat berbahasa Inggris ke Bahasa Indonesia untuk setiap katanya. Analisis sentimen saat ini sangat dinamis dalam bidang linguistik komputasi. Selain Bahasa Indonesia, terdapat berbagai bahasa yang digunakan dalam hal berkomunikasi di media sosial khususnya Twitter. Dari berbagai macam Bahasa, penulis memilih Bahasa Inggris yang akan diproses dengan diterjemahkan ke dalam Bahasa Indonesia dalam sistem yang dibuat. Sebelumnya, sempat ditemukan bahwa dengan memerhatikan multibahasa pada sistem akan meningkatkan hasil dari klasifikasi sentimen (Turchi, 2013). Sebagai contoh *translation*, kalimat “i like airplane” akan menjadi “saya suka pesawat”.

2.5.1.4 Stopword Removal

Pada tahap ini akan dilakukan penyaringan kata-kata yang sering maupun jarang muncul, biasa disebut dengan *stopword*. Proses ini disebut “*stopword removal*”. Dengan menghapus kata-kata yang jarang muncul tersebut, tampaknya akan menjadi optimal untuk mempertahankan kinerja klasifikasi sekaligus mengurangi data sparsial dan menyusutkan ruang fitur secara substansial. Sebagai contoh, kalimat “pesawat sangat nyaman” akan menjadi “pesawat nyaman” karena kata “sangat” terdapat pada *stopword list* (Saif, Fernandez, He, & Harith, 2014).

2.5.1.5 Stemming

Stemming akan merubah kata-kata dalam dokumen menjadi kata akar atau dasarnya (*root word*). Proses *stemming* pada dokumen Bahasa Indonesia cukup kompleks, karena harus dilakukan penghilangan seluruh imbuhan pada kata-kata yang terdapat pada *tweets*. Digunakan *library Sastrawi Stemming* berbahasa Indonesia yang berbasis algoritma Nazief dan Adriani. Sebagai contoh: “budi memarahi adiknya” dapat diubah menjadi “budi marah adik” (Afuan, 2013).

2.5.1.6 Tokenisasi

Pada tahap ini, dilakukan pemisahan setiap kata dalam suatu kalimat dalam dokumen. Memisahkan kata biasanya menggunakan spasi. Sebagai contoh: “budi marah adik” yang awalnya berada pada satu kalimat akan dipecah menjadi “budi”, “marah”, “adik”. Sebenarnya penulisan dapat berbeda-beda, namun tujuan utamanya adalah yaitu memotong kalimat berdasarkan tiap kata yang menyusun kalimat tersebut (Indraloka & Santosa, 2017).

2.5.2 Pembobotan Term Frequency-Inverse Document Frequency

Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah salah satu proses dari teknik ekstraksi fitur dengan proses memberikan nilai pada

masing-masing kata yang ada pada *tweets* latih (data latih). Untuk mengetahui seberapa penting sebuah kata mewakili sebuah kalimat, akan dilakukan pembobotan atau perhitungan. Pemberian skor dalam TF-IDF berdasarkan frekuensi munculnya kata dalam dokumen. Nilai TF-IDF dapat ditemukan dengan menggunakan Persamaan 2.1, 2.2 dan 2.3.

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

Pada Persamaan 2.1, dilakukan tahap awal dalam perhitungan TF-IDF yaitu perhitungan *term frequency* $tf_{t,d}$. t adalah term pada dokumen d yang berfungsi untuk menunjukkan jumlah munculnya term dalam tiap dokumen. Untuk dilanjutkan perhitungan dengan rumus seperti yang ditunjukkan Persamaan 2.2, dibutuhkan perhitungan df atau *document frequency* yang bertujuan untuk mengetahui jumlah dokumen yang mengandung sebuah *term*.

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

Setelah mendapatkan nilai df , pada Persamaan 2.2, dilanjutkan dengan proses perhitungan nilai *inverse document frequency* (idf). idf merupakan kebalikan dari hasil nilai df .

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

Setelah mendapatkan keseluruhan nilai, dilakukan perhitungan akhir dalam pembobotan TF-IDF. Pada Persamaan 2.3, dilakukan perkalian hasil $W_{tf_{t,d}}$ dengan idf_t yang sudah dihitung sebelumnya.

Keterangan:

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

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

N = jumlah seluruh dokumen

df = jumlah dokumen yang mengandung term

idf_t = bobot inverse dari nilai df

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

2.6 Lexicon Based Features

Lexicon Based Features adalah metode atau fitur yang digunakan pada sistem ini untuk mencocokkan kata-kata di data latih dengan kamus sentimen berisi kata positif dan kata negatif untuk diketahui tingkat polaritas tiap kata (Peng, 2011), sehingga dapat berfungsi sebagai pengujian. *Lexicon* merupakan suatu himpunan yang telah diketahui sentimennya (Desai & Mehta, 2016). Penerapan *Lexicon Based Features* akan meningkatkan akurasi dengan membantu penambahan bobot dari dokumen yang bersentimen positif maupun negatif.

Perhitungan bobot yang dilakukan dengan cara menghitung jumlah kata berdasarkan tiap fitur yang sebelumnya telah diseleksi terlebih dahulu. Kata-kata

yang diberi bobot akan dicocokkan dengan kamus, dan kemudian setiap fitur yang telah diberi bobot dapat digunakan pada proses selanjutnya.

2.6.1 Normalisasi *Min-Max*

Tujuan melakukan normalisasi data adalah untuk mengurangi kesalahan yang terdapat dalam proses data *mining* (Wirawan & Eksistyanto, 2015). Menurut penelitian yang dilakukan sebelumnya, digunakan nilai *newmax* sebesar 0.9 dan *newmin* sebesar 0.1 untuk pembobotan *Lexicon* menggunakan normalisasi *min-max* (Rofiqoh, 2017). Rumus perhitungan yang digunakan ditunjukkan pada Persamaan 2.4.

$$vi' = \frac{vi - min_a}{max_a - min_a} (newmax - newmin) + newmin \quad (2.4)$$

Keterangan:

vi'	= hasil dari proses normalisasi data ke-i
vi	= data ke-i yang dilakukan normalisasi
min_a	= data minimum dari seluruh data a
max_a	= data maksimum dari seluruh data a
$newmax$	= nilai maksimum dari normalisasi
$newmin$	= nilai minimum dari normalisasi

2.7 Klasifikasi

Klasifikasi, yaitu sebuah proses pemberian kelas pada setiap masukan atau input berupa dokumen (*tweets*) sesuai aturan. Aturan tersebut berdasarkan metode atau algoritme yang telah dipilih, karena tentunya akan berbeda-beda untuk setiap metode. Algoritme yang dipilih yaitu *Support Vector Machine* (SVM). Hasil dari klasifikasi SVM terbagi menjadi dua kelas, positif dan negatif.

2.7.1 *Support Vector Machine* (SVM)

Setelah melakukan pemrosesan teks dan teknik ekstraksi fitur *Term Frequency-Inverse Document Frequency*, yaitu saatnya untuk mengelompokkan atau mengklasifikasi *tweets* ke dalam beberapa kelas. Metode yang digunakan adalah *Support Vector Machine* (SVM). Dalam penelitian ini, terdapat dua kelas: kelas positif dan kelas negatif. Analisis sentimen merupakan klasifikasi biner dan sangat dianjurkan untuk menggunakan SVM yang mampu melakukan klasifikasi untuk dataset yang besar. Diawal, akan dilakukan pelabelan secara manual terlebih dahulu untuk data latih yang pada dasarnya data latih tersebut adalah hubungan antara X dengan Y yang mana X adalah skor dari ulasan atau opini dan Y adalah positif atau negative dari ulasan tersebut. Dilakukan pula perhitungan *hyperplane* pada SVM.

Support Vector Machine (SVM) termasuk *machine learning* (*supervised learning*) yang dapat memprediksi kelas berdasarkan dari hasil proses pelatihan. Dengan melakukan pelatihan menggunakan data masukan dalam bentuk numerik

dan hasil dari ekstraksi fitur didapatkan sebuah pola yang nantinya akan digunakan dalam proses pelabelan. Nilai atau pola yang dihasilkan dari Metode *Support Vector Machine* sebenarnya adalah sebuah garis pemisah yang disebut dengan *hyperplane*, yang mana garis tersebut berperan dalam memisahkan *tweet* dengan sentimen positif (berlabel 1) dengan *tweet* yang memiliki sentimen negatif (berlabel 0). Rumus perhitungan *hyperplane* ditunjukkan pada Persamaan 2.5.

$$f: w \cdot x + b = 0 \quad (2.5)$$

Keterangan:

w = parameter *hyperplane* yang dicari (garis yang tegak lurus antara garis *hyperplane* dan titik *support vector*)

x = data input SVM (x_1 = index kata, x_2 = bobot kata)

b = parameter *hyperplane* yang dicari (nilai bias)

f = fungsi *hyperplane*

Dalam mengambil keputusan dengan metode SVM digunakan fungsi kernel $K(x_i, x_d)$. Terdapat beberapa persamaan kernel seperti linier, polinomial, *radial basis function* (RBF), dan *sigmoid*. Pada penelitian ini akan digunakan persamaan kernel polinomial yang ditunjukkan pada Persamaan 2.6.

$$K(x_i, x_d) = (X_i^T X_{j+1})^d, \gamma > 0 \quad (2.6)$$

Di bawah ini adalah langkah-langkah pelatihan dalam *Support Vector Machine* menggunakan *Sequential Learning*. *Sequential Learning* merupakan algoritme sederhana untuk memproses data latih dari SVM yang digunakan dengan waktu singkat dibandingkan beberapa algoritme lainnya seperti *Quadratic Programming* yang lebih kompleks dengan membutuhkan waktu yang lebih lama dan *Sequential Minimal Optimization* yang merupakan pengembangan dari *Quadratic Programming* yang tentunya lebih kompleks, membutuhkan waktu yang lama, serta tidak terjamin menghasilkan optimasi tinggi (Vijayakumar, 1999).

1. Inisialisasi parameter: a , γ , C , dan ϵ

Keterangan:

α = alfa

γ = konstanta gamma.

C = variabel *slack*.

ϵ = epsilon

2. Hitung matriks Hessian, persamaan ditunjukkan pada Persamaan 2.7.

$$D_{ij} = y_i y_j (K(x_i x_j))^2 + \lambda^2 \quad (2.7)$$

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
- (x_i, x_j) = fungsi kernel
- λ = lamda

3. Melakukan 3 perhitungan sebagai berikut (sampai batas interasi).

$$a. E_i = \sum_{j=1}^i a_j D_{ij} \tag{2.8}$$

Keterangan:

- a_j = alfa ke-j
- D_{ij} = matriks Hessian
- E_i = error rate

$$b. \vartheta a_i = \min(\max[\gamma(1 - E_i), a_i], C - a_i) \tag{2.9}$$

Keterangan:

- a_i = alfa ke-i
- γ = konstanta gamma
- E_i = error rate
- C = variabel slack

$$c. a_i = a_i + \vartheta a_i \tag{2.10}$$

- a_i = alfa ke-i
- ϑa_i = delta alfa ke-i

4. Akan didapatkan nilai $SV = (\alpha_i > \text{thresholdSV})$, kemudian melakukan perhitungan nilai bias yang ditunjukkan pada Persamaan 2.11.

$$b = -\frac{1}{2} (\sum_{i=1}^N a_i y_i K(x_i, x^-) + \sum_{i=1}^N a_i y_i K(x_i, x^+)) \tag{2.11}$$

5. Melakukan perhitungan fungsi $f(x)$,

$$f(x) = \sum_{i=1}^m a_i y_i K(x_i, x) + b \tag{2.12}$$

2.8 Confusion Matrix

Confusion Matrix merupakan teknik yang digunakan untuk mengevaluasi klasifikasi model untuk memperkirakan objek yang benar atau salah. Sebuah matriks dari prediksi akan dibandingkan dengan kelas asli yang berisi informasi aktual dan prediksi nilai klasifikasi. Setelah sistem berhasil mengklasifikasikan *tweet*, dibutuhkan ukuran untuk menentukan seberapa *valid* atau tepat klasifikasi telah dibuat oleh sistem. Tabel 2.2 menunjukkan *confusion matrix* yang digunakan untuk membantu dalam perhitungan sistem evaluasi (Tiara, Sabariah, & Effendy,



2015). Pengujian akurasi dalam pengujian ini menggunakan *confusion matrix* empat kondisi sebagai berikut. *True Positive* (TP), *True Negatif* (TN), *False Positif* (FP), dan *False Negative* (FN).

Tabel 2.2 Confusion Matrix

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

Keterangan:

- *True positive* (TP) adalah banyaknya dokumen yang prediksi kelasnya bernilai positif dan kelas aktualnya bernilai positif.
- *False negative* (FN) adalah banyaknya dokumen yang diprediksi menjadi kelas negatif oleh sistem, namun kelas aktual dari dokumen adalah positif.
- *False positive* (FP) adalah banyaknya dokumen yang diprediksi sebagai kelas positif oleh sistem, namun kelas aktualnya adalah negatif.
- *True negative* (TN) adalah banyaknya dokumen kelas yang diberikan oleh sistem dan kelas aktualnya bernilai sama, yaitu negatif.

Selanjutnya, terdapat perhitungan yang dilakukan seperti *Accuracy*, *Precision*, *Recall*, dan *F-Measure* yang dapat dilihat pada Persamaan 2.13, 2.14, 2.15, dan 2.16.

$$Accuracy = \frac{TN+TP}{TN+TP+FP+FN} \quad (2.13)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.14)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.15)$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (2.16)$$

Dengan keterangan sebagai berikut.

- *Accuracy* adalah perhitungan seluruh keadaan yang diprediksikan dengan nilai yang benar terhadap seluruh keadaan yang diprediksi.
- *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.
- *F-Measure* merupakan perhitungan yang melibatkan *precision* dan *recall* untuk dicari nilai tengah pada kedua evaluasi tersebut.

BAB 3 METODOLOGI PENELITIAN

3.1 Tipe Penelitian

Tipe penelitian pada penelitian ini adalah non-implementatif analitik. Pada penelitian ini dilakukan analisis terhadap hubungan antar fenomena yang sedang dikaji untuk menghasilkan suatu hasil analisis. Dilakukan juga beberapa teknik seperti survei, studi kasus, dan eksperimentasi.

3.2 Strategi Penelitian

Diawali dengan mengambil beberapa permasalahan yang terdapat di sekitar (dari referensi-referensi penelitian sebelumnya) kemudian memfokuskan permasalahan tersebut untuk ditentukan sebuah objek penelitian. Setelah itu dapat dilihat manfaat dari penelitian yang akan dilakukan. Pada penelitian ini, objek yang digunakan adalah opini masyarakat mengenai maskapai penerbangan yang terdapat di media sosial Twitter.

Berdasarkan permasalahan-permasalahan yang sudah ditemukan dan ditentukan, akan dicari referensi atau sumber pustaka (seperti jurnal, buku, dan sebagainya) untuk membantu penelitian. Dimulai dengan mencari penelitian-penelitian sebelumnya yang relevan dengan masalah dan topik pada penelitian ini. Topik penelitian ini adalah analisis sentimen. Selanjutnya, dicari tahu mengenai tren dari topik tersebut, permasalahan apa yang sering muncul. Setelah ditemukan referensi yang tepat, akan segera dilakukan penelitian dengan metode yang diinginkan peneliti, tentunya dengan sebuah metode/algorithm dan fitur ekstraksi yang berbeda dari sebelum-sebelumnya. Sebelumnya, akan dilakukan terlebih dahulu tahap wawancara terhadap pakar-pakar yang bersangkutan dengan permasalahan yang diangkat untuk memperkuat dasar penelitian. Teori pendukung yang dibahas pada penelitian ini adalah:

- Analisis Sentimen.
- *Text mining*.
- Media sosial.
- Maskapai Penerbangan.
- Klasifikasi.
- *Algoritme Support Vector Machine*.

Setelah itu, dilakukan pengumpulan data berupa data latih dan data uji yang akan diambil dari opini masyarakat dari media sosial Twitter dalam bentuk *tweets* (posting dari pengguna Twitter). Akan diambil dengan proses *Crawling* menggunakan sebuah program. Setelah itu, akan dilakukan pelabelan manual untuk tiap opini tersebut (positif atau negatif).

Setelah melakukan pemrosesan teks, tahap awal sebelum melakukan proses klasifikasi adalah melakukan pembobotan pada kata dengan menggunakan

metode *Term Frequency-Inverse Document Frequency*. Setelah tiap-tiap kata sudah diberi bobot, akan dilakukan proses pelatihan menggunakan algoritme *Support Vector Machine* untuk mengetahui hasil klasifikasi sehingga dapat diketahui analisis sentimen yang ingin diketahui, dan yang terakhir akan dilakukan proses pengujian atau evaluasi dengan *confusion matrix*.

Peralatan pendukung dalam penelitian ini adalah sebagai berikut.

- Spesifikasi perangkat keras:

- Komputer Core i7
- RAM 4GB

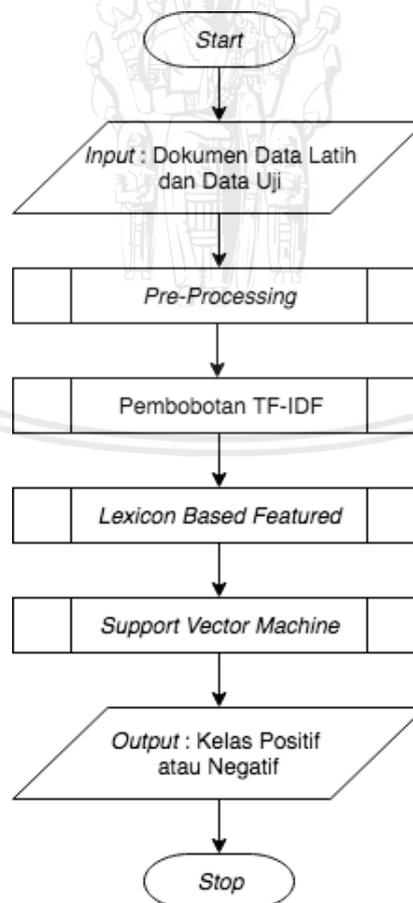
- Spesifikasi perangkat lunak:

- Windows 8.1.
- Python 2.6.7
- Microsoft Excel.

- Spesifikasi data:

Data opini yang terdapat pada Twitter (*tweets*).

Gambaran umum sistem dapat dilihat dalam Gambar 3.1.



Gambar 3.1 Deskripsi Umum Sistem

3.3 Partisipan Penelitian

Pakar bahasa dari Fakultas Ilmu Budaya akan dibutuhkan sebagai partisipan penelitian ini. Pakar Bahasa akan membantu dalam pelabelan manual tiap-tiap opini masyarakat terhadap maskapai penerbangan yang terdapat pada dokumen Twitter. Pakar Bahasa adalah mahasiswi Fakultas Ilmu Budaya Universitas Brawijaya dengan identitas sebagai berikut.

Nama: Anisa Azahra Pramudya

NIM: 155110101111054

Jurusan: Sastra Inggris

Kemampuan: Linguistics

3.4 Lokasi Penelitian

Lokasi penelitian dilakukan di dua lokasi, antara Fakultas Ilmu Budaya Universitas Brawijaya, dan Laboratorium Riset Komputasi Cerdas Fakultas Ilmu Komputer Brawijaya. Di lokasi pertama akan dilakukan wawancara terhadap pakar bahasa terkait pelabelan manual dengan mengklasifikasikan opini menjadi positif dan negatif. Di lokasi kedua, dilakukan pengambilan data dan implementasi algoritme *Support Vector Machine* pada objek opini terhadap maskapai penerbangan pada dokumen Twitter.

3.5 Teknik Penelitian Data

Pengumpulan data berupa data latih dan data uji akan diambil dari opini masyarakat dari media sosial Twitter dalam bentuk *tweets* (posting dari pengguna Twitter). Akan diambil dengan proses *tweet crawling* menggunakan sebuah program dengan bahasa pemrograman Python. Waktu pengambilan data adalah pada bulan Agustus dan September tahun 2018. Kata kunci yang dicari antara lain adalah *tweets* pengguna Twitter yang melakukan *mention* pada akun-akun sebagai berikut. @BatikAirINA, @IndonesiaGaruda, @Citilink, @LionAirID, dan @SriwijayaAir. Selain itu, akan dilakukan tahap wawancara yang akan dilakukan terhadap pakar bahasa sebagai sumber pelabelan manual untuk data yang sudah terkumpul.

3.6 Data Penelitian

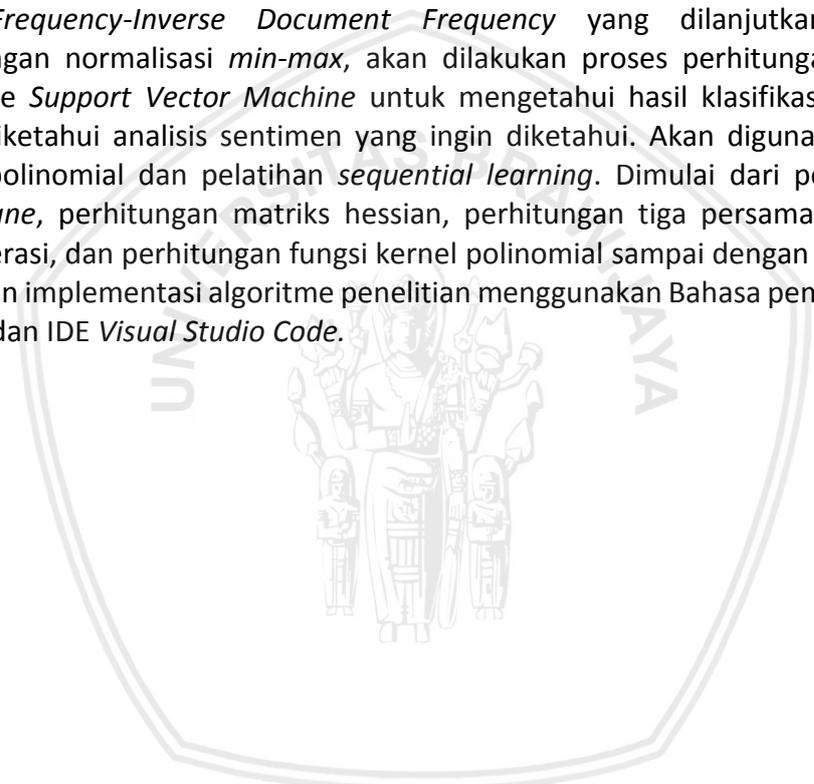
Data yang digunakan dalam penelitian ini adalah opini masyarakat tentang beberapa maskapai penerbangan yang terkenal dan banyak digunakan masyarakat, antara lain seperti Garuda Indonesia, Lion Air, Batik Air, Sriwijaya Air dan Citilink Indonesia. Data diambil dari dokumen Twitter berbahasa Indonesia dan Inggris yang ditulis pengguna Twitter untuk akun @IndonesiaGaruda, @LionAirGroup, @BatikAirINA, @SriwijayaAir, dan @Citilink. Jumlah data sebanyak 250 data yang terdiri dari 2 kelas yaitu 125 data positif dan 125 data negatif. 200 data digunakan sebagai data latih dan 50 data digunakan sebagai data uji.

3.7 Teknik Analisis Data

Teknik analisis data dapat dilakukan dengan perbandingan dan beberapa perhitungan dari proses pengujian atau evaluasi untuk melihat hasil penelitian (terutama untuk metode). Seperti mencocokkan hasil prediksi analisis sentimen dari sistem dengan hasil prediksi analisis sentimen saat proses pelabelan manual. Pengujian yang dilakukan adalah *Confusion Matrix* empat kondisi: *True Positive* (TP), *True Negatif* (TN), *False Positif* (FP), dan *False Negative* (FN). Selanjutnya, melakukan perhitungan *Accuracy*, *Precision*, *Recall*, dan *F-Measure*.

3.8 Implementasi Algoritme Penelitian

Setelah melakukan pemrosesan teks dan pembobotan pada kata menggunakan *Term Frequency-Inverse Document Frequency* yang dilanjutkan dengan perhitungan normalisasi *min-max*, akan dilakukan proses perhitungan dengan algoritme *Support Vector Machine* untuk mengetahui hasil klasifikasi sehingga dapat diketahui analisis sentimen yang ingin diketahui. Akan digunakan fungsi kernel polinomial dan pelatihan *sequential learning*. Dimulai dari perhitungan *hyperplane*, perhitungan matriks hessian, perhitungan tiga persamaan sampai batas iterasi, dan perhitungan fungsi kernel polinomial sampai dengan pengujian. Dilakukan implementasi algoritme penelitian menggunakan Bahasa pemrograman Python dan IDE *Visual Studio Code*.



BAB 4 PERANCANGAN DAN IMPLEMENTASI

4.1 Deskripsi Permasalahan

Seiring berjalannya waktu, bersamaan dengan peningkatan penggunaan media sosial Twitter sebagai sumber komunikasi dan informasi antara para pelanggan dan sebuah perusahaan, masyarakat menggunakan media sosial Twitter dengan menjadikannya referensi dalam membantu memberikan jawaban bagi masyarakat yang sedang mencari atau membutuhkan sesuatu berdasarkan pendapat orang lain, selain itu masyarakat dapat membantu pihak perusahaan dalam meningkatkan kualitas kinerja maupun pelayan dari sebuah perusahaan dengan memberikan kritik maupun apresiasi di media sosial Twitter.

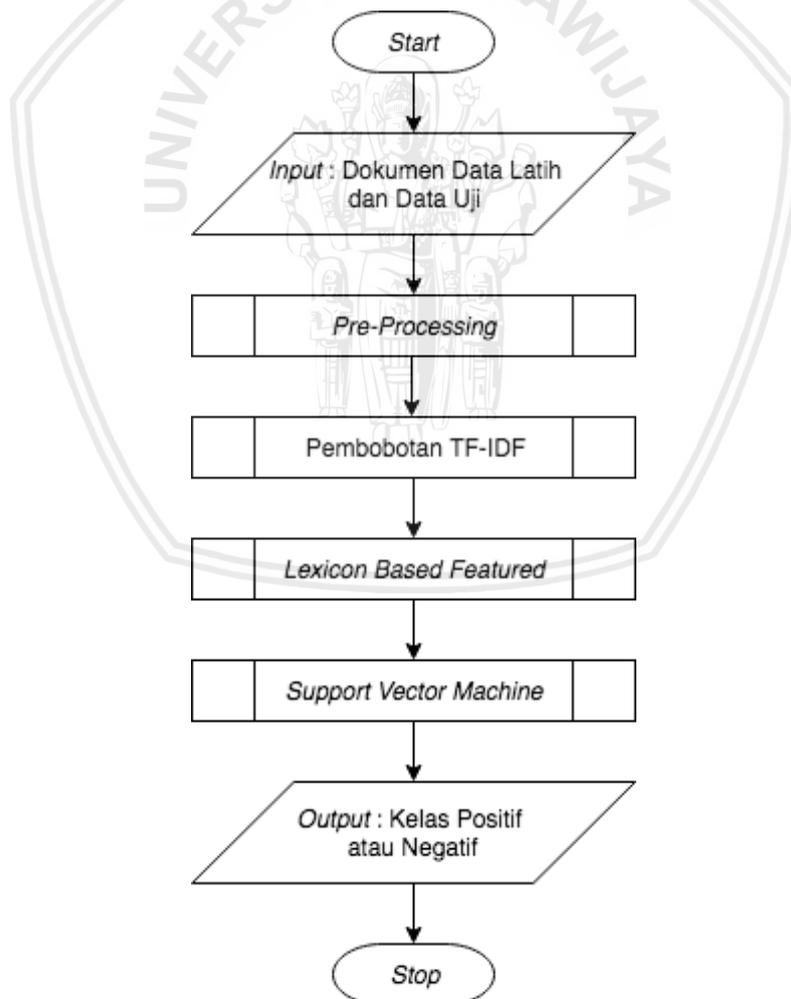
Twitter memungkinkan pengguna untuk mengekspresikan opini dan perasaan mereka mengenai banyak isu atau permasalahan. Hal mengenai transportasi sedang marak diperbincangkan di kalangan masyarakat, tidak hanya untuk bertanya-tanya seputar kebutuhan transportasi pelanggan, namun juga sering digunakan untuk memberikan pendapat serta menjadikan pendapat-pendapat pengguna lain menjadi bahan informasi pengguna lainnya. Salah satu transportasi yang menggunakan Twitter sebagai wadah komunikasi dan informasi antara pelanggan dan perusahaan adalah transportasi udara. Berbeda dengan media sosial yang lain yang harus menjadi teman terlebih dahulu baru dapat berinteraksi, Twitter memungkinkan antar pengguna tetap terhubung walaupun mereka tidak saling berteman. Bagi pengguna yang tidak melakukan registrasi akun maka hanya mendapat fasilitas untuk membaca *tweet* saja, namun bagi pengguna yang mendaftarkan akunnya mereka dapat membaca serta menulis *tweet*. Twitter juga merupakan media sosial yang bekerja secara *real-time* yaitu waktu terjadinya aktual, saat itu juga. Oleh karena itu, Twitter memungkinkan pengguna Twitter dapat melontarkan pendapat-pendapatnya dan membaca pendapat seluruh masyarakat mengenai transportasi udara di waktu yang tidak menentu, dapat dilakukan dan ditemukan kapan saja.

Untuk mengetahui opini masyarakat pengguna Twitter khususnya pada pemberian opini tentang maskapai penerbangan, maka diperlukan suatu analisis sentimen pada opini-opini yang terdapat pada media sosial Twitter yang telah disampaikan oleh para pengguna dengan *posting tweets* dengan menuliskan pendapat dan *me-mention* akun resmi dari beberapa maskapai penerbangan seperti Garuda Indonesia, Citilink, Sriwijaya, Lion Air, dan Batik Air. Sentimen yang digunakan adalah sentimen positif untuk teks yang mengandung kata-kata positif yang menunjukkan pemberian apresiasi dari masyarakat untuk maskapai penerbanga dan sentimen negatif untuk teks yang mengandung kata-kata negatif yang biasanya berisikan keluhan pelanggan terhadap maskapai penerbangan. Untuk mengetahui hasil analisis sentimen dari opini pada dokumen Twitter, akan diimplementasikan algoritme *Support Vector Machine* yang berfungsi dalam mencari *hyperplane* atau fungsi pemisah antar kelas data yang dikolaborasikan dengan pembobotan TF-IDF.

4.2 Deskripsi Umum Sistem

Analisis sentimen opini tentang maskapai penerbangan pada dokumen twitter dengan algoritme klasifikasi *Support Vector Machine* adalah suatu sistem yang dikembangkan untuk membantu menganalisis opini-opini pada Twitter yang mengandung sentimen positif atau negatif. Data yang digunakan pada penelitian ini adalah data dari dokumen Twitter yang telah diambil yang terdiri dari data latih dan data uji.

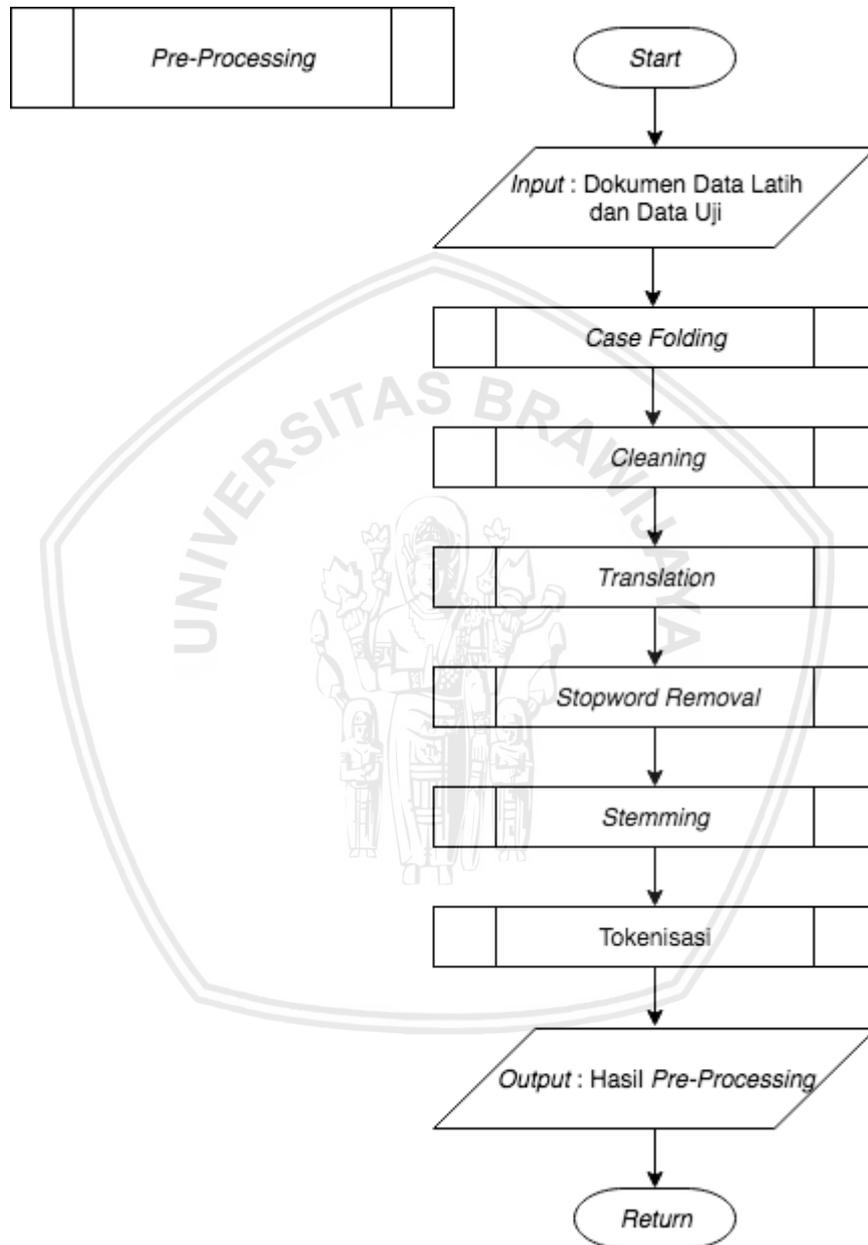
Gambaran umum sistem yang akan diimplementasikan, akan diawali dengan dilakukannya tahap *pre-processing text* untuk data latih dan data uji yang bertujuan untuk mengolah data agar dapat dianalisis pada algoritme *Support Vector Machine*. Setelah itu, akan didapatkan data hasil dari *pre-processing* yang akan dilakukan pembobotan kata dengan menggunakan teknik ekstraksi fitur TF-IDF. Hasil perhitungan dari TF-IDF akan masuk pada tahap klasifikasi menggunakan algoritme *Support Vector Machine* yang memberikan hasil sentimen yang menunjukkan apakah sebuah data akan masuk ke dalam kelas positif atau kelas negatif. Alur proses dari deskripsi umum sistem dapat dilihat dalam Gambar 4.1.



Gambar 4.1 Deskripsi Umum Sistem

4.3 Pre-Processing

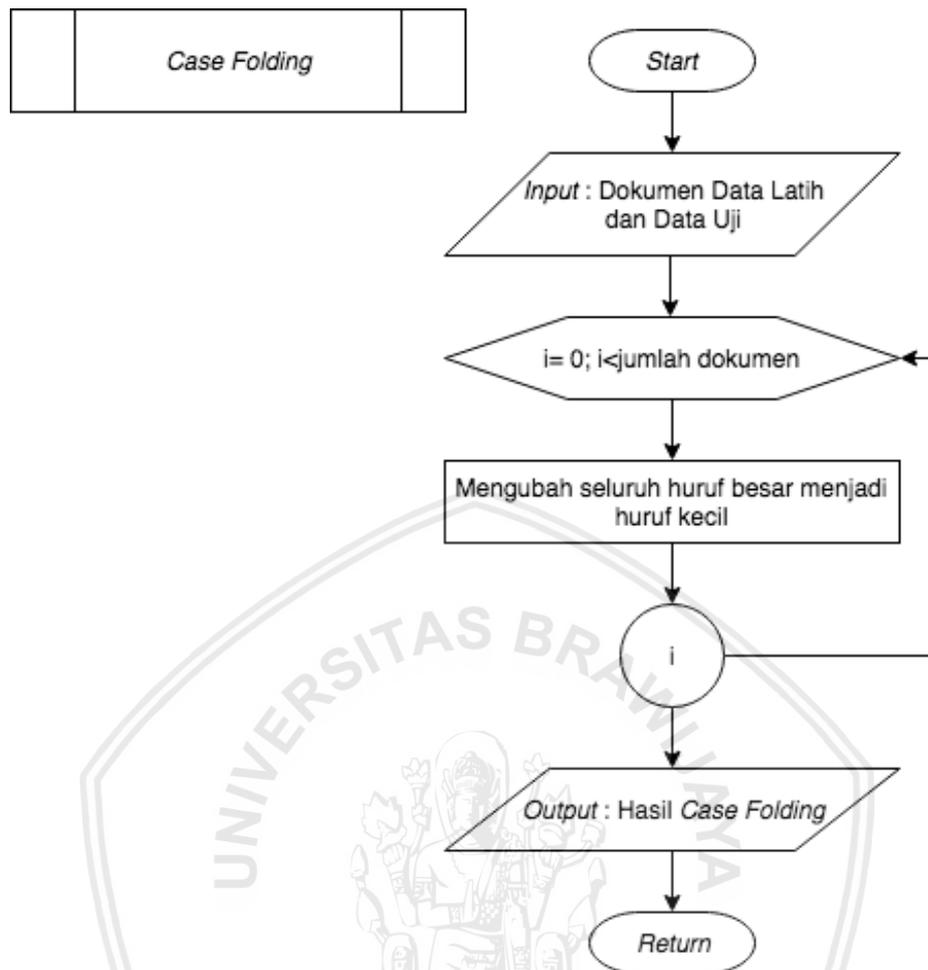
Pre-processing atau pemrosesan teks adalah langkah awal untuk data yang akan diolah masuk pada proses klasifikasi. Berikut adalah langkah-langkah yang akan dilakukan dalam pemrosesan teks. *case folding*, *cleaning*, *translation*, *stemming*, dan tokenisasi. Alur *pre-processing* dapat dilihat dalam Gambar 4.2.



Gambar 4.2 Alur *Pre-Processing*

4.3.1 Case Folding

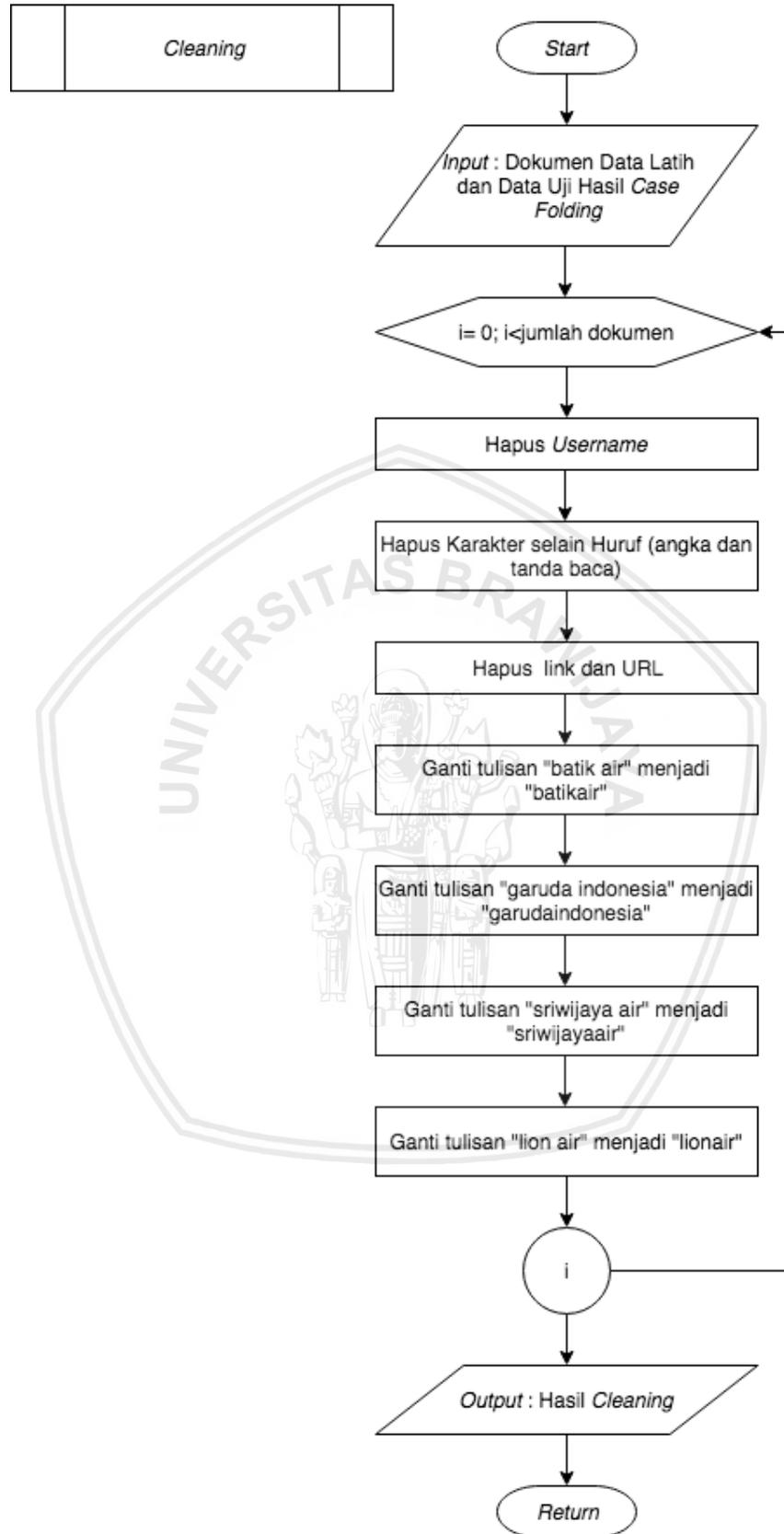
Pada tahap ini, dilakukan perubahan seluruh huruf kapital menjadi huruf kecil. *Case folding* merupakan tahap paling utama dalam *pre-processing* yang kemudian hasilnya akan dilanjutkan ke tahap *pre-processing* selanjutnya. Alur dari tahap *case folding* dapat dilihat dalam Gambar 4.3.



Gambar 4.3 Alur Case Folding

4.3.2 Cleaning

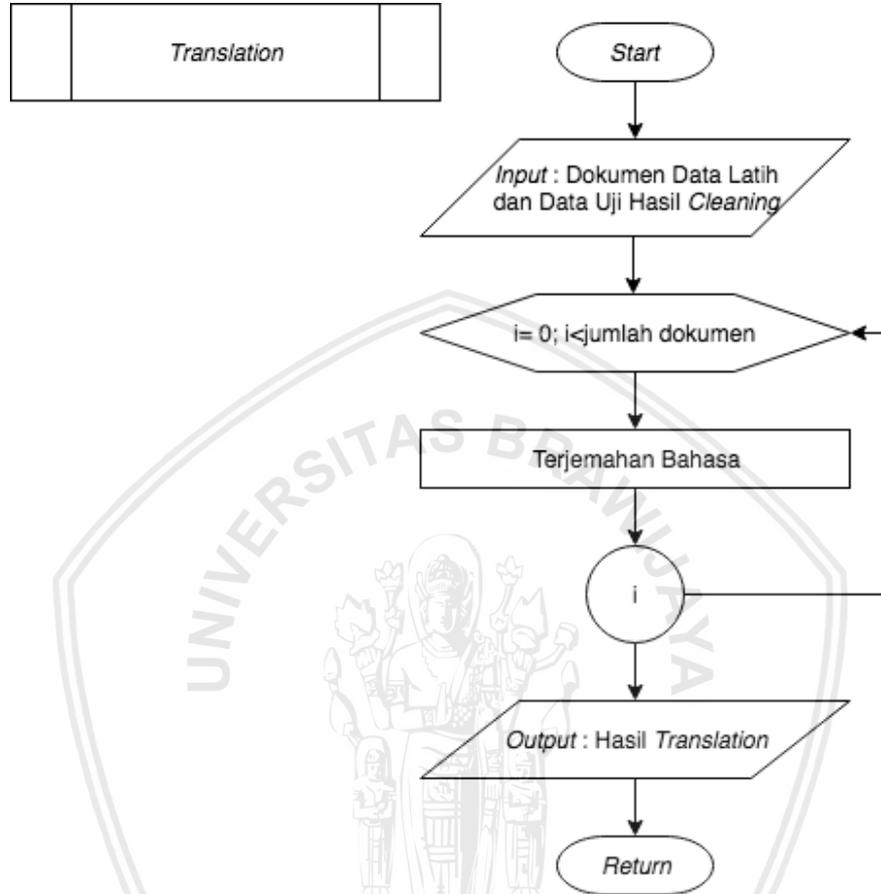
Pada tahap ini, dilakukan penghapusan karakter-karakter selain yang ditentukan seperti huruf atau karakter di luar dari alfabet a-z, menghapuskan URL, menghapuskan hashtag, menghapuskan username. Selanjutnya, dilakukan pergantian kalimat “batik air” menjadi “batikair”, “garuda Indonesia” menjadi “garudaindonesia”, “sriwijaya air” menjadi “sriwijayaair”. Alur dari tahap *cleaning* dapat dilihat dalam Gambar 4.4.



Gambar 4.4 Alur Cleaning

4.3.3 Translation

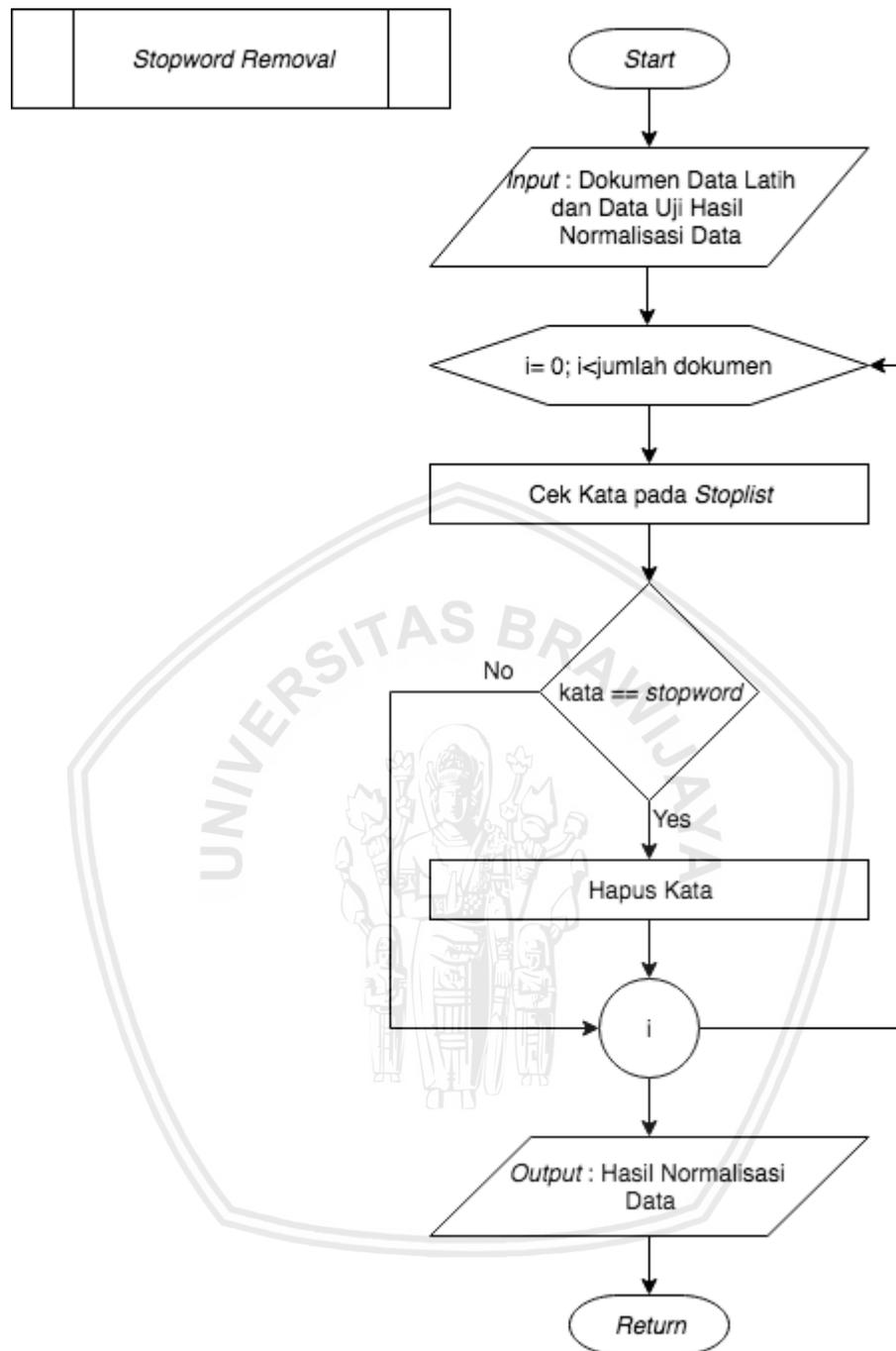
Pada tahap *translation*, akan diterjemahkan suatu kalimat berbahasa Inggris ke Bahasa Indonesia untuk setiap katanya. Akan digunakan bantuan dari Yandex *Translator* API. Alur pada tahap *translation* dapat dilihat dalam Gambar 4.5.



Gambar 4.5 Alur Translation

4.3.4 Stopword Removal

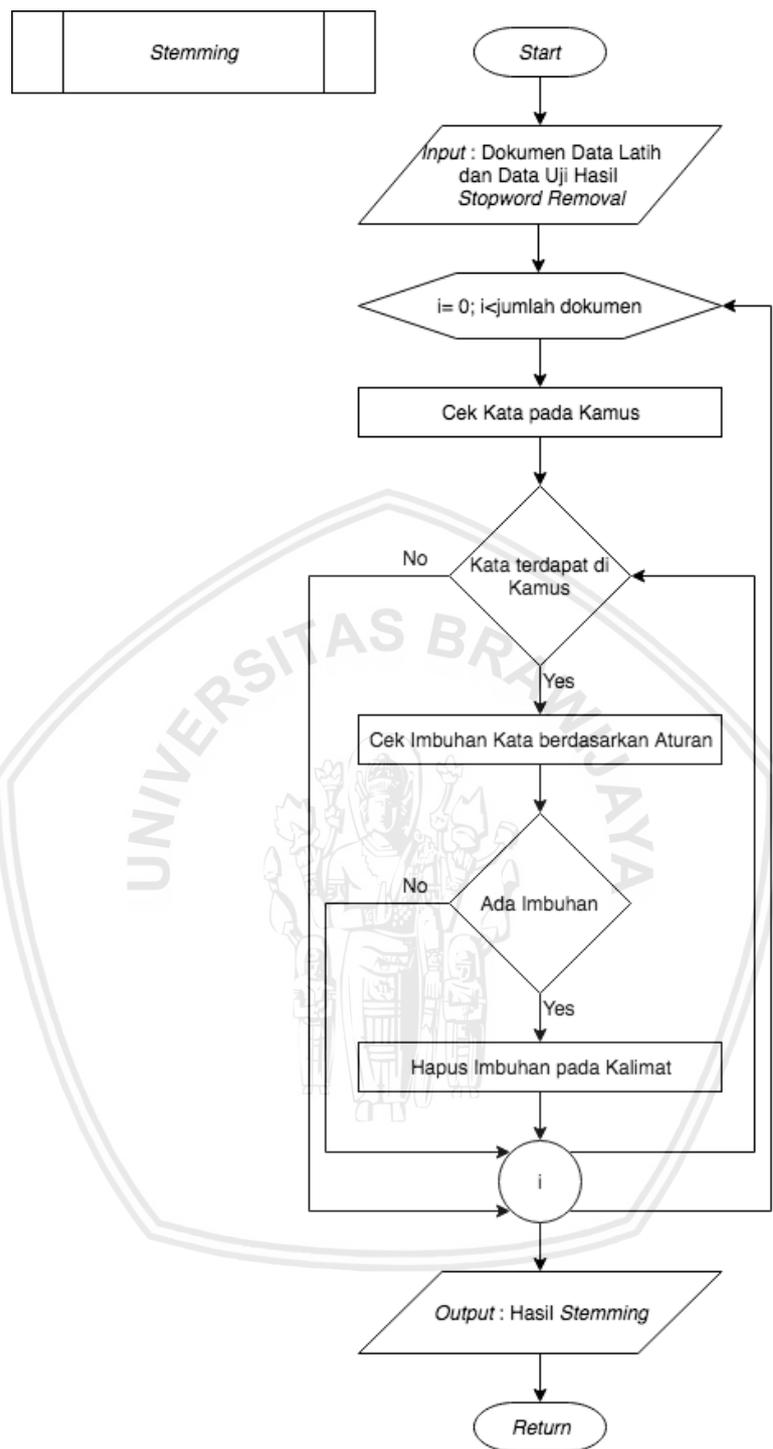
Pada tahap *stopword removal*, akan dilakukan penyaringan kata-kata yang sering maupun jarang muncul. Terdapat kumpulan kata-kata yang disebut *stopword list* yang akan diperiksa pada data latih dan data uji. Dengan menggunakan *stopword list* dari tala, alur pada tahap *stopword removal* dapat dilihat dalam Gambar 4.6.



Gambar 4.6 Alur Stopword Removal

4.3.5 Stemming

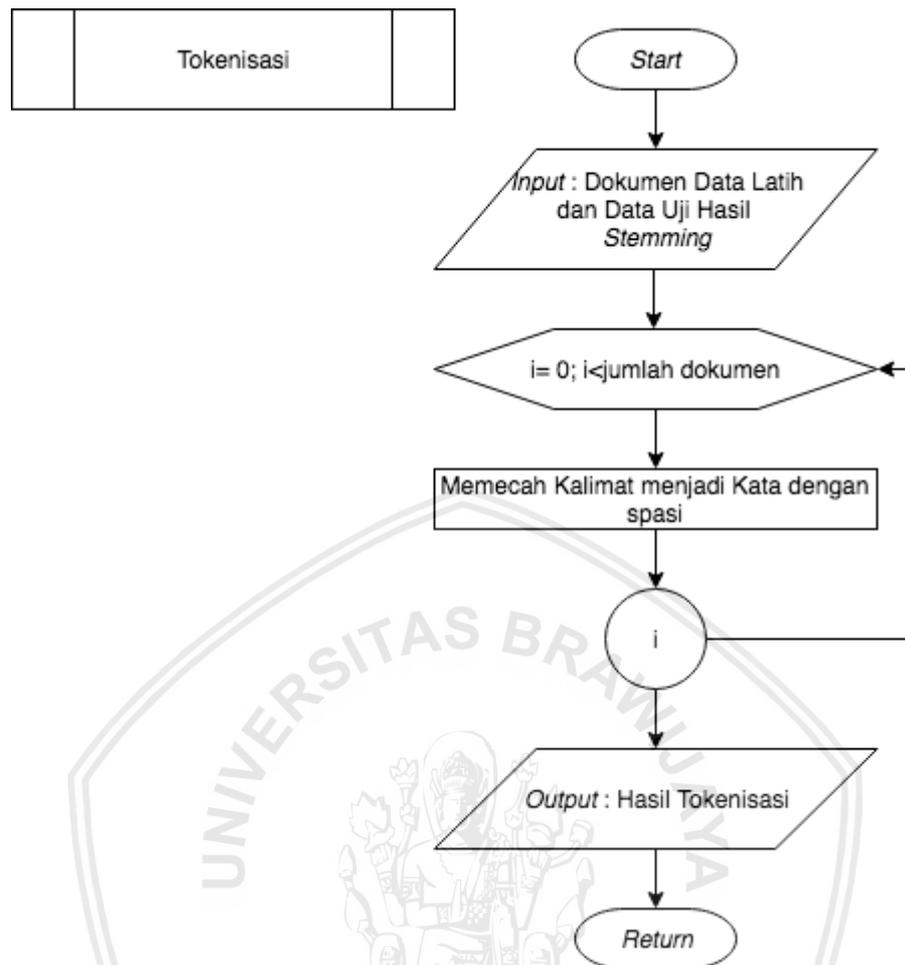
Pada tahap ini, dilakukan penghilangan seluruh imbuhan pada kata-kata yang terdapat pada *tweets*. *Output* dari tahap ini berupa kata dasar yang sebelumnya masih berimbuhan. Pada penelitian ini, tahap stemming menggunakan Sastrawi. Alur pada tahap *stemming* dapat dilihat dalam Gambar 4.7.



Gambar 4.7 Alur Stemming

4.3.6 Tokenisasi

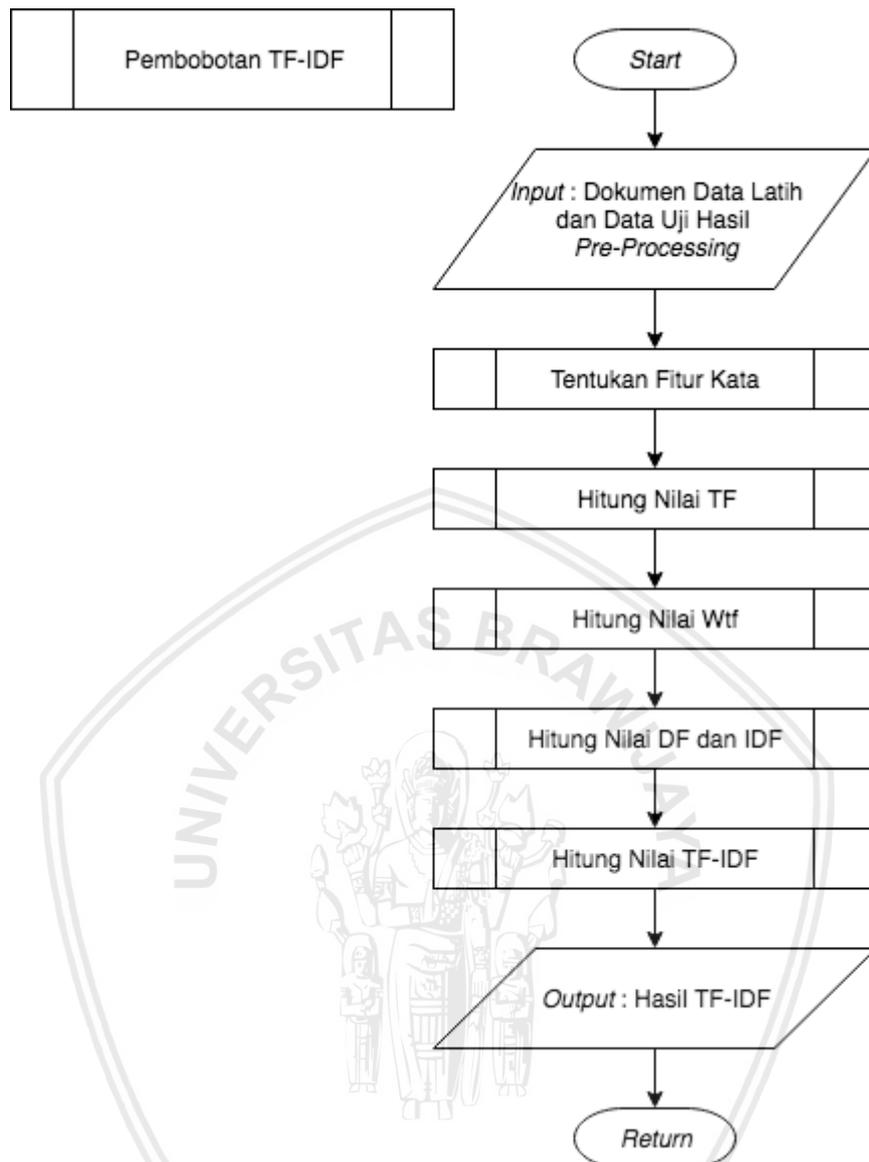
Pada tahap tokenisasi, dilakukan pemisahan setiap kata dalam suatu kalimat dalam dokumen. Tahapan ini merupakan tahapan akhir yang dilakukan oleh sistem. *Output* dari tahap tokenisasi adalah hasil tokenisasi yang disimpan selama perulangan dijalankan. Alur dari tahap ini dapat dilihat dalam Gambar 4.8.



Gambar 4.8 Alur Tokenisasi

4.4 Pembobotan TF-IDF

Hasil dari *pre-processing* akan diolah lebih lanjut. Pengolahan data dilakukan dengan menghitung bobot pada setiap kata atau *term*. Dari hasil *pre-processing* akan dicari fitur-fitur kata untuk dihitung frekuensi setiap kata (TF) dan dihitung nilai bobot frekuensi pada setiap kata (*Wtf*). Setelah itu, apabila frekuensi kata telah ditemukan akan dilakukan perhitungan dokumen yang mengandung sejumlah kata (DF) serta menghitung *inverse* dari nilai DF (*idft*). Kemudian akan dilakukan perkalian antara nilai TF dan IDF menjadi pembobotan TF-IDF. *Output* dari pembobotan tersebut akan digunakan untuk perhitungan di tahap klasifikasi dengan algoritme SVM. Alur proses pembobotan TF-IDF dapat dilihat dalam Gambar 4.9.

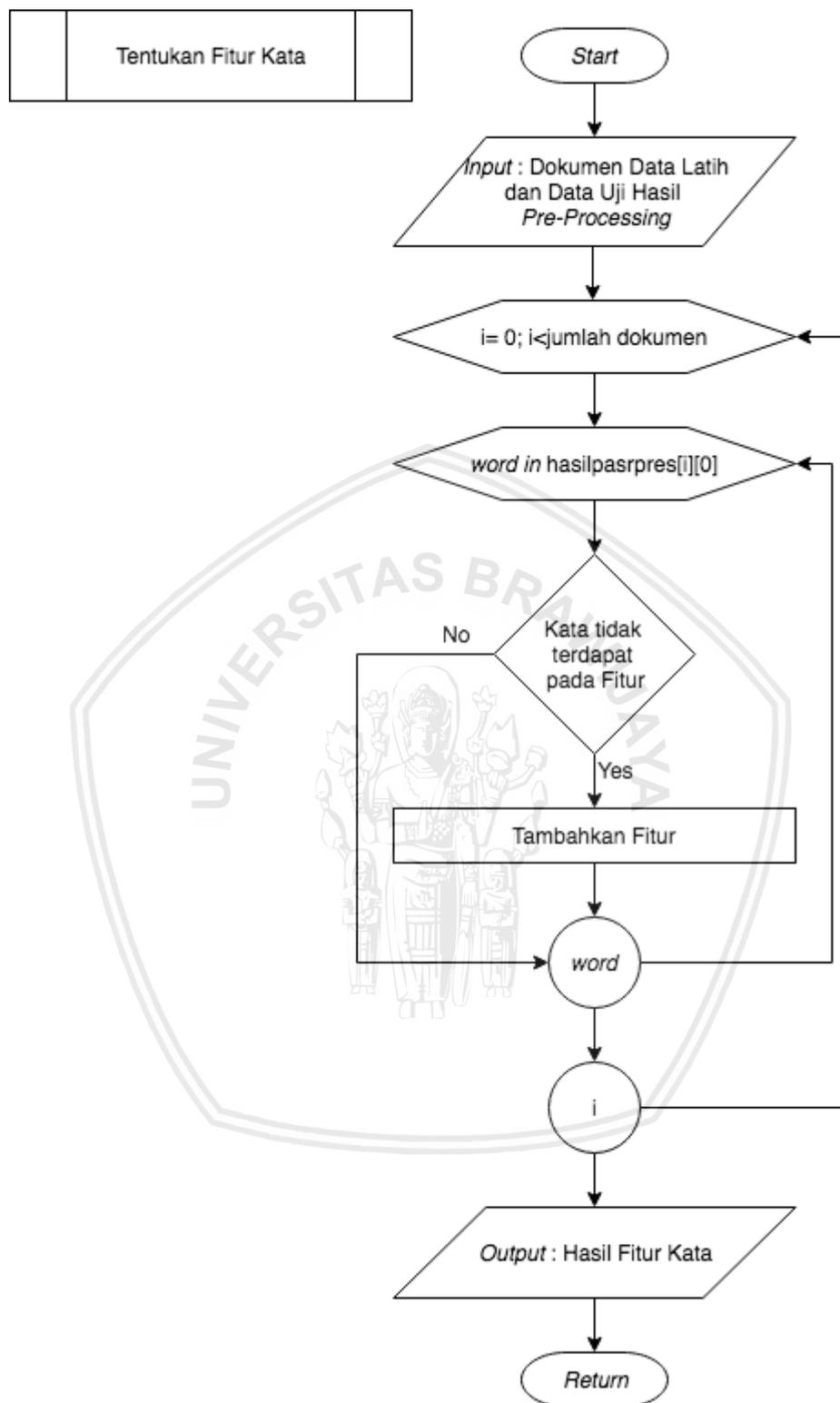


Gambar 4.9 Alur Pembobotan TF-IDF

4.4.1 Menentukan Fitur Kata

Menentukan fitur kata dilakukan untuk mendata kata-kata unik yang terdapat di dalam dokumen yang telah dimasukkan. Kata-kata tersebut akan dijadikan fitur untuk perhitungan TF-IDF. Alur dari menentukan fitur kata ditunjukkan dalam Gambar 4.10.



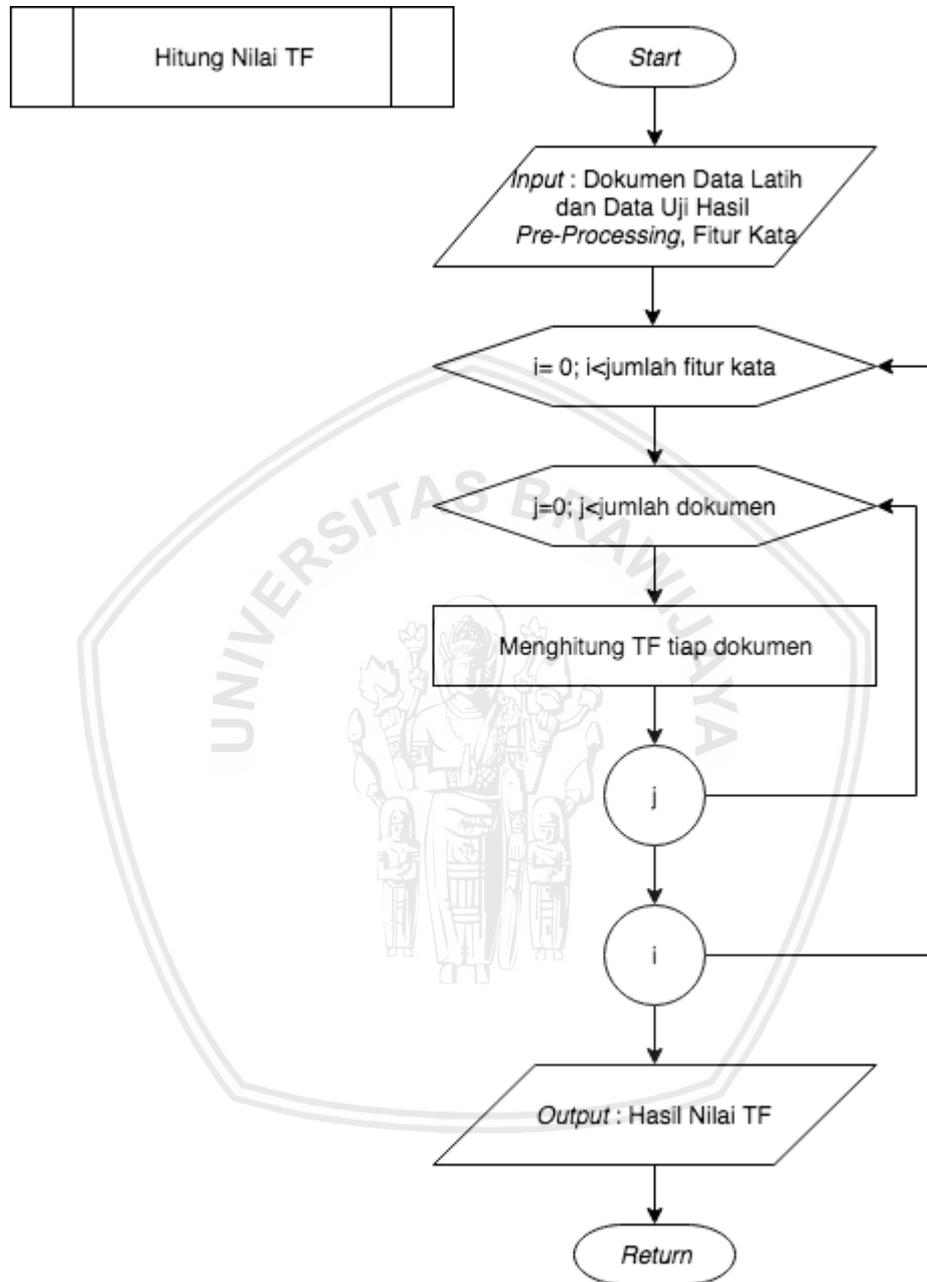


Gambar 4.10 Tentukan Fitur Kata

4.4.2 Menghitung Nilai TF

Dilakukan perulangan pada jumlah fitur kata yang telah ditentukan dan pada panjang dokumen yang dimasukkan. Jika terdapat kata (*term*) yang sama maka

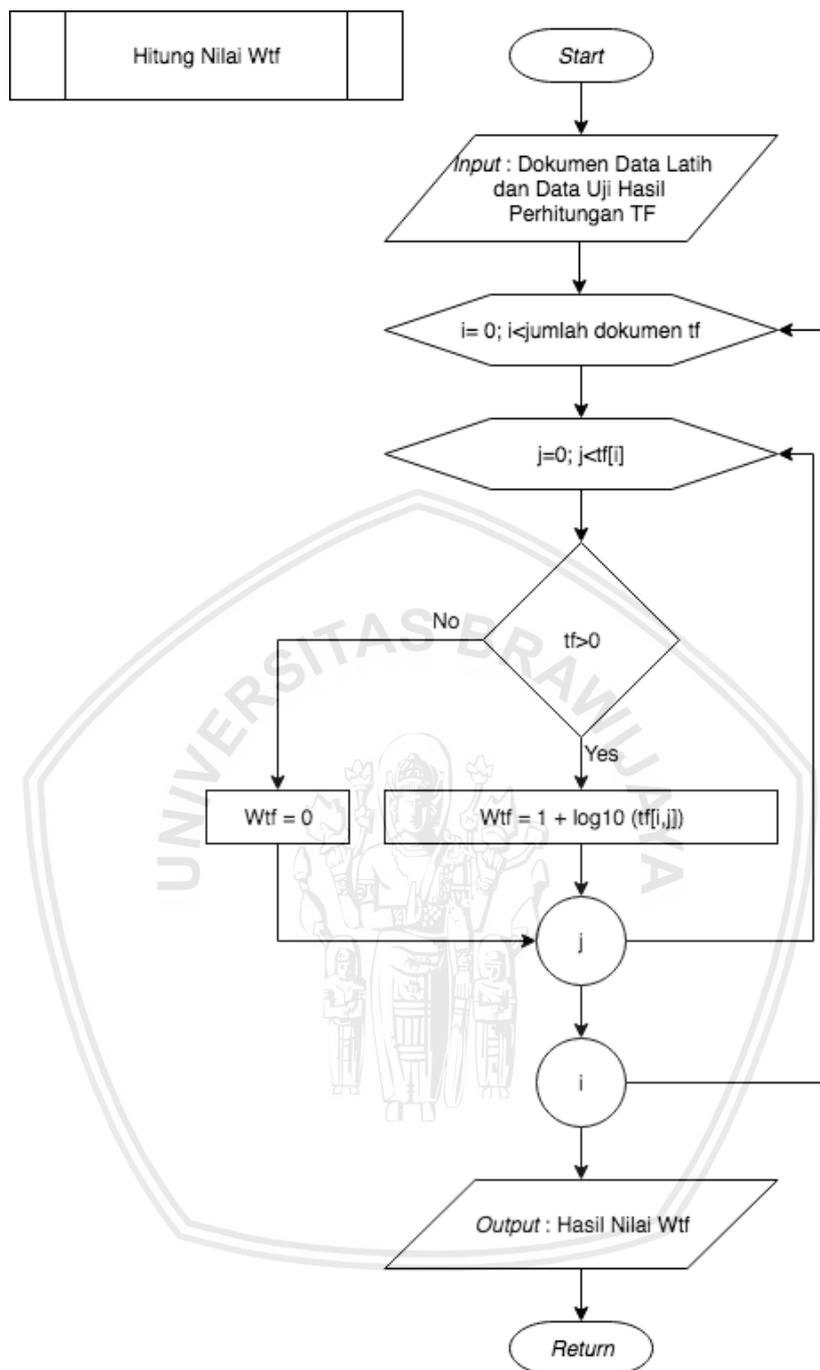
nilai TF akan bertambah seiring dengan jumlah frekuensinya. Alur perhitungan ditunjukkan dalam Gambar 4.11, data yang dimasukkan adalah hasil dari *pre-processing*.



Gambar 4.11 Alur Hitung Nilai TF

4.4.3 Menghitung Nilai Wtf

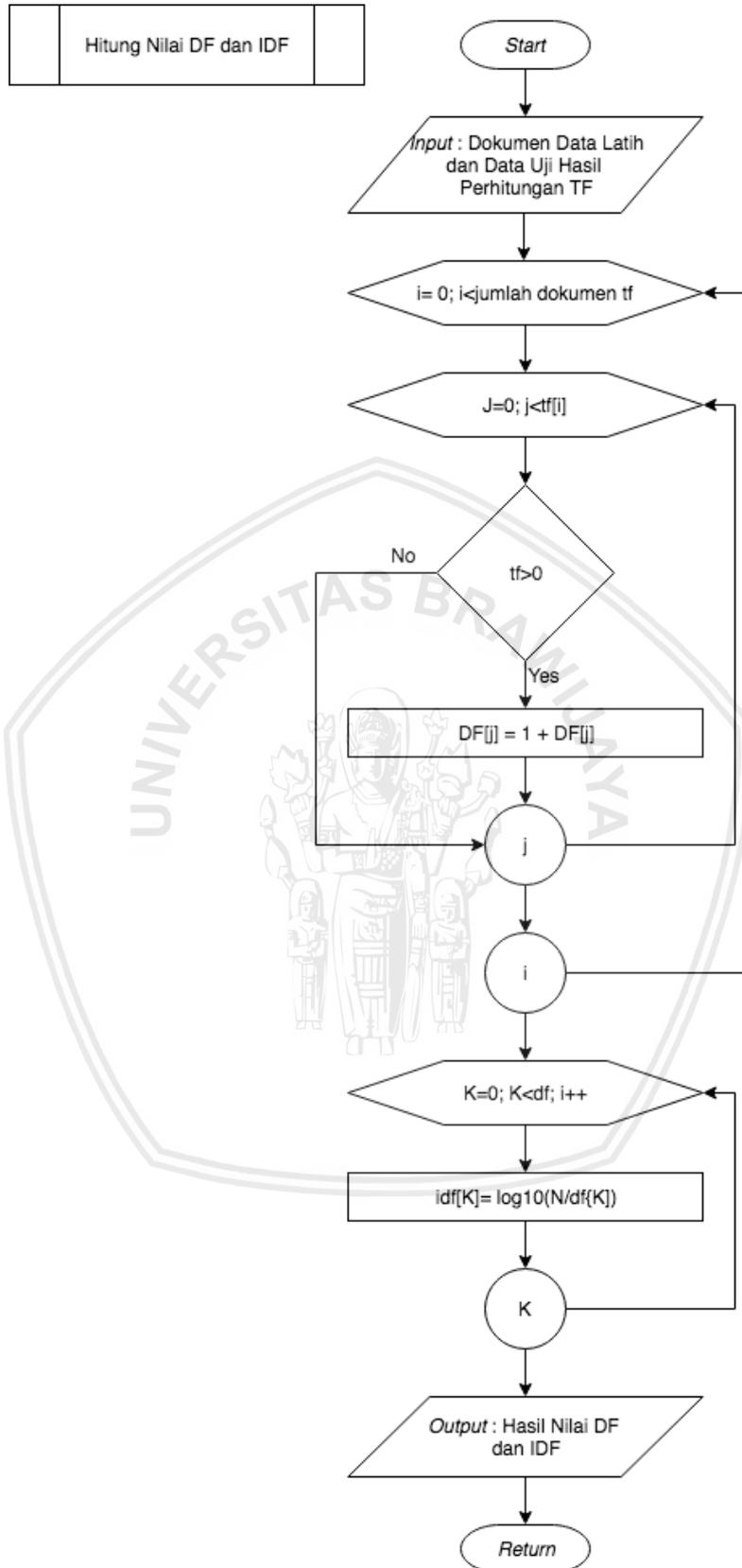
Setelah mendapatkan hasil dari perhitungan TF, akan dilakukan perhitungan bobot dari setiap *term* yang telah ditemukan frekuensinya. Alur menghitung nilai Wtf ditunjukkan dalam Gambar 4.12.



Gambar 4.12 Alur Hitung Nilai Wtf

4.4.4 Menghitung Nilai DF dan IDF

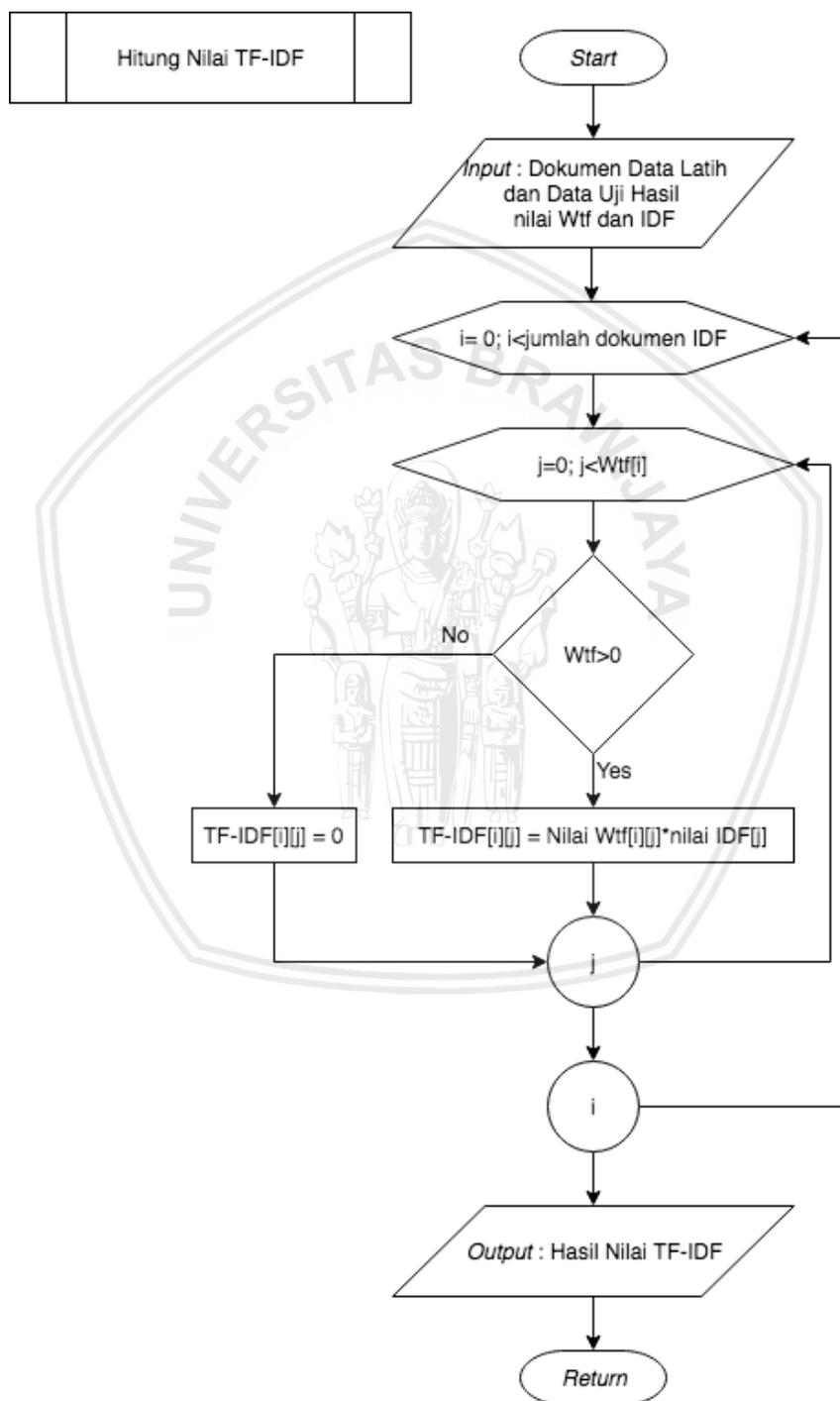
Sebelum mendapatkan nilai IDF (*Inverse Document Frequency*), perlu diketahui nilai DF (*Document Frequency*). Karena nilai IDF didapatkan dari kebalikan nilai DF. Nilai DF ditunjukkan dengan jumlah dokumen yang mengandung kata tersebut. Perhitungan ini ditunjukkan dalam Gambar 4.13.



Gambar 4.13 Alur Hitung Nilai DF dan IDF

4.4.5 Menghitung Nilai TF-IDF

Perhitungan nilai TF-IDF dilakukan untuk menghasilkan pembobotan tiap kata. Dengan mengalikan nilai Wtf dan nilai IDF maka akan dihasilkan nilai TF-IDF. Perkalian tersebut menunjukkan bahwa setiap kata yang tertinggi akan mewakili kata yang jarang ditemukan pada dokumen lainnya dan begitu juga sebaliknya. Alur perhitungan TF-IDF ditunjukkan dalam Gambar 4.14.

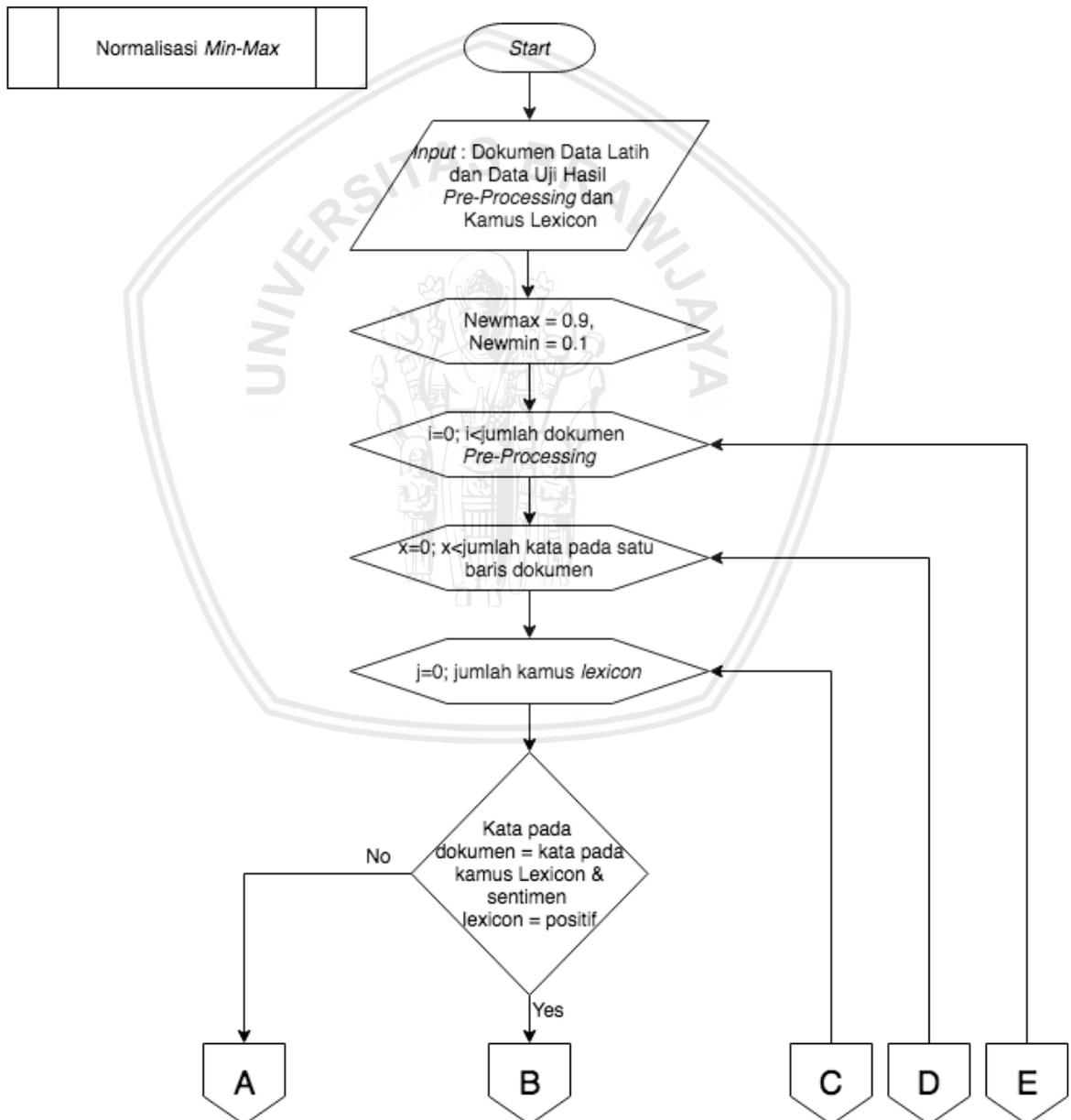


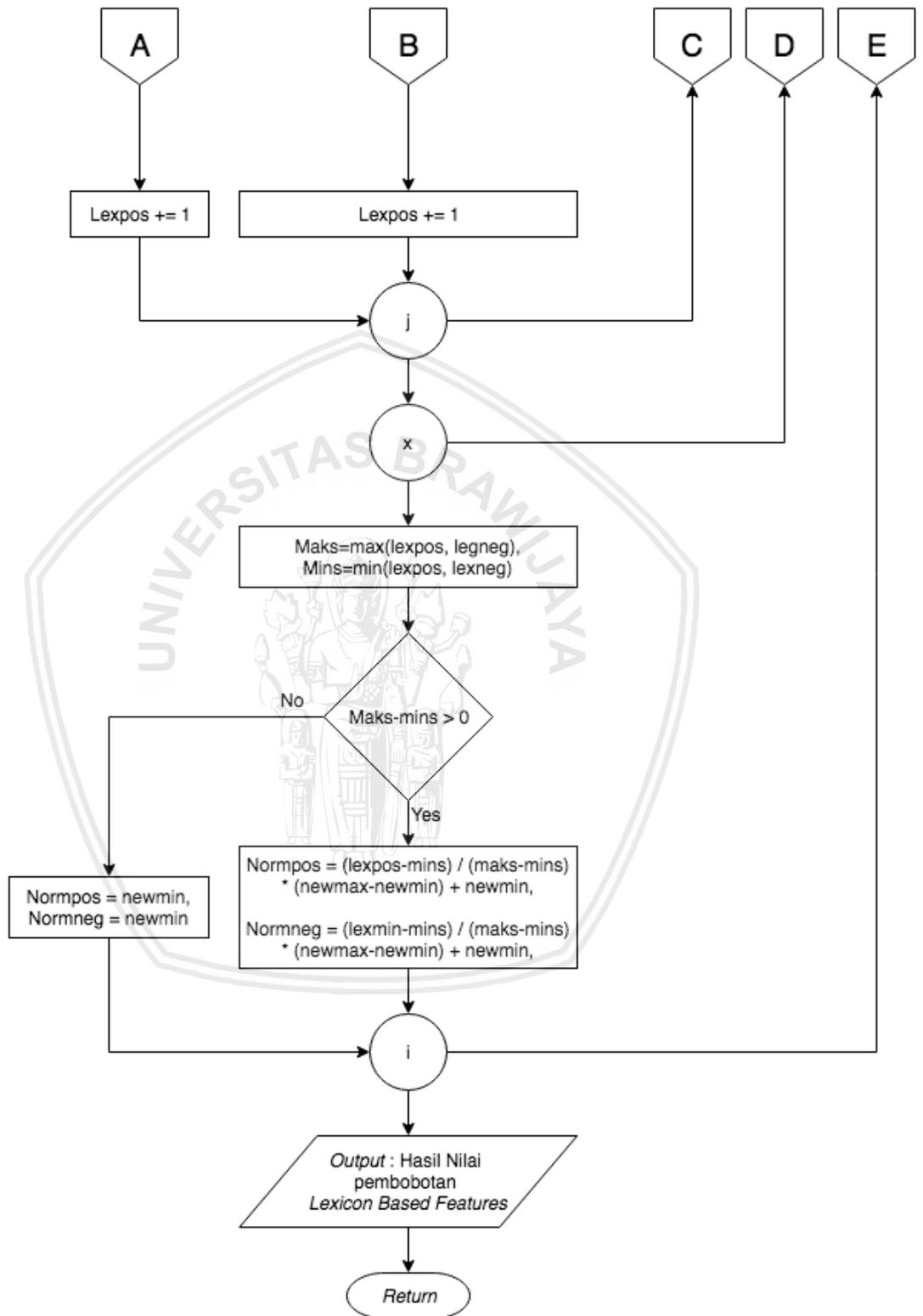
Gambar 4.14 Alur Hitung Nilai TF-IDF

4.5 Lexicon Based Features

Lexicon Based Features merupakan metode yang akan menambah bobot dari dokumen yang memiliki fitur-fitur bersentimen positif ataupun negatif. Sumber kamus didapatkan dari penelitian sebelumnya yang digunakan pada penelitian berjudul “Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine Dan Lexicon Based Features” oleh Umi Rofiqoh pada Tahun 2017. Alur dari proses metode *Lexicon Based Features* ditunjukkan dalam Gambar 4.15 dan Gambar 4.16.

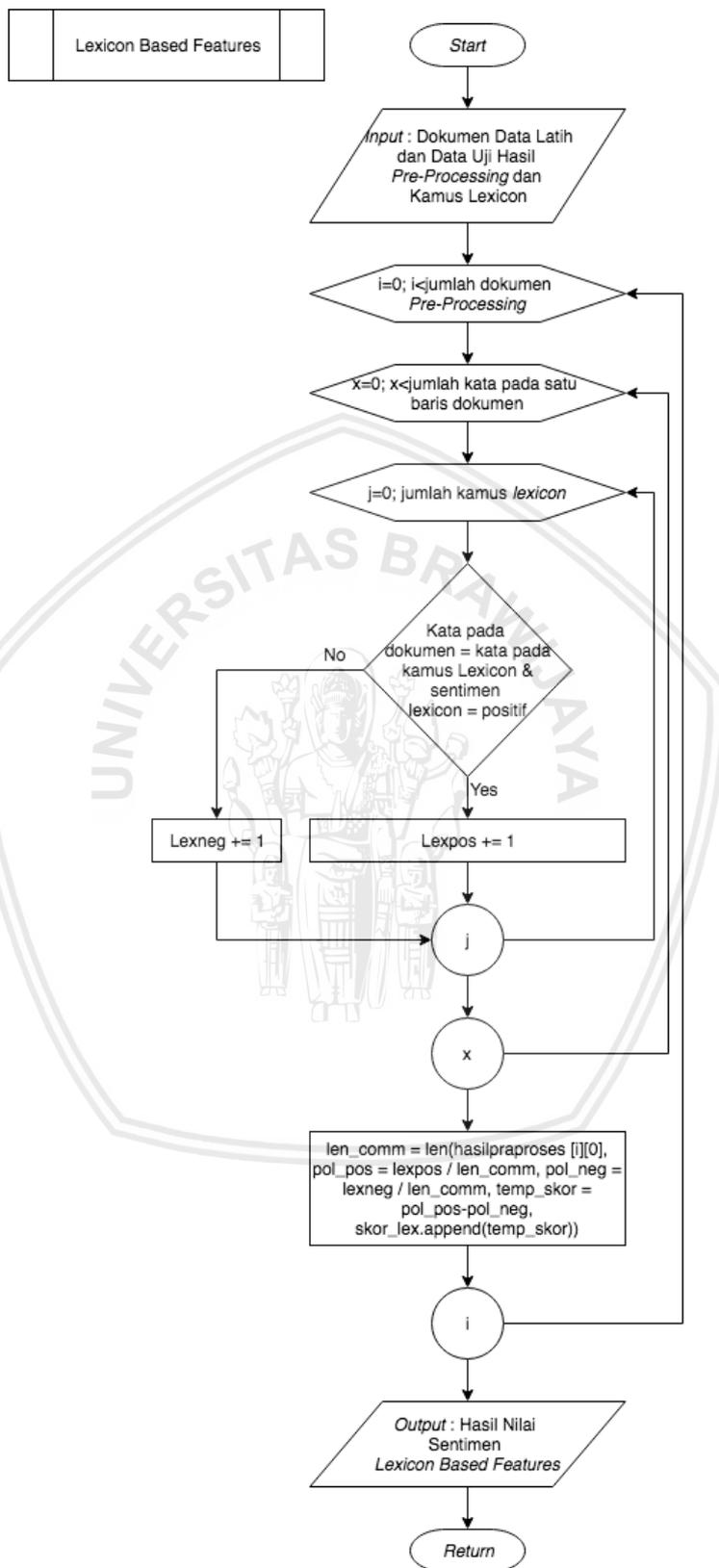
4.5.1 Normalisasi Min-Max





Gambar 4.15 Alur *Lexicon Based Features* dengan Normalisasi *Min-Max*

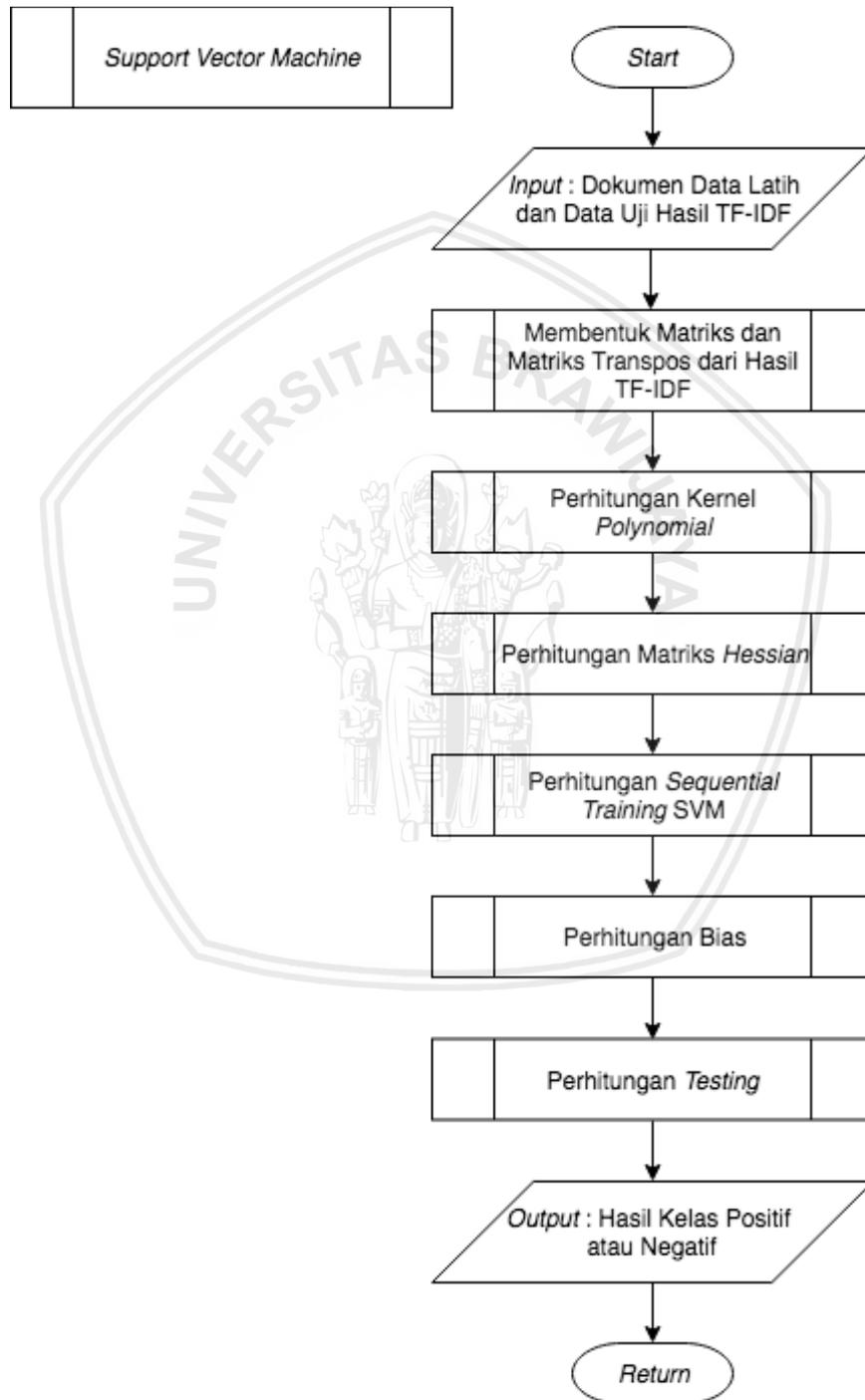
4.5.2 Nilai Sentimen



Gambar 4.16 Alur *Lexicon Based Features* dengan Nilai Sentimen

4.6 Support Vector Machine

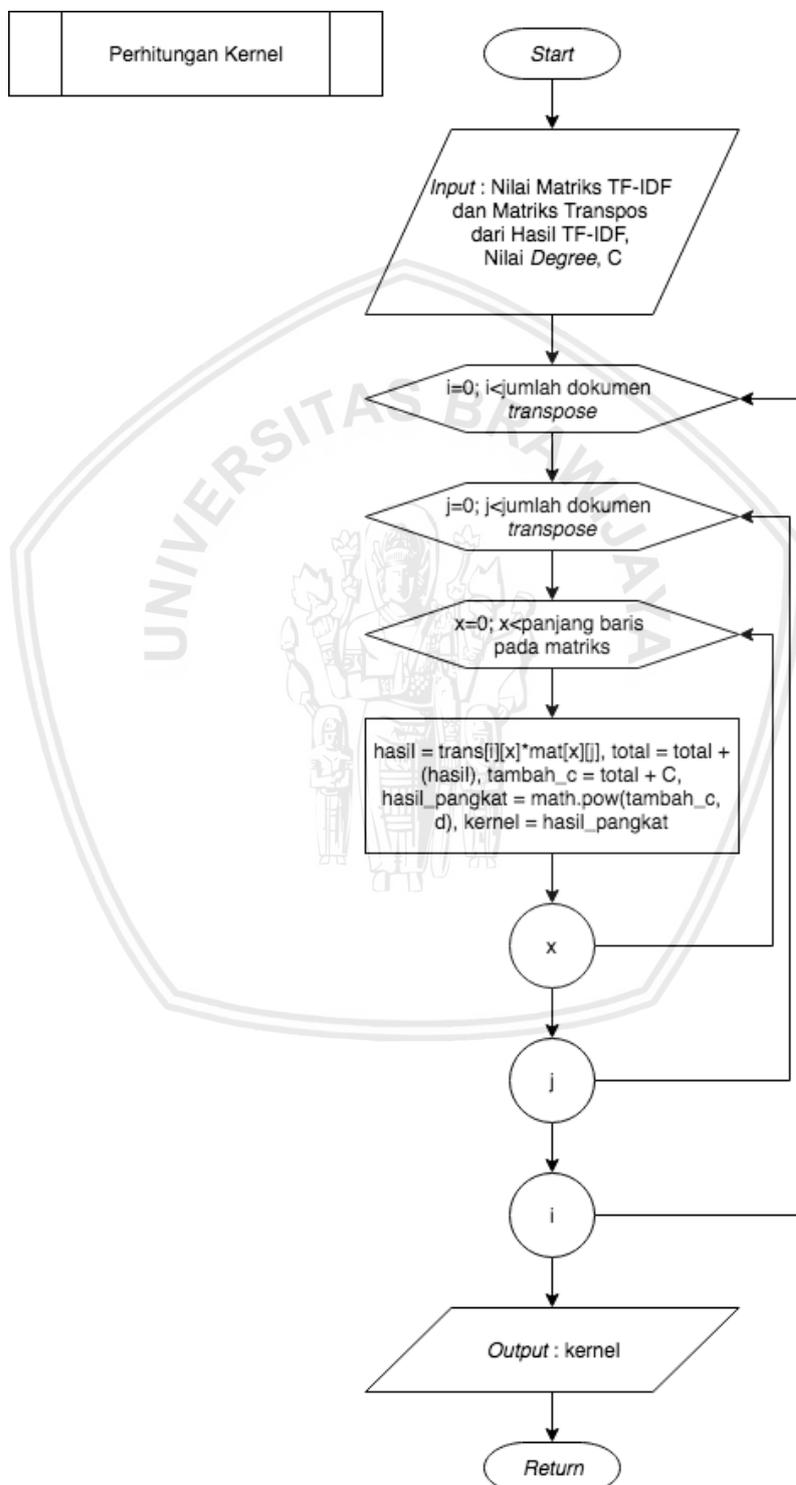
Setelah dilakukan tahap *pre-processing* dan pembobotan dengan TF-IDF, dilakukan perhitungan klasifikasi dengan SVM. *Support Vector Machine* merupakan metode atau algoritme yang digunakan untuk analisis sentimen pada penelitian ini. Hasil yang diperoleh adalah klasifikasi kelas positif dan kelas negatif. Alur perhitungan dengan SVM ditunjukkan dalam Gambar 4.17.



Gambar 4.17 Alur Perhitungan SVM

4.6.1 Perhitungan Kernel

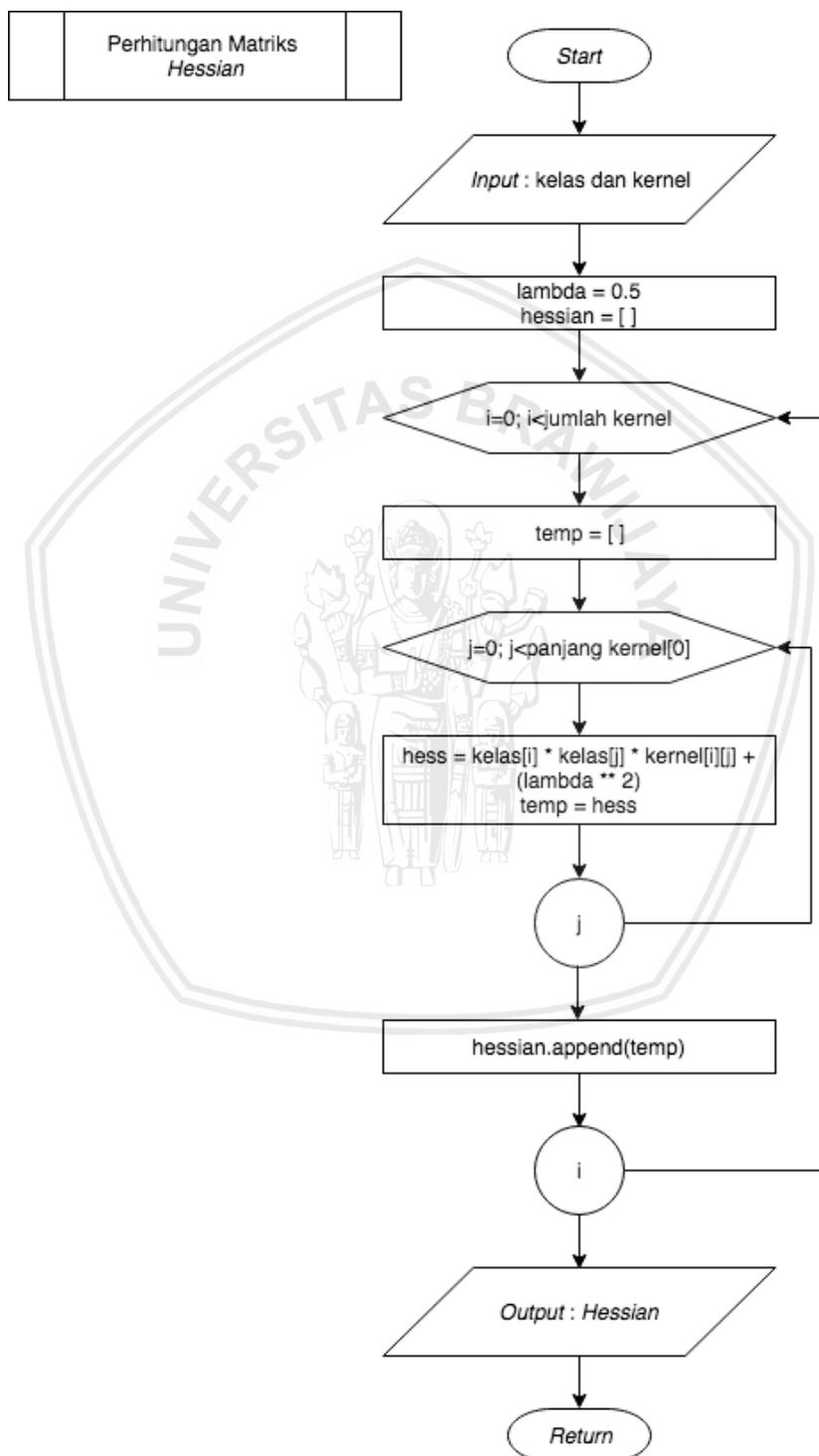
Perhitungan kernel merupakan tahap awal dari *Support Vector Machine*. Kernel yang digunakan yaitu kernel Polinomial. Data yang digunakan merupakan hasil dari perhitungan pembobotan TF-IDF. Alur perhitungan kernel dapat dilihat dalam Gambar 4.18.



Gambar 4.18 Alur Perhitungan Kernel

4.6.2 Perhitungan Matriks *Hessian*

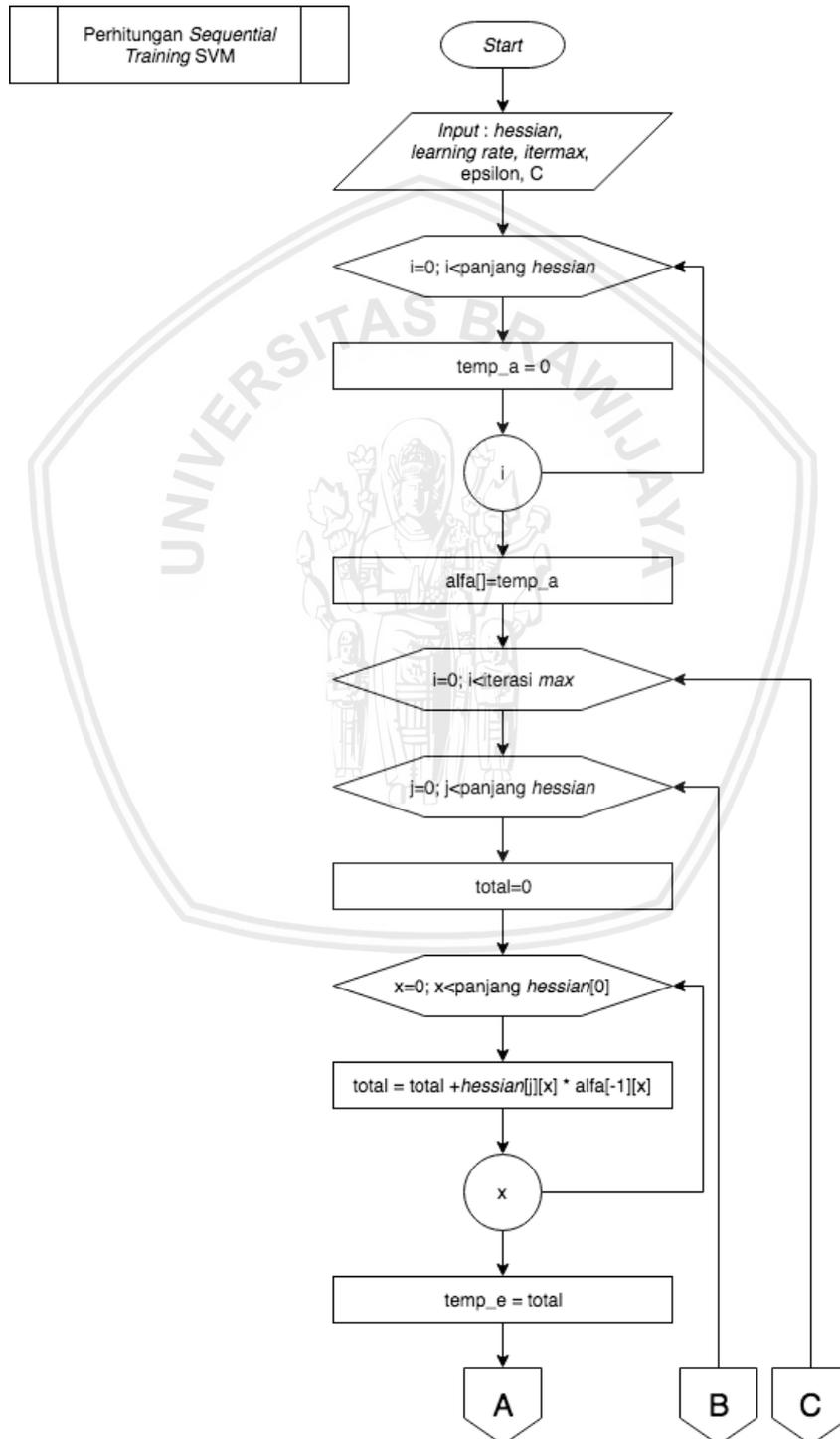
Setelah menghitung kernel, dilakukan perhitungan Matriks *Hessian* pada hasil perhitungan kernel data ke-x dan data ke-y dengan hasil matriks perkalian kelas positif (1) dan kelas negatif (-1) yang kemudian dikalikan dengan lambda. Alur perhitungan Matriks *Hessian* ditunjukkan dalam Gambar 4.19.

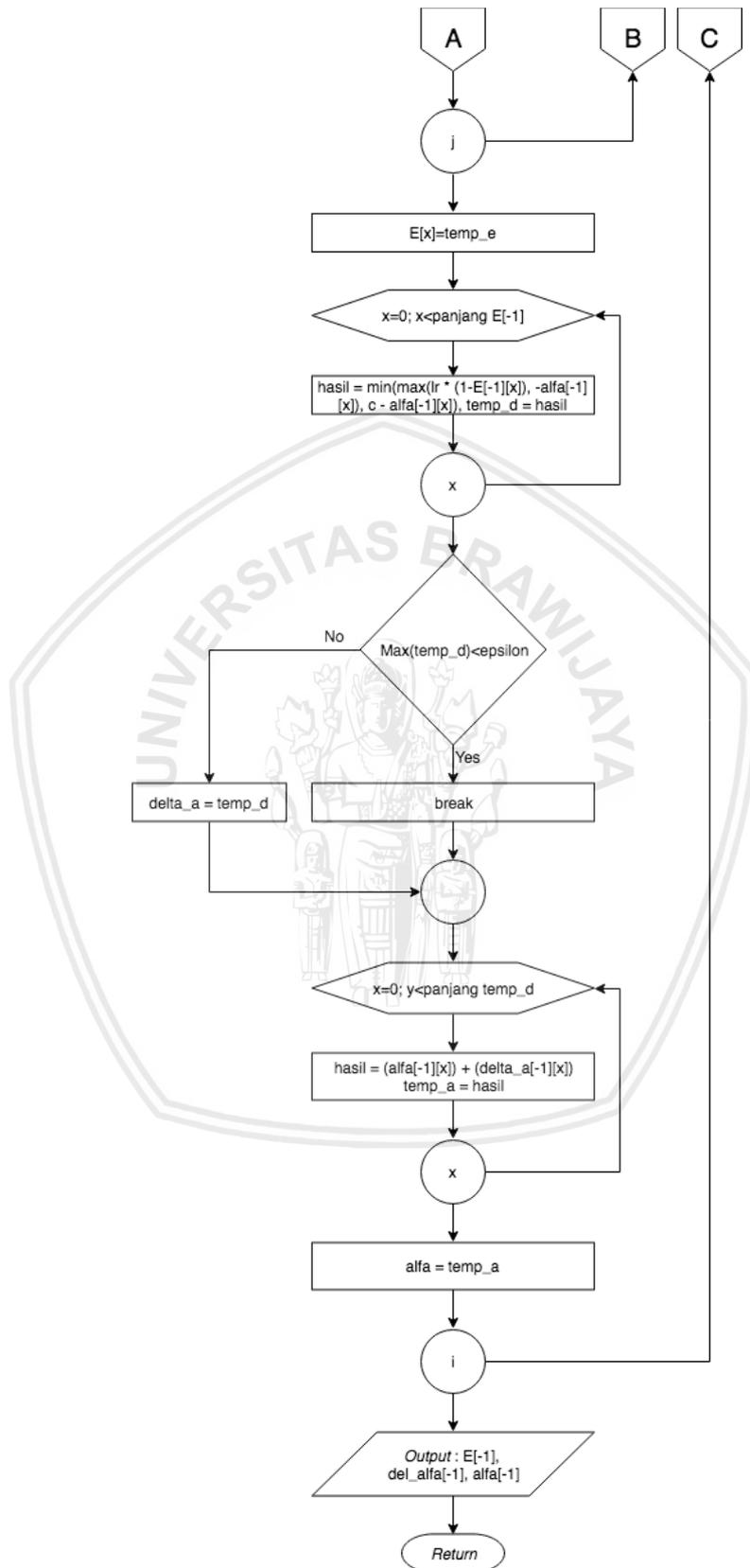


Gambar 4.19 Alur Perhitungan Matriks *Hessian*

4.6.3 Perhitungan Training SVM

Pada tahap ini dilakukan perhitungan nilai E, d, dan alfa. Proses perhitungan sequential training dilakukan sebanyak jumlah iterasi maksimum yang dimasukkan. Jika nilai lebih kecil dari nilai epsilon, maka iterasi akan berhenti sebelum mencapai iterasi maksimum. Nilai akhir dari merupakan nilai *support vector*, yaitu titik data terdekat dengan *hyperplane* antar kelas data. Alur perhitungan *sequential training* SVM ditunjukkan dalam Gambar 4.20.

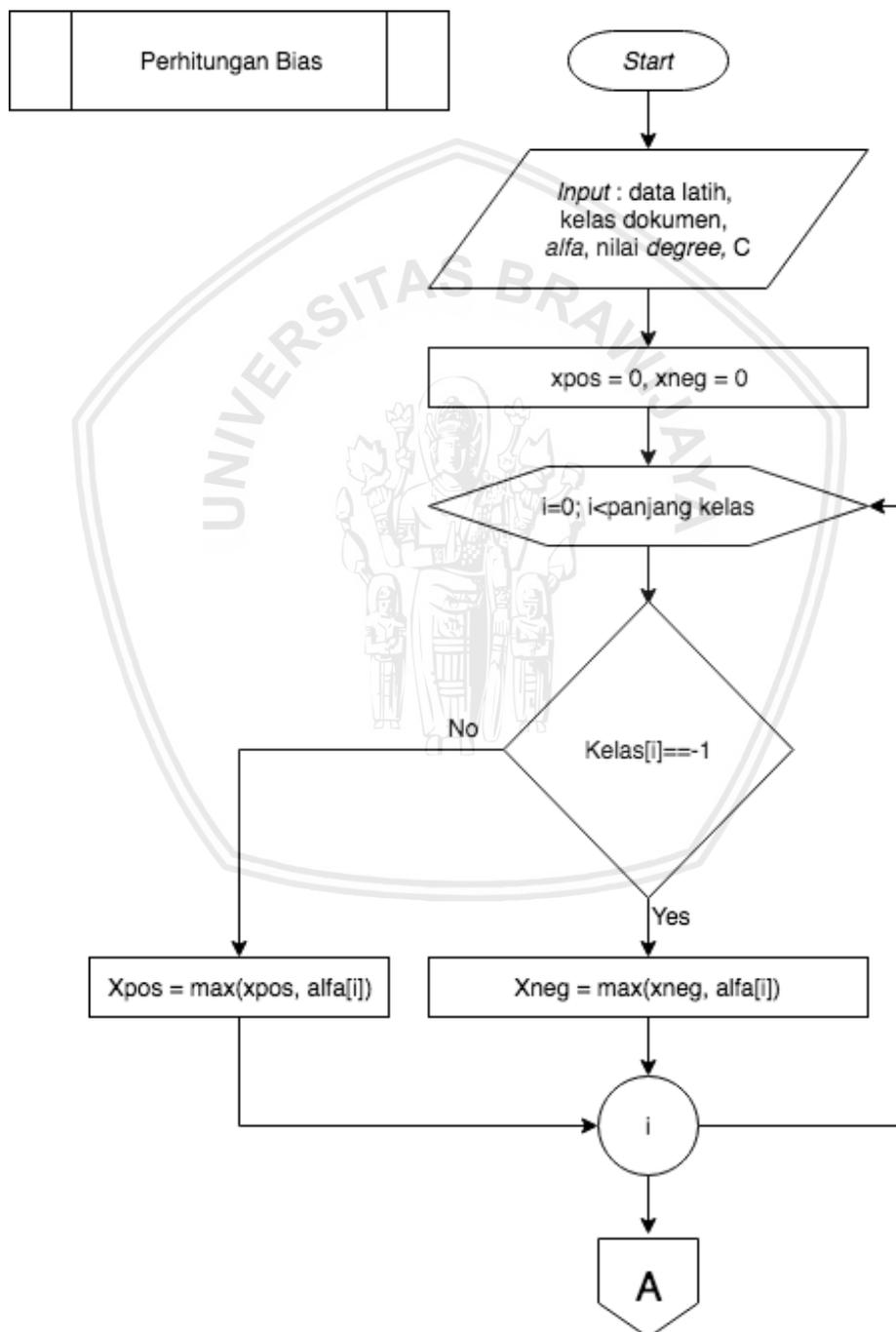


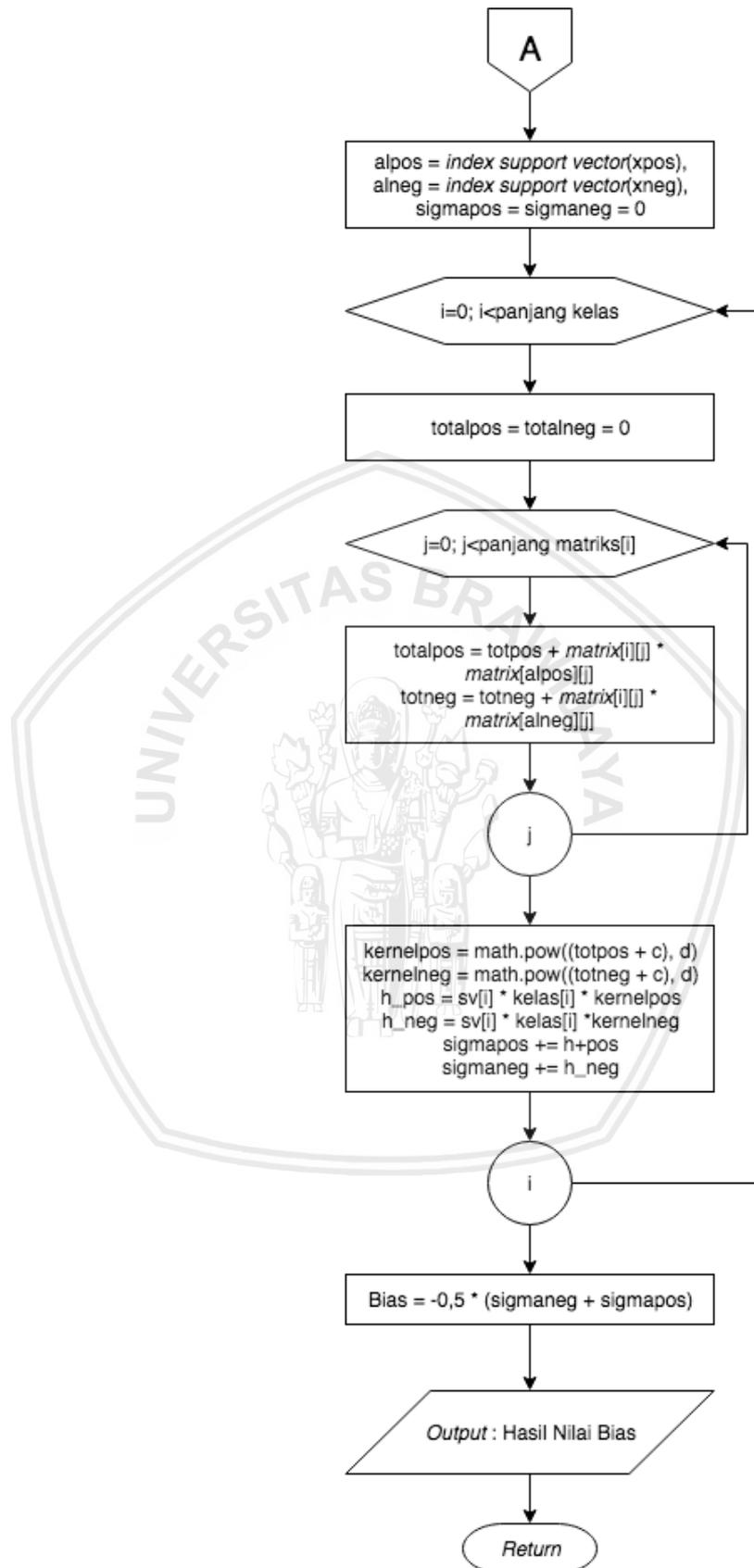


Gambar 4.20 Alur Perhitungan *Training SVM*

4.6.4 Perhitungan Bias

Menghitung nilai bias adalah tahap terakhir dari perhitungan klasifikasi menggunakan metode SVM, untuk pembentukan *hyperplane*. Pada tahap ini, akan dipilih nilai tertinggi pada tiap kelas positif dan kelas negatif dari *support vector*. Nilai tertinggi menunjukkan sebagai pembatas antar kelas data. Hasil *support vector* yang sebelumnya didapat akan dikalikan dengan hasil perhitungan kernel polinomial yang memiliki nilai tertinggi. Alur perhitungan bias ditunjukkan dalam Gambar 4.21.

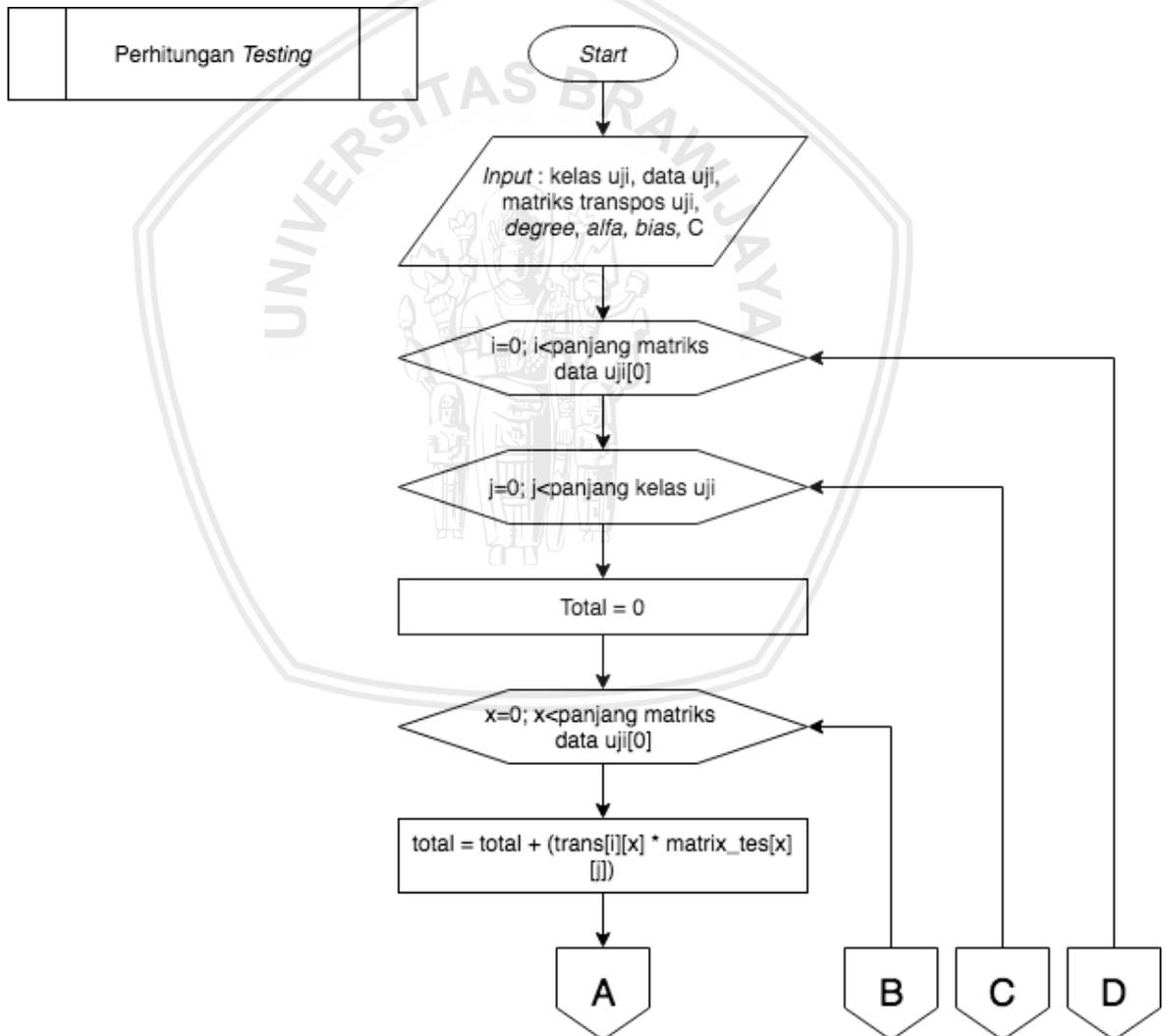


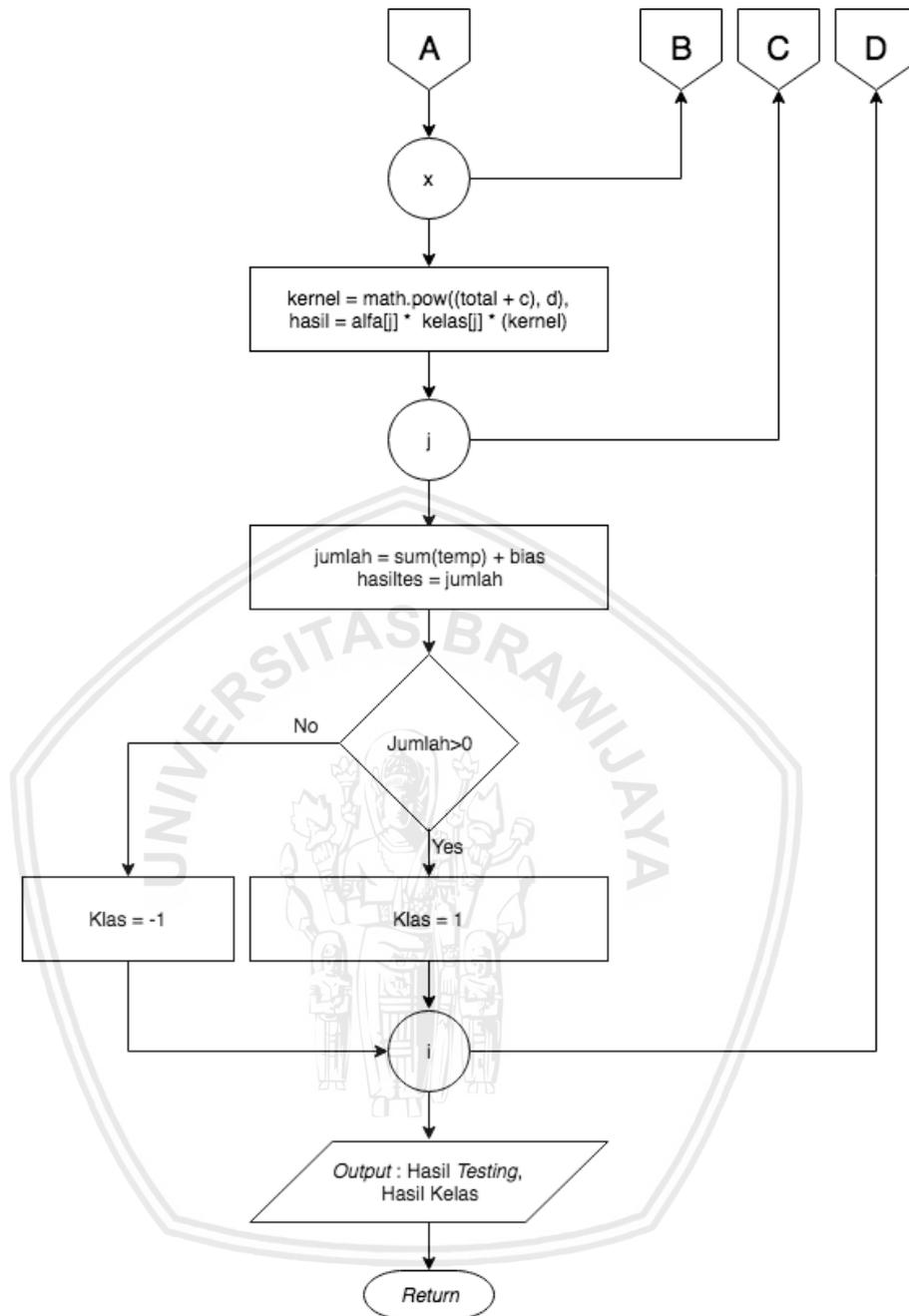


Gambar 4.21 Alur Perhitungan Bias

4.6.5 Perhitungan *Testing SVM*

Tahapan untuk mendapatkan hasil analisis sentimen adalah menghitung data uji yang ingin diketahui kelasnya. Kelas yang ditunjukkan adalah kelas positif atau negatif. 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 polinomial. Hasil perkalian tersebut akan dihitung kembali dengan nilai 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 dalam Gambar 4.22.





Gambar 4.22 Alur Perhitungan *Testing SVM*

4.7 Manualisasi dengan Algoritme SVM

Pada penelitian ini, dilakukan perhitungan menggunakan algoritme *Support Vector Machine* (SVM). Sebelum melakukan perhitungan SVM, terdapat beberapa tahap yang dapat dilakukan seperti yang akan dijelaskan selanjutnya. Diketahui kode dokumen d1, d2, d3, d4, d5, dan d6 sebagai masukkan data latih dan kode dokumen d7, d8, dan d9 sebagai masukkan data uji.

4.7.1 Manualisasi *Case Folding*

Proses manualisasi dengan algoritme SVM, diawali dengan tahap *pre-processing*. Langkah awal dalam tahap *pre-processing*, yaitu *case folding*. Tahap *case folding* ditunjukkan pada Tabel 4.1 dan 4.2. Pada Tabel 4.1 merupakan proses manualisasi untuk masukkan data latih dan pada Tabel 4.2 merupakan proses manualisasi untuk masukan data uji.

Tabel 4.1 Data Latih Proses *Case Folding*

Kode Dokumen	Data Asli	Data setelah <i>Case Folding</i>
d1	Dear @SriwijayaAir pesawat delay lebih dari 4,5 jam CGK-TKG (SJ 094) tapi tdk ada kompensasi sama sekali Mhn bantuannya Cc : @alvinlie21 @OmbudsmanRI137 @YLKI_ID	dear @sriwijayaair pesawat delay lebih dari 4,5 jam cgk-tdk (sj 094) tapi tdk ada kompensasi sama sekali mhn bantuannya cc : @alvinlie21 @ombudsmanri137 @ylki_id
d2	asli lah dibikin kesel, ada aja dah si @LionAirID mindahin jadwal seenak jidat https://t.co/mk4pLqy9Dq	asli lah dibikin kesel, ada aja dah si @lionairid mindahin jadwal seenak jidat https://t.co/mk4plqy9dq
d3	@IndonesiaGaruda makin hari makin delay	@indonesiagaruda makin hari makin delay
d4	Always a good and pleasant experience flying with @Citilink . Thanks QG-929 PKU-KNO 😊	always a good and pleasant experience flying with @citilink . thanks qg-929 pku-kno :)
d5	Pake @BatikAirINA dr jogja makin nyaman	pake @batikairina dr jogja makin nyaman
d6	Love @Citilink tetap nyaman dan smooth ...	love @citilink tetap nyaman dan smooth ...

Tabel 4.2 Data Uji Proses *Case Folding*

Kode Dokumen	Data Asli	Data setelah <i>Case Folding</i>
d7	@Arie_Kriting @BatikAirINA Barang saya tdk kembali pakai maskapai Batik air dan Malindo....group lion air adalah maskapai pencuri dan pemelihara sindikat pembobol bagasi.	@arie_kriting@batikairina barang saya tdk kembali pakai maskapai batik air dan malindo....group lion air adalah maskapai

Tabel 4.2 Data Uji Proses *Case Folding* (lanjutan)

d8	Worth the wait. Delay 90 menit dapet biskuat dan goodtime!! OMG!! THX BGT!! Love you @LionAirID OMG OMG OMG PLZ KILL ME	worth the wait. delay 90 menit dapet biskuat dan goodtime!! omg!! thx bgt!! love you @lionairid omg omg omg plz kill me
d9	@SriwijayaAir Wah parah neh Pramugarinya masa ada ibu2 bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai NO TEMPAT DUDUK yg tertera ditiket.SUMPAH KAPOK NAIK @SriwijayaAir @YLKI_ID @detikcom https://t.co/TFMXnnmLcl	@sriwijayaair wah parah neh pramugarinya masa ada ibu2 bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai no tempat duduk yg tertera ditiket.sumpah kapok naik @sriwijayaair @ykli_id @detikcom https://t.co/tfmxxnmlici

4.7.2 Manualisasi *Data Cleaning*

Langkah berikutnya dalam tahap *pre-processing*, yaitu *data cleaning*. Tahap *data cleaning* ditunjukkan pada Tabel 4.3 dan 4.4. Pada Tabel 4.3 merupakan proses manualisasi untuk masukkan data latih dan pada Tabel 4.4 merupakan proses manualisasi untuk masukan data uji.

Tabel 4.3 Data Latih Proses *Data Cleaning*

Kode Dokumen	Data Asli	Data setelah <i>Cleaning</i>
d1	dear @sriwijayaair pesawat delay lebih dari 4,5 jam cgk-tkg (sj 094) tapi tdk ada kompensasi sama sekali mhn bantuannya cc : @alvinlie21 @ombudsmanri137 @ylki_id	dear pesawat delay lebih dari jam cgk tkg sj tapi tdk ada kompensasi sama sekali mhn bantuannya cc
d2	asli lah dibikin kesel, ada aja dah si @lionairid mindahin jadwal seenak jidat https://t.co/mk4plqy9dq	asli lah dibikin kesel ada aja dah si mindahin jadwal seenak jidat



Tabel 4.3 Data Latih Proses *Data Cleaning* (lanjutan)

d3	@indonesiagaruda makin hari makin delay	makin hari makin delay
d4	always a good and pleasant experience flying with @citilink . thanks qg-929 pku-kno :)	always a good and pleasant experience flying with thanks qg pku kno
d5	pake @batikairina dr jogja makin nyaman	pake dr jogja makin nyaman
d6	love @citilink tetap nyaman dan smooth ...	love tetap nyaman dan smooth

Tabel 4.4 Data Uji Proses *Data Cleaning*

Kode Dokumen	Data Asli	Data setelah <i>Cleaning</i>
d7	@arie_kriting @batikairina barang saya tdk kembali pakai maskapai batik air dan malindo....group lion air adalah maskapai pencuri dan pemelihara sindikat pembobol bagasi. maskapai ini tdk ada tanggung jawabnya https://t.co/zfvs1koya	barang saya tdk kembali pakai maskapai batikair dan malindo group lionair adalah maskapai pencuri dan pemelihara sindikat pembobol bagasi maskapai ini tdk ada tanggung jawabnya
d8	worth the wait. delay 90 menit dapet biskuat dan goodtime!! omg!! thx bgt!! love you @lionairid omg omg omg plz kill me	worth the wait delay menit dapet biskuat dan goodtime omg thx bgt love you omg omg omg plz kill me
d9	@sriwijayaair wah parah neh pramugarinya masa ada ibu2 bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai no tempat duduk yg tertera ditiket.sumpah kapok naik @sriwijayaair @ykli_id @detikcom https://t.co/tfmxnmlci	wah parah neh pramugarinya masa ada ibu bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai no tempat

4.7.3 Manualisasi *Translation*

Langkah berikutnya dalam tahap *pre-processing*, yaitu *translation*. Tahap *translation* ditunjukkan pada Tabel 4.5 dan 4.6. Pada Tabel 4.5 merupakan proses manualisasi untuk masukkan data latih dan pada Tabel 4.6 merupakan proses manualisasi untuk masukan data uji.

Tabel 4.5 Data Latih Proses *Translation*

Kode Dokumen	Data Asli	Data setelah <i>Translation</i>
d1	dear pesawat delay lebih dari jam ckg tkg sj tapi tdk ada kompensasi sama sekali mhn bantuannya cc	terhormat pesawat menunda lebih dari jam ckg tkg sj tapi tdk ada kompensasi sama sekali mhn bantuannya cc
d2	asli lah dibikin kesel ada aja dah si mindahin jadwal seenak jidat	asli lah dibikin kesel ada aja dah si mindahin jadwal seenak jidat
d3	makin hari makin delay	makin hari makin menunda
d4	always a good and pleasant experience flying with thanks qg pku kno	selalu sebuah baik dan nyaman pengalaman terbang dengan terima kasih qg pku kno
d5	pake dr jogja makin nyaman	pake dr jogja makin nyaman
d6	love tetap nyaman dan smooth	cinta tetap nyaman dan halus

Tabel 4.6 Data Uji Proses *Translation*

Kode Dokumen	Data Asli	Data setelah <i>Translation</i>
d7	@arie_kriting @batikairina barang saya tdk kembali pakai maskapai batik air dan malindo....group lion air adalah maskapai pencuri dan pemelihara sindikat pembobol bagasi. maskapai ini tdk ada tanggung jawabnya https://t.co/zfvs1koya	barang saya tdk kembali pakai maskapai batikair dan malindo grup lionair adalah maskapai pencuri dan pemelihara sindikat pembobol

Tabel 4.6 Data Uji Proses *Translation* (lanjutan)

d8	worth the wait. delay 90 menit dapet biskuat dan goodtime!! omg!! thx bgt!! love you @lionairid omg omg omg plz kill me	bernilai itu tunggu menunda menit dapet biskuat dan goodtime omg thx bgt cinta kamu omg omg omg plz bunuh aku
d9	@sriwijayaair wah parah neh pramugarinya masa ada ibu2 bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai no tempat duduk yg tertera ditiket.sumpah kapok naik @sriwijayaair @ykli_id @detikcom https://t.co/tfmxnmlci	wah parah neh pramugarinya masa ada ibu bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai no tempat duduk yg tertera ditiket sumpah kapok naik

4.7.4 Manualisasi *Stopword Removal*

Setelah itu dalam tahap *pre-processing*, yaitu *stopward removal*. Tahap *stopward removal* ditunjukkan pada Tabel 4.7 dan 4.8. Pada Tabel 4.7 merupakan proses manualisasi untuk masukkan data latih dan pada Tabel 4.8 merupakan proses manualisasi untuk masukan data uji.

Tabel 4.7 Data Latih Proses *Stopword Removal*

Kode Dokumen	Data Asli	Data setelah <i>Stopword Removal</i>
d1	terhormat pesawat menunda lebih dari jam cengkareng lampung sriwijaya tapi tidak ada kompensasi sama sekali mohon bantuannya cc	terhormat pesawat menunda jam cengkareng lampung sriwijaya kompensasi mohon bantuannya cc
d2	asli lah dibuat kesal ada saja deh si mindahin jadwal seenak jidat	asli dibuat kesal deh si mindahin jadwal seenak jidat
d3	makin hari makin menunda	hari menunda



Tabel 4.7 Data Latih Proses *Stopword Removal* (lanjutan)

d4	selalu sebuah baik dan nyaman pengalaman terbang dengan terima kasih qg pekanbaru medan	baik nyaman pengalaman terbang terima kasih qg pekanbaru medan
d5	pakai dari jogja makin nyaman	pakai jogja nyaman
d6	cinta tetap nyaman dan halus	cinta tetap nyaman halus

Tabel 4.8 Data Uji Proses *Stopword Removal*

Kode Dokumen	Data Asli	Data setelah <i>Stopword Removal</i>
d7	barang saya tidak kembali pakai maskapai batikair dan malindo grup lionair adalah maskapai pencuri dan pemelihara sindikat pembobol bagasi maskapai ini tidak ada tanggung jawabnya	barang kembali pakai maskapai batikair malindo grup lionair maskapai pencuri pemelihara sindikat pembobol bagasi maskapai tanggung jawabnya
d8	bernilai tunggu menunda menit biskuat goodtime omg terima kasih cinta omg omg omg tolong bunuh	bernilai tunggu menunda menit biskuat goodtime omg terima kasih cinta omg omg omg tolong bunuh
d9	parah nih pramugarinya masa ibu salah tempat duduk bukannya awal diarahkan agar salah duduk itu kan sudah tugasnya pramugari memastikan penumpang duduk sesuai nomor tempat duduk yang tertera ditiket sumpah kapok naik	parah nih pramugarinya masa ibu salah tempat duduk awal diarahkan salah duduk tugasnya pramugari memastikan penumpang duduk sesuai nomor tempat duduk tertera ditiket sumpah kapok naik

4.7.5 Manualisasi *Stemming*

Langkah berikutnya dalam tahap *pre-processing*, yaitu proses *stemming*. Tahap *stemming* ditunjukkan pada Tabel 4.9 dan 4.10. Pada Tabel 4.9 merupakan proses



manualisasi untuk masukkan data latih dan pada Tabel 4.10 merupakan proses manualisasi untuk masukan data uji.

Tabel 4.9 Data Latih Proses *Stemming*

Kode Dokumen	Data Asli	Data setelah <i>Stemming</i>
d1	terhormat pesawat menunda jam cengkareng lampung sriwijaya kompensasi mohon bantuannya cc	hormat pesawat tunda jam cengkareng lampung sriwijaya kompensasi mohon bantu cc
d2	asli dibuat kesal deh si mindahin jadwal seenak jidat	asli buat kesal deh si pindah jadwal enak jidat
d3	hari menunda	hari tunda
d4	baik nyaman pengalaman terbang terima kasih qg pekanbaru medan	baik nyaman pengalaman terbang terima kasih qg pekanbaru medan
d5	pakai jogja nyaman	pakai jogja nyaman
d6	cinta tetap nyaman halus	cinta tetap nyaman halus

Tabel 4.10 Data Uji Proses *Stemming*

Kode Dokumen	Data Asli	Data setelah <i>Stemming</i>
d7	barang kembali pakai maskapai batikair malindo grup lionair maskapai pencuri pemelihara sindikat pembobol bagasi maskapai tanggung jawabnya	barang kembali pakai maskapai batikair malindo grup lionair maskapai curi pelihara sindikat bobol bagasi maskapai tanggung jawab
d8	bernilai tunggu menunda menit biskuat goodtime omg terima kasih cinta omg omg omg tolong bunuh	nilai tunggu tunda menit biskuat goodtime omg terima kasih cinta omg omg omg tolong bunuh

Tabel 4.10 Data Uji Proses *Stemming* (lanjutan)

d9	parah nih pramugarinya masa ibu salah tempat duduk awal diarahkan salah duduk tugasnya pramugari memastikan penumpang duduk sesuai nomor tempat duduk tertera ditiket sumpah kapok naik	parah nih pramugari masa ibu salah tempat duduk awal arah salah duduk tugas pramugari pasti penumpang duduk sesuai nomor tempat duduk tera tiket sumpah kapok naik
----	---	--

4.7.6 Manualisasi Tokenisasi

Selanjutnya dalam tahap *pre-processing*, yaitu proses tokenisasi. Tahap tokenisasi ditunjukkan pada Tabel 4.11 dan 4.12. Pada Tabel 4.11 merupakan proses manualisasi untuk masukkan data latih dan pada Tabel 4.12 merupakan proses manualisasi untuk masukan data uji.

Tabel 4.11 Data Latih Proses Tokenisasi

Kode Dokumen	Data Asli	Data setelah Tokenisasi
d1	hormat pesawat tunda jam cengkareng lampung sriwijaya kompensasi mohon bantu cc	hormat, pesawat, tunda, jam, cengkareng, lampung, sriwijaya, kompensasi, mohon, bantu, cc
d2	asli buat kesal deh si pindah jadwal enak jidat	asli, buat, kesal, deh, si, pindah, jadwal, enak, jidat
d3	hari tunda	hari, tunda
d4	baik nyaman pengalaman terbang terima kasih qg pekanbaru medan	baik, nyaman, pengalaman, terbang, terima, kasih, qg, pekanbaru, medan
d5	pakai jogja nyaman	pakai, jogja, nyaman
d6	cinta tetap nyaman halus	cinta, tetap, nyaman, halus



Tabel 4.12 Data Uji Proses Tokenisasi

Kode Dokumen	Data Asli	Data setelah Tokenisasi
d7	barang kembali pakai maskapai batikair malindo grup lionair maskapai curi pelihara sindikat bobol bagasi maskapai tanggung jawab	barang, kembali, pakai, maskapai, batikair, malindo, grup, lionair, maskapai, curi, pelihara, sindikat, bobol, bagasi, maskapai, tanggung, jawab
d8	nilai tunggu tunda menit biskuat goodtime omg terima kasih cinta omg omg omg tolong bunuh	nilai, tunggu, tunda, menit, biskuat, goodtime, omg, thx, cinta, omg, omg, omg, plz, bunuh
d9	parah nih pramugari masa ibu salah tempat duduk awal arah salah duduk tugas pramugari pasti penumpang duduk sesuai nomor tempat duduk tera tiket sumpah kapok naik	parah, nih, pramugari, masa, ibu, salah, tempat, duduk, awal, arah, salah, duduk, tugas, pramugari, pasti, penumpang, duduk, sesuai, nomor, tempat, duduk, tera, tiket, sumpah, kapok, naik

4.7.7 Manualisasi Perhitungan Tf, Wtf, DF, dan IDF

Setelah melalui tahap *pre-processing*, dilakukan proses perhitungan bobot tiap kata dengan metode TF-IDF. Langkah awal dalam pembobotan adalah perhitungan nilai TF, DF, Wtf, dan IDF yang ditunjukkan pada Tabel 4.13 dan 4.14.

Tabel 4.13 Perhitungan Nilai TF dan DF pada Data Latih dan Data Uji

Term	TF									df
	Data Latih						Data Uji			
	d1	d2	d3	d4	d5	d6	d7	d8	d9	
Hormat	1	0	0	0	0	0	0	0	0	1
Pesawat	1	0	0	0	0	0	0	0	0	1
Jam	1	0	0	0	0	0	0	0	0	1
Cengkareng	1	0	0	0	0	0	0	0	0	1
Lampung	1	0	0	0	0	0	0	0	0	1



Tabel 4.13 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	d7	d8	d9	
Sriwijaya	1	0	0	0	0	0	0	0	0	1
Kompensasi	1	0	0	0	0	0	0	0	0	1
Mohon	1	0	0	0	0	0	0	0	0	1
Bantu	1	0	0	0	0	0	0	0	0	1
Cc	1	0	0	0	0	0	0	0	0	1
Asli	0	1	0	0	0	0	0	0	0	1
Buat	0	1	0	0	0	0	0	0	0	1
Kesal	0	1	0	0	0	0	0	0	0	1
Deh	0	1	0	0	0	0	0	0	0	1
Si	0	1	0	0	0	0	0	0	0	1
Pindah	0	1	0	0	0	0	0	0	0	1
Jadwal	0	1	0	0	0	0	0	0	0	1
Enak	0	1	0	0	0	0	0	0	0	1
Jidat	0	1	0	0	0	0	0	0	0	1
Hari	0	0	1	0	0	0	0	0	0	1
Tunda	1	0	1	0	0	1	0	0	0	3
Baik	0	0	0	1	0	0	0	0	0	1
Nyaman	0	0	0	1	1	1	0	0	0	3
Pengalaman	0	0	0	1	0	0	0	0	0	1
Terbang	0	0	0	1	0	0	0	0	0	1
Terima	0	0	0	1	0	1	0	0	0	2
Kasih	0	0	0	1	0	1	0	0	0	2
Qg	0	0	0	1	0	0	0	0	0	1
Pekanbaru	0	0	0	1	0	0	0	0	0	1
Medan	0	0	0	1	0	0	0	0	0	1
Pakai	0	0	0	0	1	0	0	0	0	1
Jogja	0	0	0	0	1	0	0	0	0	1
Cinta	0	0	0	0	0	1	0	0	0	1
Tetap	0	0	0	0	0	1	0	0	0	1
Halus	0	0	0	0	0	1	0	0	0	1
Barang	0	0	0	0	0	0	1	0	0	1

Tabel 4.14 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	d7	d8	d9	
Hormat	1	0	0	0	0	0	0	0	0	0.954

Tabel 4.14 Perhitungan Nilai $W_{tf,d}$ dan idf pada Data Latih dan Data Uji (lanjutan)

Term	$W_{tf,d}$									idf
	Data Latih						Data Uji			
	d1	d2	d3	d4	d5	d6	d7	d8	d9	
Pesawat	1	0	0	0	0	0	0	0	0	0.954
Jam	1	0	0	0	0	0	0	0	0	0.954
Cengkareng	1	0	0	0	0	0	0	0	0	0.954
Lampung	1	0	0	0	0	0	0	0	0	0.954
Sriwijaya	1	0	0	0	0	0	0	0	0	0.954
Kompensasi	1	0	0	0	0	0	0	0	0	0.954
Mohon	1	0	0	0	0	0	0	0	0	0.954
Bantu	1	0	0	0	0	0	0	0	0	0.954
Cc	1	0	0	0	0	0	0	0	0	0.954
Asli	0	1	0	0	0	0	0	0	0	0.954
Buat	0	1	0	0	0	0	0	0	0	0.954
Kesal	0	1	0	0	0	0	0	0	0	0.954
Deh	0	1	0	0	0	0	0	0	0	0.954
Si	0	1	0	0	0	0	0	0	0	0.954
Pindah	0	1	0	0	0	0	0	0	0	0.954
Jadwal	0	1	0	0	0	0	0	0	0	0.954
Enak	0	1	0	0	0	0	0	0	0	0.954
Jidat	0	1	0	0	0	0	0	0	0	0.954
Hari	0	0	1	0	0	0	0	0	0	0.954
Tunda	1	0	1	0	0	1	0	0	0	0.477
Baik	0	0	0	1	0	0	0	0	0	0.954
Nyaman	0	0	0	1	1	1	0	0	0	0.477
Pengalaman	0	0	0	1	0	0	0	0	0	0.954
Terbang	0	0	0	1	0	0	0	0	0	0.954
Terima	0	0	0	1	0	1	0	0	0	0.653
Kasih	0	0	0	1	0	1	0	0	0	0.653
Qg	0	0	0	1	0	0	0	0	0	0.954
Pekanbaru	0	0	0	1	0	0	0	0	0	0.954
Medan	0	0	0	1	0	0	0	0	0	0.954
Pakai	0	0	0	0	1	0	0	0	0	0.954
Jogja	0	0	0	0	1	0	0	0	0	0.954
Cinta	0	0	0	0	0	1	0	0	0	0.954
Tetap	0	0	0	0	0	1	0	0	0	0.954

4.7.8 Manualisasi Perhitungan TF-IDF

Setelah mendapatkan nilai TF, DF, W_{tf} , dan IDF, dilakukan perhitungan TF-IDF yang mengalikan nilai W_{tf} dan IDF. Manualisasi perhitungan TF-IDF ditunjukkan pada Tabel 4.15.

Tabel 4.15 Perhitungan TF-IDF pada Data Latih dan Data Uji

TF-IDF									
Term	Data Latih						Data Uji		
	d1	d2	d3	d4	d5	d6	d7	d8	d9
Hormat	0.954	0	0	0	0	0	0	0	0
Pesawat	0.954	0	0	0	0	0	0	0	0
Jam	0.954	0	0	0	0	0	0	0	0
Cengkareng	0.954	0	0	0	0	0	0	0	0
Lampung	0.954	0	0	0	0	0	0	0	0
Sriwijaya	0.954	0	0	0	0	0	0	0	0
Kompensasi	0.954	0	0	0	0	0	0	0	0
Mohon	0.954	0	0	0	0	0	0	0	0
Bantu	0.954	0	0	0	0	0	0	0	0
Cc	0.954	0	0	0	0	0	0	0	0
Asli	0	0.954	0	0	0	0	0	0	0
Buat	0	0.954	0	0	0	0	0	0	0
Kesal	0	0.954	0	0	0	0	0	0	0
Deh	0	0.954	0	0	0	0	0	0	0
Si	0	0.954	0	0	0	0	0	0	0
pindah	0	0.954	0	0	0	0	0	0	0
jadwal	0	0.954	0	0	0	0	0	0	0
Enak	0	0.954	0	0	0	0	0	0	0
Jidat	0	0.954	0	0	0	0	0	0	0
Hari	0	0	0.954	0	0	0	0	0	0

4.7.9 Manualisasi Perhitungan Normalisasi *Min-Max*

Selanjutnya, dilakukan perhitungan normalisasi dengan *Min-Max*. Manualisasi normalisasi dengan *min-max* ditunjukkan pada Tabel 4.16.

Tabel 4.16 Normalisasi Pembobotan *Lexicon*

Kode Dokumen	Jumlah Kata Sentimen	
	Positif	Negatif
d1	2	1
d2	2	2
d3	0	1
d4	3	0
d5	1	0
d6	3	0
d7	1	0
d8	4	2
d9	1	4

Tabel 4.17 Normalisasi Min-Max

Kode Dokumen	Jumlah Kata Sentimen		Normalisasi	
	Positif	Negatif	Positif	Negatif
d1	2	1	0.9	0.1
d2	2	2	0.9	0.1
d3	0	1	0.1	0.9
d4	3	0	0.9	0.1
d5	1	0	0.9	0.1
d6	3	0	0.9	0.1
d7	1	0	0.9	0.1
d8	4	2	0.9	0.1
d9	1	4	0.1	0.9

4.7.10 Manualisasi Perhitungan Klasifikasi SVM

Setelah melakukan tahap *pre-processing*, pembobotan TF-IDF, dan normalisasi data dengan *min-max*, dilanjutkan proses perhitungan dengan algoritme SVM. Tahap-tahap sebelumnya, dilakukan untuk membantu menambah akurasi yang didapatkan dari perhitungan algoritme SVM. Beberapa langkah dilakukan dalam perhitungan SVM, antara lain seperti perhitungan kernel yang ditunjukkan pada Tabel 4.18, perhitungan *error rate* yang ditunjukkan pada Tabel 4.19, perhitungan delta alfa yang ditunjukkan pada Tabel 4.20, perhitungan α baru yang ditunjukkan pada Tabel 4.21, perhitungan nilai bias yang ditunjukkan pada Tabel 4.22, dan perhitungan hasil klasifikasi data uji yang ditunjukkan pada Tabel 4.23.

Tabel 4.18 Hasil Perhitungan Kernel pada Data Latih

K(xi,xi)	d1	d2	d3	d4	d5	d6
d1	124.2933	1.21	1.981138	3.3124	3.3124	4.192375
d2	1.21	84.84333	1.21	1.21	1.21	1.21
d3	1.981138	1.21	8.747664	1.3924	1.3924	1.981138
d4	3.3124	1.21	1.3924	69.90704	4.192375	8.412013
d5	3.3124	1.21	1.3924	4.192375	18.4399	4.192375
d6	4.192375	1.21	1.981138	8.412013	4.192375	34.31879

Tabel 4.19 Hasil Perhitungan Error Rate pada Data Latih

E	d1	d2	d3	d4	d5	d6
	0	0	0	0	0	0
E	d1	d2	d3	d4	d5	d6
	0.011831	0.008509	0.001074	0.007495	0.001753	0.00379
E	d1	d2	d3	d4	d5	d6
	1.584492	0.830881	0.243034	0.383281	-0.13906	0.005818



Tabel 4.20 Hasil Perhitungan Delta Alfa pada Data Latih

d	d1	d2	d3	d4	d5	d6
	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
d	d1	d2	d3	d4	d5	d6
	9.88169E-05	9.88169E-05	9.98926E-05	9.92505E-05	9.98247E-05	9.9621E-05
d	d1	d2	d3	d4	d5	d6
	-5.84492E-05	1.69119E-05	7.56966E-05	6.16719E-05	0.000113906	9.9418E-05

Tabel 4.21 Hasil Perhitungan α Baru pada Data Latih

	d1	d2	d3	d4	d5	d6
α	0	0	0	0	0	0
α	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
α	0.01193	0.008608	0.001173	0.007595	0.001853	0.003889
α	1.584433	0.830898	0.24311	0.383343	-0.13894	0.005918

Tabel 4.22 Hasil Perhitungan Nilai Bias

Xi	K(xi,x+)	K(xi,x-)	$\alpha_{iyi}K(xi,x+)$	$\alpha_{iyi}K(xi,x-)$
d1	3.3124	124.2933	-5.24828	0.000238
d2	1.21	1.21	-1.00539	-0.05554
d3	1.3924	1.981138	-0.33851	-0.00097
d4	69.90704	3.3124	26.79837	-0.00125
d5	4.192375	3.3124	-0.5825	-0.00041
d6	8.412013	4.192375	0.04978	0.000918
Total			19.67348	-0.05702
B			9.808231	

Tabel 4.23 Hasil Perhitungan Klasifikasi pada Data Uji

$\alpha_{iyi}K(x,xi)$	d7	d8	d9
d1	-5.24828	-5.24828	-2.20616
d2	-1.00539	-1.00539	-1.00539
d3	-0.33851	-0.33851	-0.80528
d4	1.269785	0.025157	0.533767
d5	-0.70115	-0.46023	-0.19346
d6	0.019602	0.019602	0.00824
Total	-6.00393	-7.02724	-3.66829
Total + bias	3.804298	2.780987	6.139946
Kelas	1	1	1
Sentimen	Positif	Positif	Positif



4.7.11 Manualisasi Perhitungan Evaluasi

Untuk mengetahui akurasi dari perhitungan klasifikasi menggunakan algoritme SVM, dilakukan pencocokan hasil klasifikasi data uji yang didapatkan dengan label awal tiap dokumen. Pencocokan tersebut ditunjukkan pada Tabel 4.24.

Tabel 4.24 Hasil *Confusion Matrix*

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

Berdasarkan Tabel 4.24, didapatkan hasil perhitungan manualisasi terhadap tiga data uji. Pada tabel ditunjukkan bahwa satu diantara tiga data uji telah diklasifikasikan dengan benar, namun dua 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.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TN}+\text{TP}}{\text{TN}+\text{TP}+\text{FP}+\text{FN}} \\ &= \frac{0+1}{0+1+0+2} \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ &= \frac{1}{1+0} \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\ &= \frac{1}{1+2} \\ &= 0.33 \end{aligned}$$

$$\begin{aligned} F - \text{Measure} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \\ &= \frac{2 \times 1 \times 0.33}{1+0.33} \\ &= 0.496 \end{aligned}$$

4.8 Skenario Pengujian

Skenario pengujian akan digunakan untuk mengetahui tingkat akurasi sistem yang telah dibuat. Terdapat dua macam pengujian yang diterapkan, yaitu pengujian terhadap parameter *Support Vector Machine* (derajat pada Kernel Polinomial, *Learning Rate* dan nilai C) dan pengaruh penggunaan *Lexicon Based Features*.

4.8.1 Skenario Pengujian terhadap Parameter *Support Vector Machine*

Beberapa parameter digunakan dalam pengujian terhadap parameter *Support Vector Machine*, yaitu parameter *learning rate* dan parameter nilai c. Pengujian terhadap parameter *Support Vector Machine* berfungsi untuk mengetahui jumlah iterasi maksimum dan parameter-parameter paling optimal. Oleh karena itu, pada tabel skenario pengujian *learning rate* dan nilai C diberi ketentuan jumlah iterasi yang digunakan. Skenario pengujian pada parameter *Support Vector Machine* ditunjukkan pada Tabel 4.25 dan 4.26.

Tabel 4.25 Skenario Pengujian Konstanta *Learning Rate*

Iterasi	Evaluasi	<i>Learning Rate</i>									
		0.0001	0.006	0.008	0.015	0.02	0.03	0.04	0.045	0.1	1.2
50	<i>Accuracy</i>										
	<i>Precision</i>										
	<i>Recall</i>										
	<i>F-Measure</i>										
100	<i>Accuracy</i>										
	<i>Precision</i>										
	<i>Recall</i>										
	<i>F-Measure</i>										
150	<i>Accuracy</i>										
	<i>Precision</i>										
	<i>Recall</i>										
	<i>F-Measure</i>										
200	<i>Accuracy</i>										
	<i>Precision</i>										
	<i>Recall</i>										
	<i>F-Measure</i>										

Tabel 4.26 Skenario Pengujian Konstanta Nilai C

Iterasi	Evaluasi	Nilai C					
		10 ⁰	10 ¹	10 ²	10 ³	10 ⁴	10 ⁵
50	Accuracy						
	Precision						
	Recall						
	F-Measure						
100	Accuracy						
	Precision						
	Recall						
	F-Measure						
150	Accuracy						
	Precision						
	Recall						
	F-Measure						
200	Accuracy						
	Precision						
	Recall						
	F-Measure						

4.8.2 Skenario Pengujian Implementasi *Lexicon Based Features*

Pengujian dengan mengimplementasikan *Lexicon Based Features*, bertujuan untuk membandingkan hasil akurasi ketika sistem menggunakan *Lexicon Based Features* dan tidak menggunakan *Lexicon Based Features*. Pada proses penggunaan *Lexicon Based Features*, di dalamnya digunakan proses normalisasi dengan menggunakan *min-max normalization*. Setelah mengimplementasikan pengujian ini, diharapkan dapat diketahui apakah penggunaan *Lexicon Based Features* dengan fitur *term* bersentimen positif atau negatif dapat memberi pengaruh yang besar atau tidak dalam proses klasifikasi pada sistem. Pada pengujian ini juga diberikan ketentuan jumlah iterasi maksimum yang digunakan. Skenario pengujian ditunjukkan pada Tabel 4.27.

Tabel 4.27 Skenario Pengujian Implementasi *Lexicon Based Features*

Iterasi	Evaluasi	<i>Lexicon Based Features</i> dengan Normalisasi <i>Min-Max</i>	Tanpa <i>Lexicon Based Features</i>
50	Accuracy		
	Precision		
	Recall		
	F-Measure		
100	Accuracy		
	Precision		



Tabel 4.27 Skenario Pengujian Implementasi *Lexicon Based Features* (lanjutan)

Iterasi	Evaluasi	<i>Lexicon Based Features</i> dengan Normalisasi <i>Min-Max</i>	Tanpa <i>Lexicon Based Features</i>
100	<i>Recall</i>		
	<i>F-Measure</i>		
150	<i>Accuracy</i>		
	<i>Precision</i>		
	<i>Recall</i>		
	<i>F-Measure</i>		
200	<i>Accuracy</i>		
	<i>Precision</i>		
	<i>Recall</i>		
	<i>F-Measure</i>		

4.9 Penarikan Kesimpulan

Penarikan kesimpulan akan dilakukan setelah pengujian dilakukan. Hasil dari pengujian akan dianalisis kemudian akan divisualisasikan dengan grafik atau tabel. Setelah diketahui hasil dari pengujian akan diketahui kekurangan dan kelebihan dari sistem maupun isi dokumen. Penulis dapat menuliskan kekurangan dan kelebihan tersebut pada bagian kesimpulan yang kemudian dapat penulis berikan saran agar penelitian selanjutnya dapat menghasilkan hasil yang lebih baik dan mempertahankan yang sudah baik.

4.10 Implementasi

Pada implementasi sistem, dilakukan pembuatan program menggunakan Bahasa pemrograman Python. Dilakukan penguraian kode program tahap-tahap analisis sentimen tentang opini maskapai penerbangan pada dokumen Twitter yang menggunakan *Support Vector Machine* sebagai algoritme klasifikasi. Pada bab-bab sebelumnya, sudah dijelaskan apa saja tahapan dari proses analisis sentimen. Tahap-tahap tersebut meliputi *pre-processing*, pembobotan TF-IDF, *Lexicon Based Features*, dan klasifikasi menggunakan algoritme *Support Vector Machine* (SVM).

4.10.1 Pre-Processing

Langkah pertama sebelum masuk ke pengolahan data yang lebih lanjut, dilakukan pemrosesan teks atau *pre-processing text*. Terdapat beberapa tahap di dalam *pre-processing* tersebut antara lain *case folding*, *cleaning*, *translation*,

stemming, tokenisasi, dan stopwords removal. Implementasi dari tahap *pre-processing* ditunjukkan pada Kode Program 4.1.

```

1 class preprocessing():
2     def casefolding(self, arrayDoc):
3         hasil = []
4         for i in range(len(arrayDoc)):
5             for j in range(len(arrayDoc[i])):
6                 lowers = arrayDoc[i][j].lower()
7                 hasil.append(lowers)
8         return hasil
9
10    def cleaning(self, doc):
11        hasil_cleaning=[]
12        arrayganti = ['batik air', 'lion air', 'garuda
13 indonesia']
14        arrayhasilganti =
15 ['batikair', 'lionair', 'garudaindonesia']
16        for i in range(len(doc)):
17            hilangusername = re.sub(r'(\@[a-zA-Z0-
18 9]*)', '', str(doc[i]))
19            hilanghashtag = re.sub(r'(\#[a-zA-Z0-
20 9]*)', '', hilangusername)
21            hilanglink = re.sub(r'(http[a-zA-Z0-9\\-
22 \:\/\.\.]*)', '', hilanghashtag)
23            hilangtandabaca = re.sub(r'([\^a-zA-Z0-
24 9\s+])', '', hilanglink)
25            hilangangka = re.sub(r'([0-
26 9+])', '', hilangtandabaca)
27            ganti =
28 re.sub(arrayganti[0], arrayhasilganti[0], hilangangka)
29            ganti =
30 re.sub(arrayganti[1], arrayhasilganti[1], ganti)
31            ganti = re.sub(arrayganti[2],
32 arrayhasilganti[2], ganti)
33            if(ganti!=''):
34                hasil_cleaning.append(ganti)
35        return hasil_cleaning
36
37    def translation(self, text):
38        hasiltranslation=[]
39        tr = Translator()
40
41        tr.set_key('trnsl.1.1.20181208T055507Z.aa500fcbe36fd
42 e9.8075d3e45c11bf0c8f58efc6072e8b190b0e8957')
43        tr.set_from_lang('en')
44        tr.set_to_lang('id')
45        for i in range(len(text)):
46            tr.set_text(text[i])
47            hT = tr.translate()
48            hasiltranslation.append(hT)
49        return hasiltranslation
50
51    def Stemming(self, text):
52        factory= StemmerFactory()
53        stemmer = factory.create_stemmer()
54        hasilstem =[]
55        print('Stemm ', text)
56        for i in range(len(text)):

```

```

57         perdoc=[]
58         for j in range(len(text[i])):
59
60     perdoc.append(stemmer.stem(text[i][j]))
61         hasilstem.append(perdoc)
62     return hasilstem
63
64     def tokenisasi(self,text):
65         hasiltokenisasi = []
66         for i in range(len(text)):
67             hasil=text[i].split(' ')
68             hasiltokenisasi.append(hasil)
69         return hasiltokenisasi
70
71     def stopwordremoval(self,text):
72         stopword = []
73         with open('stopword.csv', 'r') as
74     stopwordList:
75         reader = csv.reader(stopwordList)
76         for i in reader:
77             stopword.append(i[0])
78         term=[]
79         for i in range(len(text)):
80             termperdoc= []
81             for j in range(len(text[i])):
82                 if text[i][j] not in stopword:
83                     if(text[i][j]!=''):
84
85     termperdoc.append(text[i][j].encode('utf-8'))
86         term.append(termperdoc)
87     return term
88
89     def termUnik(self,HasilStopwordRemoval):
90         term=[]
91         for i in range (len(HasilStopwordRemoval)):
92             for j in range
93 (len(HasilStopwordRemoval[i])):
94                 if HasilStopwordRemoval[i][j] not in
95     term:
96
97     term.append(HasilStopwordRemoval[i][j])
98         return term
99

```

Kode Program 4.1 Implementasi Tahap *Pre-Processing*

Kode Program 4.1 dapat diuraikan sebagai berikut.

- Baris 2-8 adalah proses *case folding*. Terjadi proses perulangan pada panjang data yang akan merubah seluruh huruf besar menjadi huruf kecil.
- Baris 10-36 adalah proses *cleaning*. Terjadi proses perulangan pada panjang data yang akan merubah beberapa ketentuan seperti berikut. Apabila kalimat tidak kosong dan terdapat *username*, angka, tanda baca, *hashtag*, *link* atau url akan diganti dengan spasi sehingga akan hilang. Kemudian, apabila terdapat tulisan “batik air”, “garuda Indonesia”, “sriwijaya air”, “lion air” akan diubah menjadi “batikair”, “garudaindonesia”, “sriwijayaair”, dan “lionair”.

- Baris 38-50 adalah proses *translation*. Dilakukan proses perulangan pada panjang data untuk menerjemahkan tiap kata berbahasa Inggris menjadi Bahasa Indonesia dengan bantuan Yandex API, sehingga akan memanggil kunci atau *key* yang telah disediakan pada akun Yandex yang telah didaftarkan sebelumnya untuk mendapatkan akses melakukan terjemah kata tersebut.
- Baris 52-63 adalah proses *stemming*. Terjadi proses perulangan pada panjang data yang kemudian dengan bantuan Sastrawi, akan dipanggil *Stemmer Factory*. Proses ini berfungsi untuk menghilangkan imbuhan pada tiap kata sehingga hasil yang dikembalikan berupa kata-kata dasar.
- Baris 65-70 adalah proses tokenisasi. Terjadi proses perulangan pada panjang data yang kemudian dilakukan pemisahan atau pemecahan dokumen menjadi per kata.
- Baris 72-88 adalah proses *stopword removal*. Terjadi proses perulangan pada panjang data, namun diawali terlebih dahulu dengan pembukaan dan pembacaan file csv bernama *stopword.csv* yang kemudian dicocokkan dengan tiap kata dari hasil tahap sebelumnya. Apabila kata pada hasil sebelumnya terdapat di dalam *stopword list*, maka kata tersebut akan dihilangkan. Sehingga hasil yang akan dikembalikan dari proses ini adalah kata-kata yang tidak terdapat di dalam *stopword list*.
- Baris 90-99 adalah proses didapatkannya *term* unik yang dijadikan fitur pada penelitian ini. Akan dilakukan penghapusan *term* yang duplikat.

4.10.2 Pembobotan TF-IDF

Selanjutnya, dilakukan pembobotan dengan *Term Frequency-Inverse Document Frequency* (TF-IDF). Beberapa tahap dalam pembobotan TF-IDF antara lain mencari nilai TF, WTF, IDF, dan terakhir mengalikan nilai TF dan IDF untuk mendapatkan *Weight* TF-IDF. Implementasi dari tahap pembobotan TF-IDF ditunjukkan pada Kode Program 4.2.

```

1 class bobot:
2     def findWeightTF(self, document, term):
3         tf = []
4         for i in range(len(document)):
5             term_perdokumen = {}
6             docs = ' '.join(document[i])
7             for j in term:
8                 docterm = docs.count(j)
9                 term_perdokumen[j] =
10 1+math.log(docterm) if docterm>0 else 0
11             tf.append(term_perdokumen)
12         return tf
13
14     def findIDF(self, tf, term, doc):
15         idf = {}
16         for j in term:
17             for i in range(len(tf)):
18                 countdf = 0
19                 if tf[i][j]>0:
20                     countdf=countdf+1
21

```

```

22         idf[j] = math.log(len(doc)/countdf) if
23         countdf>0 else 0
24         return idf
25
26     def countWTFIDF(self,wtf,idf):
27         tfidf=wtf
28         for i in range(len(wtf)):
29             for j in idf:
30                 tfidf[i][j] = wtf[i][j]*idf[j]
                 return tfidf

```

Kode Program 4.2 Implementasi Tahap Pembobotan TF-IDF

Kode Program 4.2 dapat diuraikan sebagai berikut.

- Baris 2-12 adalah proses perhitungan nilai *Weight* TF. Tf pada baris 3 adalah variabel untuk menyimpan hasil perhitungan TF dalam bentuk *list*. Terjadi proses perulangan pada panjang dokumen yang kemudian dilakukan perhitungan pada tiap dokumen atau data ada berapa jumlah sebuah kata dalam setiap dokumen. Selanjutnya dilakukan perhitungan log TF untuk mendapatkan nilai Wtf, jika bernilai 0 maka menghasilkan nilai 0 juga.
- Baris 14-23 adalah proses perhitungan nilai IDF. Idf pada baris ke 15 merupakan variabel untuk menyimpan hasil perhitungan IDF. Terjadi proses perulangan pada hasil nilai TF yang kemudian dilakukan perhitungan jumlah dokumen yang mengandung setiap kata. Selanjutnya dilakukan perhitungan dengan rumus untuk mendapatkan nilai IDF, jika bernilai 0 maka menghasilkan nilai 0 juga.
- Baris 25-30 adalah proses menghitung nilai TF-IDF. Dilakukan perkalian hasil nilai Wtf dengan IDF. Kemudian hasil dari perhitungan TF-IDF disimpan ke dalam variabel tfidf.

4.10.3 *Lexicon Based Features*

Sebelum memasuki tahap klasifikasi dengan algoritme SVM, dilakukan pembobotan *Lexicon Based Features*. Dilakukan pencocokan pada kamus *Lexicon* terlebih dahulu kemudian dinormalisasi dengan *min-max normalization*. Implementasi dari tahap *Lexicon Based Features* ditunjukkan pada Kode Program 4.3.

```

1 class lexicon():
2     def importLexicon(self):
3         datalexicon={}
4         with open('lexicon.txt', 'r') as file:
5             reader = csv.reader(file)
6             for i in reader:
7                 dt = i[0].strip().split(';')
8                 datalexicon[dt[0]] = int(dt[1])
9         return datalexicon
10
11     def hitungLexicon(self,text,daftarLexicon):
12         hasilCount=[]

```

```

13         pos=0
14         neg=0
15         for i in range(len(text)):
16             if(text[i] in daftarLexicon):
17                 if(daftarLexicon[text[i]]>0):
18                     pos+=daftarLexicon[text[i]]
19                 else:
20                     neg+=daftarLexicon[text[i]]
21             hasilCount.append((pos, neg))
22         return hasilCount
23
24     def
25     normalisasi(self, hasilCount, maxBaru=0.9, minBaru=0.1):
26         hasilNorm=[]
27         for i in range(len(hasilCount)):
28             maxData = max(hasilCount[i])
29             minData = min(hasilCount[i])
30             print(maxData, minData)
31             hasilnormpos = ((hasilCount[i][0]-
32 minData) / (maxData-minData)) * (maxBaru-
33 minBaru) + minBaru #xpos-min/max-min
34             hasilnormneg = ((hasilCount[i][1]-
35 minData) / (maxData-minData)) * (maxBaru-
36 minBaru) + minBaru #xneg-min/max-min
37
38         hasilNorm.append((hasilnormpos, hasilnormneg))
39         return hasilNorm

```

Kode Program 4.3 Implementasi Tahap *Lexicon Based Features*

Kode Program 4.3 dapat diuraikan sebagai berikut.

- Baris 2-9 adalah proses memunculkan dan membaca data dengan nama *file lexicon*. *File* tersebut berisikan kata-kata yang sudah diberi label positif dengan angka 1 dan negatif dengan angka -1.
- Baris 11-22 adalah proses perhitungan *lexicon*. Variabel hasil *Count* untuk menyimpan hasil perhitungan *lexicon* dalam bentuk *list*. Terjadi perulangan pada panjang data yang kemudian akan dihitung jumlah kata positif dan jumlah kata negatif pada setiap dokumen berdasarkan ketentuan yang terdapat pada *file lexicon* yang sudah di buka dan *load* sebelumnya. Pos dan neg diinisialisasikan dengan nilai 0 terlebih dahulu karena nantinya akan bertambah ketika pada sebuah dokumen mengandung kata positif ataupun negatif.
- Baris 24-39 adalah proses normalisasi data setelah perhitungan *lexicon*. Variabel hasil Norm digunakan untuk menyimpan hasil dari normalisasi dalam bentuk list. Terjadi proses perulangan hasil *Count*, yaitu hasil perhitungan *lexicon* yang akan dihitung *max* dan *min*nya terlebih dahulu, kemudian dapat masuk ke dalam perhitungan normalisasi sesuai rumus dengan diketahui *min*. baru dan *max*. baru di awal perhitungan.

4.10.4 Klasifikasi *Support Vector Machine*

Untuk melakukan klasifikasi dokumen Twitter berdasarkan sentimen positif atau negatif, digunakan algoritme *Support Vector Machine* (SVM). Terdapat

beberapa tahap dalam algoritme SVM, seperti yang telah dijelaskan pada bagian 2.7.1 sebelumnya. Implementasi dari tahap klasifikasi dengan algoritme *Support Vector Machine* (SVM) ditunjukkan pada Kode Program 4.4.

```

1      def transposeData(self, data):
2          hasil = []
3          for i in range(len(data[0])):
4              temp = []
5              for j in range(len(data)):
6                  temp.append(data[j][i])
7              hasil.append(temp)
8          print("hasil transpose")
9          print(hasil)
10         return hasil
11
12         def getKernel(self, data, dataTranspose, c):
13             hasilKernel = []
14             self.C=c
15             hasil = []
16             for i in range(len(data)):
17                 temp=[]
18                 for k in range(len(dataTranspose[i])):
19                     hasilJ = 0
20                     for j in range(len(dataTranspose)):
21                         hasilJ += data[i][j] *
22 dataTranspose[j][k]
23                     hasilJ = math.pow(hasilJ+c,2)
24                     temp.append(hasilJ)
25                 hasil.append(temp)
26                 print('hasil get kernel')
27                 print(hasil)
28                 return hasil
29
30         def getMatrixHessian(self, kernel, kelas, lamda):
31             x,y = len(kernel), len(kernel[0])
32             hessian = np.zeros((x,y))
33             for i in range(x):
34                 for j in range(y):
35                     hessian[i][j] =
36 (kelas[i]*kelas[j])*(kernel[i][j]+pow(lamda,2))
37             return hessian
38
39         def
40 getSequential(self, hessian, gamma, iterasiMax=1000, eps=
41 0.0001):
42             x,y = hessian.shape
43             alfa = np.zeros(x)
44             e = np.zeros(y)
45             d = np.zeros(y)
46             iterasi = 0
47             while iterasi<iterasiMax:
48                 for i in range(x):
49                     for j in range(y):
50                         e[i] += hessian[i,j]*alfa[j]
51                 for i in range(x):
52                     print('e[i]',e[i])
53                     maks = max((gamma*(1-e[i])), -alfa[i])
54                     print('maks', maks)

```

```

55         d[i]+= min(maks, (self.C-alfa[i]))
56         print('d[i]',d[i])
57         alfa[i] = alfa[i]+d[i]
58         print('alpha', alfa)
59         iterasi+=1
60     print(hessian)
61     print(self.C)
62     return alfa
63
64     def getAlfaMax(self, alfa, y):
65         pos = []
66         neg=[]
67         for i in range(len(alfa)):
68             if y[i] == 1:
69                 pos.append(alfa[i])
70             elif y[i] == -1:
71                 neg.append(alfa[i])
72         return max(pos),max(neg)
73
74     def getB(self, alfa, kernel, y):
75         print(alfa)
76         alfa = list(alfa)
77         pos,neg = self.getAlfaMax(alfa,y)
78         indexpos= alfa.index(pos)
79         indexneg= alfa.index(neg)
80         pos=0
81         neg=0
82         for i in range(len(kernel)):
83             pos+= alfa[i]*y[i]*kernel[i][indexpos]
84             neg+= alfa[i]*y[i]*kernel[i][indexneg]
85         b= 0.5*(pos+neg)
86         return b
87
88     def getKlasifikasi(self,
89 datauji, dataLatih, alfa, kelas, bias):
90         hasilkernel=0
91         for i in range(len(dataLatih)):
92             sums = self.C
93             for j in range(len(datauji)):
94                 sums+= datauji[j]*dataLatih[i][j]
95             hasilkernel += alfa[i]*kelas[i]*sums
96         jml =hasilkernel+bias
97         print('jumlah', bias)
98         if jml>0:
99             return 1
100        return -1

```

Kode Program 4.4 Implementasi Tahap Klasifikasi *Support Vector Machine*

Kode Program 4.4 dapat diuraikan sebagai berikut.

- Baris 1-10 adalah langkah untuk melakukan transposisi dari data yang telah dibentuk.
- Baris 12-28 adalah langkah perhitungan kernel, di mana akan dilakukan perkalian matriks yang telah ditranposisi sebelumnya yang kemudian dijumlahkan dengan nilai c yang telah diinisialisasikan, selanjutnya dipangkatkan dengan nilai *degree*. Nilai c yang digunakan adalah 1 dan nilai *degree* yang digunakan adalah 2.

- Baris 30-37 adalah proses menghitung nilai matriks Hessian. Perhitungan dilakukan dengan mengalikan kelas setiap dokumen dengan hasil kernel yang telah didapatkan sebelumnya, yang kemudian dijumlahkan nilai lambda kuadrat. Nilai lambda yang digunakan adalah 0.5.
- Baris 39-62 adalah proses *sequential learning*. Pertama-tama dilakukan pemberian nilai alfa awal yaitu 0. Proses ini dilakukan secara beberapa kali (iterasi), pada setiap satu kali iterasi, akan dihitung nilai e, d, dan alfa. E adalah *error rate* dan d adalah delta alfa. Kemudian nilai delta alfa tersebut berguna untuk memperbaharui nilai alfa. Jumlah iterasi yang akan dilakukan sampai esuai yang diinginkan.
- Baris 64-72 adalah proses mencari nilai minimum dan maksimum untuk delta alfa.
- Baris 74-86 adalah proses perhitungan nilai bias menjumlahkan total hasil dari kelas positif dan negatif yang dikalikan dengan 0.5. akan dijumlahkan.
- Baris 88-100 adalah proses klasifikasi untuk data uji. Diawali dengan mencari nilai kernel data uji tersebut kemudian dikalikan dengan kelas dari setiap dokumen data latih dan juga alfa dari setiap dokumen laith.

4.10.5 Evaluasi

Setelah mendapatkan hasil klasifikasi dokumen, langkah terakhir yang dilakukan adalah tahap evaluasi atau pengujian. Tahap ini berguna untuk mengetahui akurasi sistem dengan algoritme yang digunakan. Evaluasi yang digunakan antara lain menghitung *confusion matrix*, *accuracy*, *precision*, *recall*, dan *f-measure*.

4.10.5.1 Confusion Matrix

Pada evaluasi, tahap pertama adalah menghitung *confusion matrix*. Dilakukan perhitungan jumlah data uji yang benar dan salah dalam proses klasifikasi. Implementasi pada tahap *Confusion Matrix* ditunjukkan pada Kode Program 4.5.

```

1 class evaluasi():
2     def conf_matrix(self, kelas, testing_kelas):
3         conf = [[0,0],[0,0]]
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        return conf

```

Kode Program 4.5 Implementasi Tahap Evaluasi *Confusion Matrix*

Kode Program 4.5 dapat diuraikan sebagai berikut.

- Baris 2-13 adalah evaluasi *confusion matrix*. Pada baris awal, diketahui nilai *list* awal pada *confusion matrix*. Terjadi proses perulangan untuk

memeriksa setiap kelas latih dan kelas uji. Apabila 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 negative*. Pada baris kedua, kolom kesatu menunjukkan nilai *false positive*. Pada baris kedua, kolom kedua menunjukkan nilai *true negative*.

4.10.5.2 Accuracy, Precision, Recall, F-Measure

Selanjutnya, dilakukan perhitungan yang digunakan untuk mengetahui kinerja sistem antara lain *accuracy* (mengetahui ketepatan sistem dalam menggunakan algoritme SVM), *precision* (mengetahui ketepatan sistem pada proses klasifikasi), *recall* (mengetahui tingkat keberhasilan sistem), dan *f-measure* (kombinasi pengujian dengan *precision* dan *recall*). Implementasi pada tahap *accuracy*, *precision*, *recall*, dan *f-measure* ditunjukkan pada Kode Program 4.6.

```

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    return accuracy, precision, recall, fmeas

```

Kode Program 4.6 Implementasi Tahap Evaluasi Accuracy, Precision, Recall, F-Measure

Kode Program 4.6 dapat diuraikan sebagai berikut.

- Baris 1-10 adalah tahap evaluasi *Accuracy, Precision, Recall, F-Measure*. Setelah mendefinisikan fungsi, 4 baris di bawahnya mendeklarasikan nilai *true positive, false negative, false positive, dan true negative*. Kemudian dilakukan perhitungan sesuai rumus masing-masing *tp, fn, fp, tn, accuracy, precision, recall, dan f-measure*.

BAB 5 HASIL DAN PEMBAHASAN

5.1 Hasil

5.1.1 Hasil Pengujian terhadap Parameter *Support Vector Machine*

Bertujuan untuk mengetahui besarnya pengaruh parameter yang tepat yang terdapat pada metode *Support Vector Machine*, pengujian terhadap parameter SVM digunakan dalam penelitian ini. Parameter yang akan diuji adalah *learning rate* dan nilai C. Akan dicari tingkat akurasi tertinggi pada pengujian parameter SVM ini. Pada pengujian ini akan dilihat juga iterasi maksimum yang paling optimal, selain melihat dari akurasi, presisi, *recall*, dan *f-measure* tertingginya. Pengujian terhadap parameter SVM dijelaskan dalam skenario pengujian 5.1.1.1.

5.1.1.1 Skenario Pengujian terhadap Parameter *Support Vector Machine*

Pengujian yang dilakukan pada parameter *Support Vector Machine* dilakukan dengan menguji nilai *learning rate* dan nilai C pada metode SVM. Pengujian yang dilakukan dengan menentukan dan memperhatikan jumlah iterasi maksimum yang dilakukan oleh sistem pada proses *training Support Vector Machine*. Diawal pengujian, digunakan 250 data *tweets*.

Untuk memulai pengujian, dilakukan pengujian pada pengaruh nilai *learning rate*. Nilai *learning rate* yang digunakan pada pengujian ini adalah 0.0001, 0.006, 0.008, 0.015, 0.02, 0.03, 0.04, 0.045, 0.1 dan 1.2 dengan digunakan nilai C adalah 1. Kemudian, nilai C yang diujikan adalah 10^0 , 10^1 , 10^2 , 10^3 , 10^4 , dan 10^5 dengan digunakan nilai *learning rate* sebesar 0.0001. Iterasi maksimum yang dilakukan dalam pengujian ini adalah 50, 100, 150, dan 200 kali. Hasil pengujian yang telah dilakukan ditunjukkan pada Tabel 5.1 dan 5.2.

Tabel 5.1 Hasil Pengujian Konstanta *Learning Rate*

Iterasi	Evaluasi	<i>Learning Rate</i>									
		0.0001	0.006	0.008	0.015	0.02	0.03	0.04	0.045	0.1	1.2
50	<i>Accuracy</i>	0.4	0.6	0.6	0.4	0.4	0.4	0.4	0.4	0.4	0.6
	<i>Precision</i>	0.4	0	0	0.4	0.4	0.4	0.4	0.4	0.4	0
	<i>Recall</i>	1	0	0	1	1	1	1	1	1	0
	<i>F-Measure</i>	0.5714	0	0	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714	0
100	<i>Accuracy</i>	0.6	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	<i>Precision</i>	0	0	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	<i>Recall</i>	0	0	1	1	1	1	1	1	1	1
	<i>F-Measure</i>	0	0	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714
150	<i>Accuracy</i>	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
	<i>Precision</i>	0	0	0	0	0	0	0	0	0	0
	<i>Recall</i>	0	0	0	0	0	0	0	0	0	0

Tabel 5.1 Hasil Pengujian Konstanta *Learning Rate* (lanjutan)

Iterasi	Evaluasi	<i>Learning Rate</i>									
		0.0001	0.006	0.008	0.015	0.02	0.03	0.04	0.045	0.1	1.2
150	<i>F-Measure</i>	0	0	0	0	0	0	0	0	0	0
200	<i>Accuracy</i>	0.6	0.6	0.6	0.4	0.4	0.4	0.4	0.4	0.4	0.6
	<i>Precision</i>	0	0	0	0.4	0.4	0.4	0.4	0.4	0.4	0
	<i>Recall</i>	0	0	0	1	1	1	1	1	1	0
	<i>F-Measure</i>	0	0	0	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714	0

Tabel 5.2 Hasil Pengujian Konstanta Nilai C

Iterasi	Evaluasi	Nilai C					
		10 ⁰	10 ¹	10 ²	10 ³	10 ⁴	10 ⁵
50	<i>Accuracy</i>	0.4	0.4	0.4	0.4	0.6	0.6
	<i>Precision</i>	0.4	0.4	0.4	0.4	0	0
	<i>Recall</i>	1	1	1	1	0	0
	<i>F-Measure</i>	0.5714	0.5714	0.5714	0.5714	0	0
100	<i>Accuracy</i>	0.6	0.4	0.6	0.6	0.4	0.4
	<i>Precision</i>	0	0.4	0	0	0.4	0.4
	<i>Recall</i>	0	1	0	0	1	1
	<i>F-Measure</i>	0	0.5714	0	0	0.5714	0.5714
150	<i>Accuracy</i>	0.6	0.6	0.6	0.4	0.6	0.6
	<i>Precision</i>	0	0	0	0.4	0	0
	<i>Recall</i>	0	0	0	1	0	0
	<i>F-Measure</i>	0	0	0	0.5714	0	0
200	<i>Accuracy</i>	0.4	0.6	0.4	0.6	0.6	0.6
	<i>Precision</i>	0.4	0	0.4	0	0	0
	<i>Recall</i>	1	0	1	0	0	0
	<i>F-Measure</i>	0.5714	0	0.5714	0	0	0

5.1.2 Hasil Pengujian 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 jumlah iterasi maksimum 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.

5.1.2.1 Skenario Pengujian Implementasi *Lexicon Based Features*

Dalam melakukan pengujian pada tahapan ini, akan mempertimbangkan jumlah iterasi maksimum sebanyak 50, 100, 150, dan 200 kali. 2 tahap akan dilakukan, yang pertama dengan mengimplementasikan *lexicon based features* dan yang kedua tanpa mengimplementasikan *lexicon based features*.

Untuk memulai pengujian pada tahapan ini, nilai *learning rate* adalah 0.0001, dan nilai C adalah 1. Hasil pengujian dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil Pengujian Implementasi *Lexicon Based Features*

Iterasi	Evaluasi	<i>Lexicon Based Features</i> dengan Normalisasi <i>Min-Max</i>	Tanpa <i>Lexicon Based Features</i>
50	<i>Accuracy</i>	0.4	0.25
	<i>Precision</i>	0.6	0.25
	<i>Recall</i>	1	0.5
	<i>F-Measure</i>	0.75	0.333
100	<i>Accuracy</i>	0.6	0.6
	<i>Precision</i>	0	0.5
	<i>Recall</i>	0	1
	<i>F-Measure</i>	0	0.666
150	<i>Accuracy</i>	0.6	0.4
	<i>Precision</i>	0	0
	<i>Recall</i>	0	0
	<i>F-Measure</i>	0	0
200	<i>Accuracy</i>	0.6	0.6
	<i>Precision</i>	0	0
	<i>Recall</i>	0	0
	<i>F-Measure</i>	0	0

5.2 Pembahasan

Hasil pengujian yang dilakukan terhadap parameter *Support Vector Machine* yang menggunakan iterasi maksimum sebanyak 50, 100, 150, dan 200 kali pada 250 tweets berupa opini tentang maskapai penerbangan ditunjukkan pada Tabel 5.1 dan 5.2. Pada Tabel 5.1, parameter *learning rate* yang digunakan adalah 0.0001, 0.006, 0.008, 0.015, 0.02, 0.03, 0.04, 0.045, 0.1, dan 1.2 dengan nilai C sebesar 1. Didapatkan hasil sebagai berikut.

- Pada iterasi sebanyak 50 kali, nilai *learning rate* yang terlihat optimal dan stabil ketika digunakan *learning rate* dalam rentang nilai 0.015 sampai dengan 0. Dapat dilihat ketika menggunakan nilai 1.2 grafik langsung turun drastis.

- Pada iterasi sebanyak 100 kali, nilai *learning rate* terlihat optimal dan stabil ketika digunakan *learning rate* dalam rentang nilai 0.08 sampai dengan 1.2. pada iterasi sebanyak 100 kali juga menunjukkan bahwa ketika dilakukan sebanyak 100 kali, nilai akurasi, presisi, *recall*, dan *f-measure* terlihat stabil dan tidak mengalami penurunan drastis seperti yang terjadi sebelumnya pada iterasi sebanyak 50 kali. Dapat dikatakan bahwa iterasi sebanyak 100 kali adalah jumlah yang lebih optimal dibandingkan iterasi sebanyak 50 kali.
- Pada iterasi sebanyak 150 kali memang terlihat stabil, namun banyak hasil yang bernilai 0 pada hasil *presisi*, *recall*, dan *f-measure*.
- Pada iterasi sebanyak 200 kali, terlihat serupa dengan iterasi sebanyak 50 kali. Terdapat kenaikan dan penurunan ketika menggunakan *learning rate* yang bervariasi. Nilai *learning rate* terlihat stabil dan optimal ketika *learning rate* yang digunakan adalah dalam rentang 0.015 sampai dengan 0.1.
- Parameter *learning rate* paling optimal dapat diambil dari nilai yang berada di tengah seperti 0.03 yang selalu terlihat stabil dilihat dari nilai akurasi, presisi, *recall*, dan *f-measure*. Jumlah iterasi optimal yang dapat dilakukan adalah sebanyak 150 kali, karena pada jumlah iterasi tersebut terlihat adanya kenaikan dan stabil sampai akhir. Kalau pun terdapat sebuah penurunan, itu hanya sedikit dan kemudian langsung meningkat lagi.

Kemudian, pada Tabel 5.2, parameter nilai c yang digunakan adalah 1, 10, 100, 1000, dan 100000 dengan nilai *learning rate* sebesar 0.0001. Didapatkan hasil sebagai berikut.

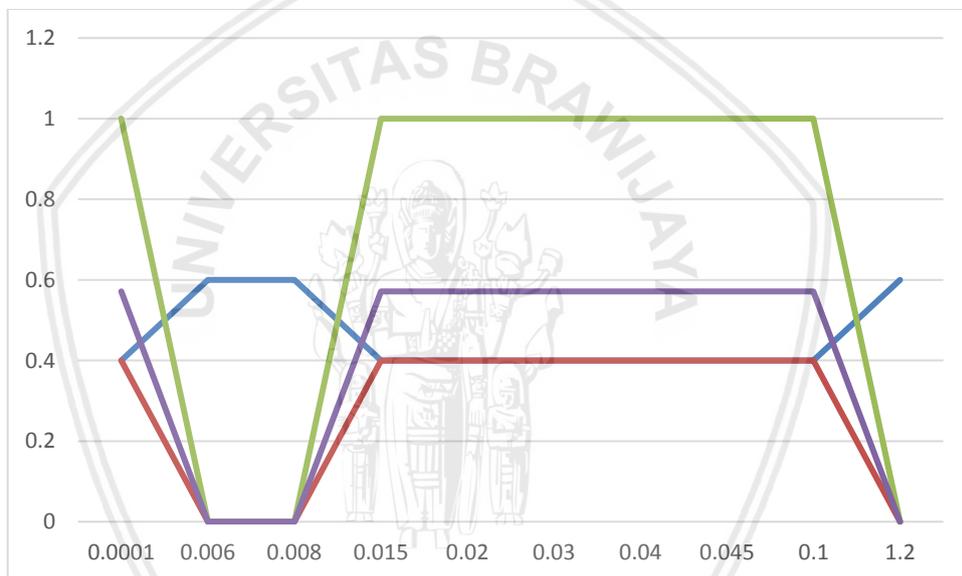
- Pada iterasi sebanyak 50 kali, nilai c yang terlihat optimal ketika digunakan nilai c dalam rentang 1 sampai dengan 1000. Dapat dilihat ketika menggunakan nilai 10000 grafik langsung turun drastis.
- Pada iterasi sebanyak 100 kali, nilai c terlihat optimal ketika digunakan nilai c sebesar 10 dan 10000. Dapat dilihat terjadi peningkatan drastis pada kedua nilai tersebut dari nilai-nilai c sebelumnya, terutama pada nilai *f-measure*-nya. Namun, jika dilihat dari iterasi sebelumnya, nilai 10000 tidak termasuk dalam nilai yang optimal, bahkan pada iterasi sebelumnya ketika nilai c adalah 10000 terjadi penurunan drastis.
- Pada iterasi sebanyak 150 kali, nilai c terlihat optimal ketika digunakan nilai c sebesar 1000. Namun pada iterasi sebanyak 150 kali, tidak sebaik ketika iterasi dilakukan sebanyak 100 kali. Di kedua iterasi tersebut memang sama-sama terjadi penurunan, namun pada iterasi sebanyak 100 kali terjadi peningkatan kembali dan stabil dan pada iterasi sebanyak 150 kali terjadi penurunan di akhir dan tidak ada peningkatan kembali.
- Pada iterasi sebanyak 200 kali, terlihat nilai c yang optimal ketika digunakan nilai c sebesar 1 dan 1000. Pada iterasi sebanyak 200 kali,

terdapat kenaikan dan penurunan ketika menggunakan nilai c yang bervariasi. Namun, di akhir tidak terjadi peningkatan kembali.

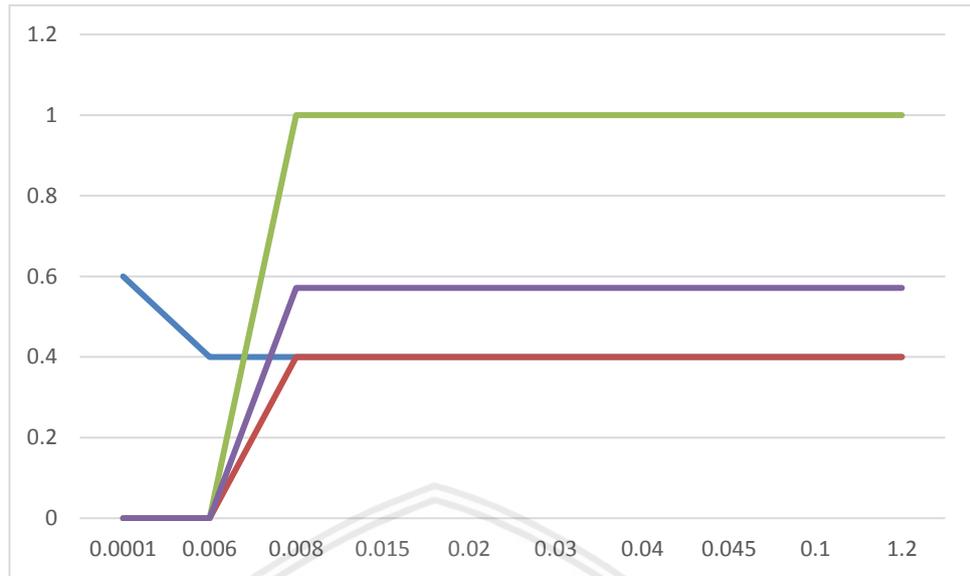
- Berdasarkan penjelasan sebelumnya, didapatkan parameter nilai c paling optimal adalah 10 dengan dilakukan iterasi sebanyak 100 kali.

Sedangkan pada Tabel 5.3, dapat dilihat perbandingannya secara jelas ketika digunakan *lexicon* dan tidak. Terutama pada iterasi sebanyak 50 kali, terlihat bahwa ketika *lexicon* digunakan akan lebih tinggi akurasi, presisi, dan *recall*-nya dibandingkan ketika tidak menggunakan *lexicon*. Namun, untuk nilai *f-measure* terlihat seimbang. Pada jumlah iterasi lainnya, tidak seluruhnya mendapatkan nilai, banyak yang bernilai 0.

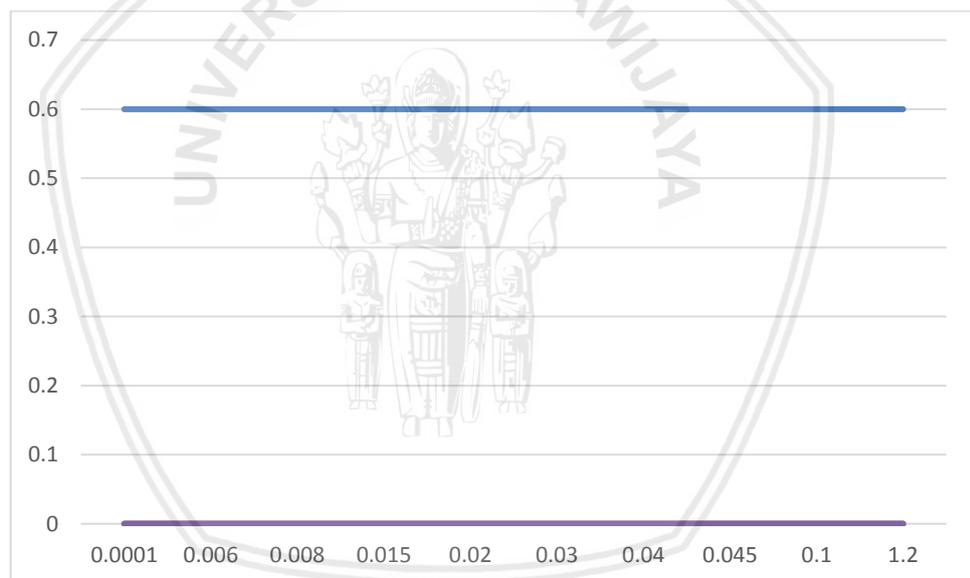
Grafik kenaikan dan penurunan akurasi, presisi, *recall*, dan *f-measure* dengan memperhatikan jumlah iterasi maksimum dapat dilihat pada Gambar 5.1 sampai dengan Gambar 5.9.



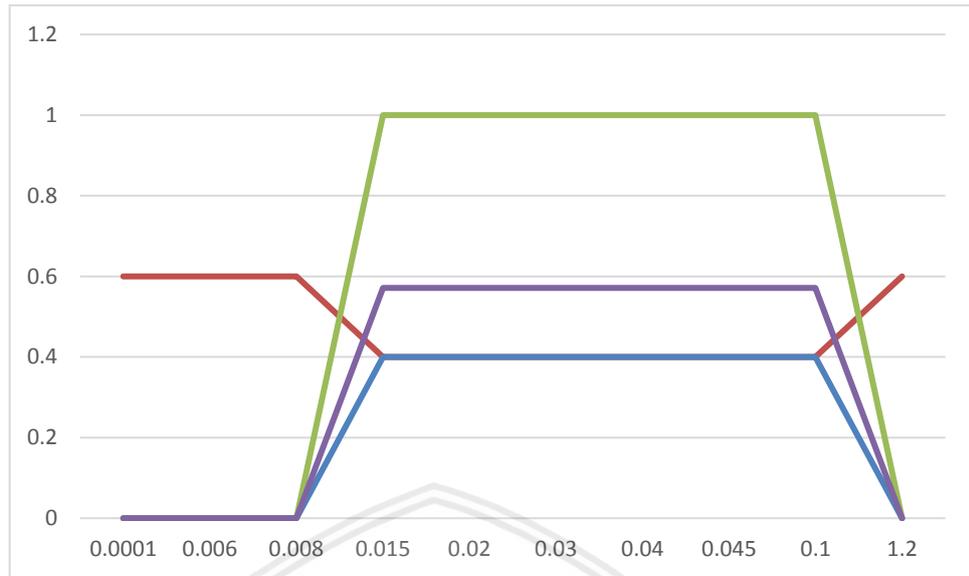
Gambar 5.1 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=50)



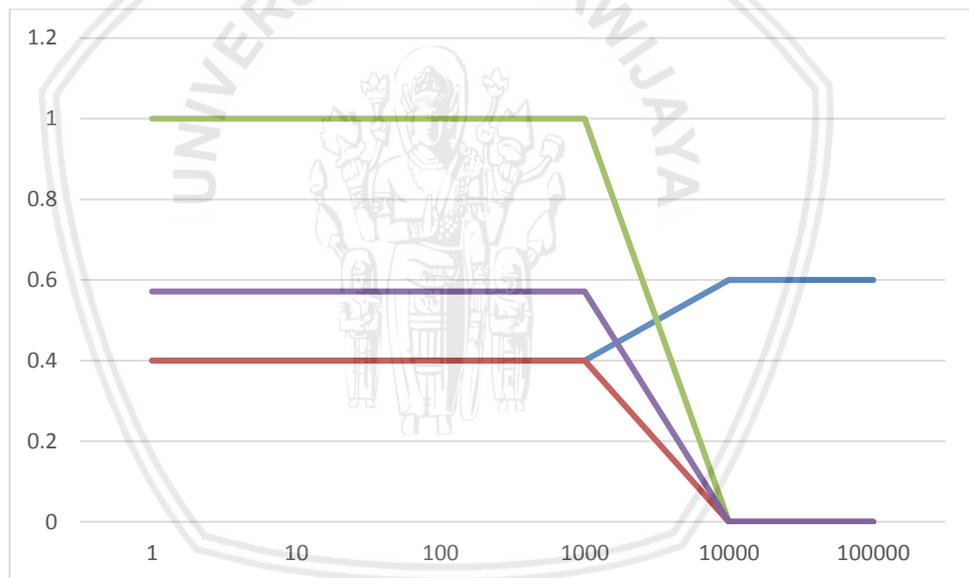
Gambar 5.2 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=100)



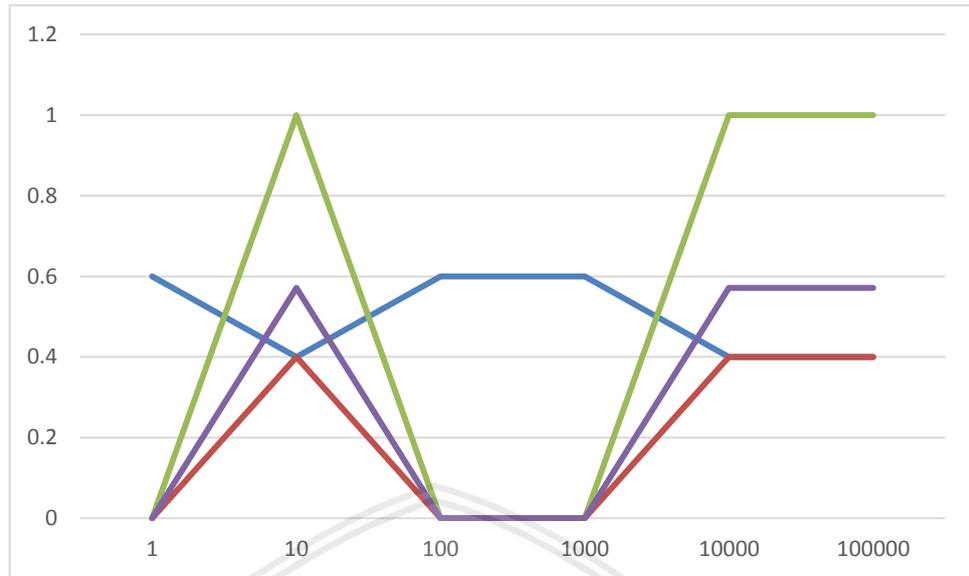
Gambar 5.3 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=150)



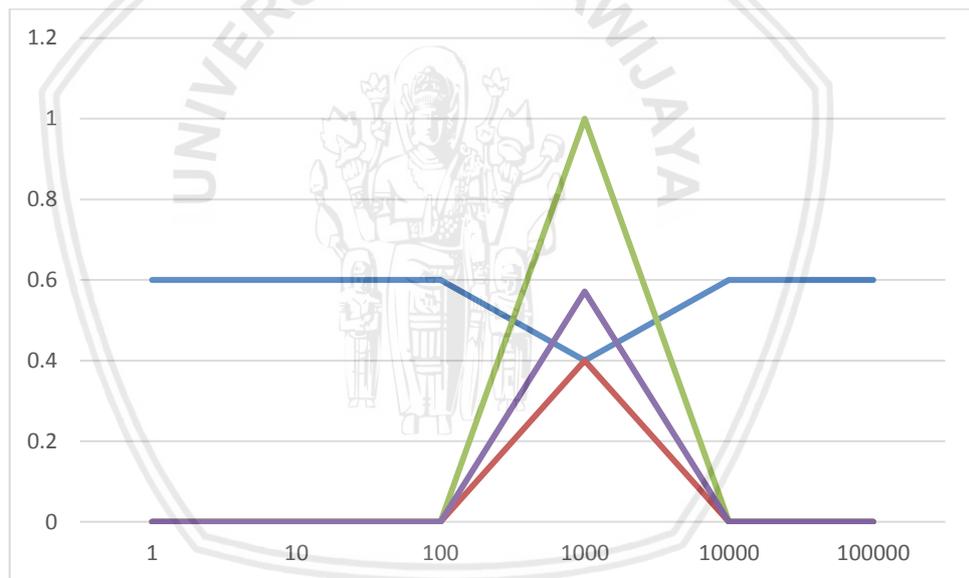
Gambar 5.4 Grafik Hasil Pengujian terhadap Parameter *Learning Rate* (Iterasi=200)



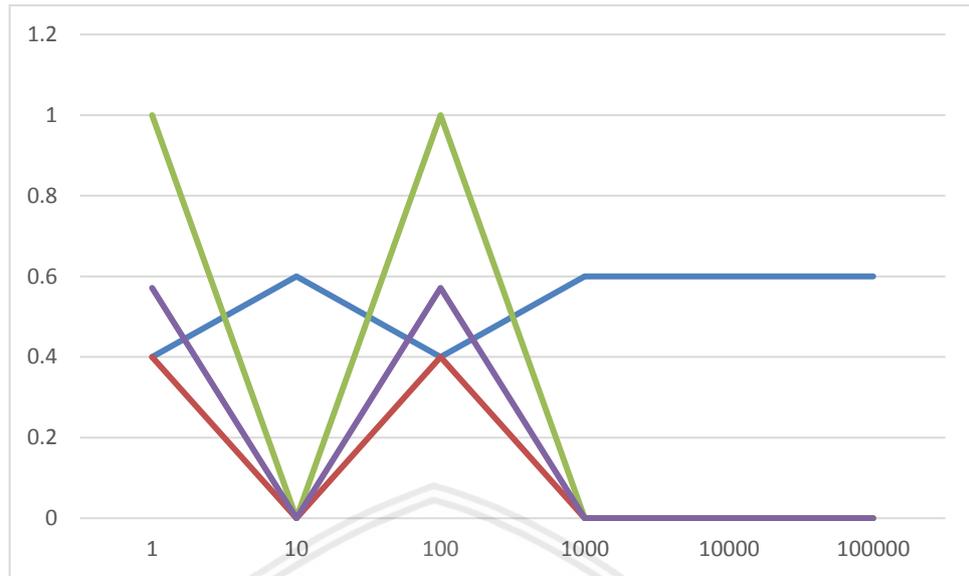
Gambar 5.5 Grafik Hasil Pengujian terhadap Parameter *Nilai C* (Iterasi=50)



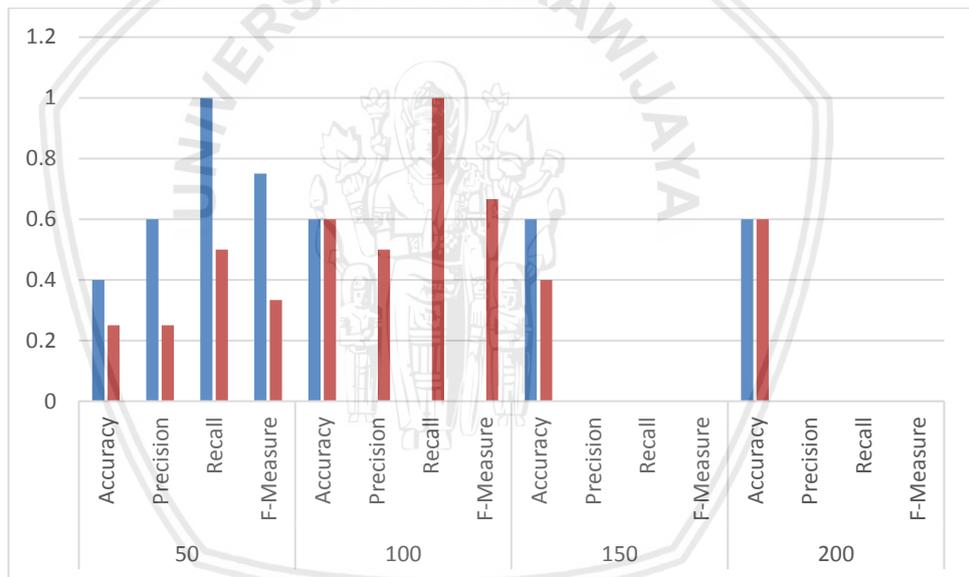
Gambar 5.6 Grafik Hasil Pengujian terhadap Parameter Nilai C (Iterasi=100)



Gambar 5.7 Grafik Hasil Pengujian terhadap Parameter Nilai C (Iterasi=150)



Gambar 5.8 Grafik Hasil Pengujian terhadap Parameter Nilai C (Iterasi=200)



Gambar 5.9 Grafik Hasil Pengujian Implementasi *Lexicon Based Features*

Berdasarkan parameter *learning rate*, nilai *c*, dan implementasi *lexicon based features* dengan jumlah iterasi maksimum yang sudah didapatkan dari pengujian, dilakukan percobaan yang memiliki hasil seperti yang ditunjukkan pada Tabel 5.4.

Tabel 5.4 Hasil Analisis Data Uji dengan Parameter Optimal

No	Tweet	Ground Truth/Aktual	Prediksi
1	@Arie_Kriting @BatikAirINA Barang saya tdk kembali pakai maskapai Batik air dan Malindo....group lion air adalah	-1	1
2	kamis lalu saya naik @citilink, harga sedang normal dan terbang enak	1	1
3	@SriwijayaAir Wah parah neh Pramugarinya masa ada ibu2 bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramugari memastikan penumpang duduk sesuai NO TEMPAT DUDUK yg tertera ditiket.SUMPAH KAPOK NAIK @SriwijayaAir @YLKI_ID @detikcom https://t.co/TFMXnnmLcI	-1	1

Tabel 5.4 Hasil Analisis Data Uji dengan Parameter Optimal (lanjutan)

4	I'm happy for @IndonesiaGaruda	1	1
5	harga jadi mahal @sriwijayaair	-1	1

```

1. @Arie_Kriting @BatikAirINA Barang saya tdk kembali pakai maskapai Batik air dan Malindo....group lion air adalah maskapai pencuri dan pemelihara sindikat p
embobol bagasi. Maskapai ini tdk ada tanggung jawabnya https://t.co/zFVSA1k0yA
2. kamis lalu saya naik @citilink, harga sedang normal dan terbang enak
3. @SriwijayaAir Wah parah neh Pramugarinya masa ada ibu2 bisa salah tempat duduk bukannya dari awal diarahkan biar gak salah duduk itu kan sdh tugasnya pramu
gari memastikan penumpang duduk sesuai NO TEMPAT DUDUK yg tertera ditiket.SUMPAH KAPOK NAIK @SriwijayaAir @YLKI_ID @detikcom https://t.co/TFMXnnmLcI
4. I'm happy for @IndonesiaGaruda
5. harga jadi mahal @sriwijayaair

+-----+-----+
| No Tweets | GT | Prediksi |
+-----+-----+
| 1         | -1 | 1         |
| 2         | 1  | 1         |
| 3         | -1 | 1         |
| 4         | 1  | 1         |
| 5         | -1 | 1         |
+-----+-----+

C:\Users\user\PycharmProjects\SkrripsiArsya1\

```

Gambar 5.10 Hasil Analisis Data Uji dengan Parameter Optimal

BAB 6 PENUTUP

6.1 Kesimpulan

Dari penelitian dan pengujian yang telah dilakukan, dihasilkan beberapa kesimpulan dalam Analisis Sentimen tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme *Support Vector Machine* (SVM). Beberapa hal yang dapat disimpulkan dari penelitian ini adalah sebagai berikut.

1. Metode klasifikasi *Support Vector Machine* dengan fitur *Lexicon Based* dapat digunakan dalam menganalisis sentimen opini maskapai penerbangan pada dokumen Twitter dengan optimal. Didapatkan nilai parameter *learning rate* (γ) sebesar 0.03 dan nilai C sebesar 10 sebagai nilai parameter paling optimal. Proses yang dilakukan untuk melakukan analisis sentimen pada penelitian ini dimulai dari tahapan *pre-processing*, pembobotan TF-IDF, pembobotan dengan fitur *Lexicon Based* serta normalisasi dengan *min-max*, kemudian perhitungan klasifikasi teks dengan metode *Support Vector Machine*. Hasil klasifikasi dibagi menjadi kelas positif dan kelas negatif.
2. Didapatkan tingkat akurasi paling baik sebesar 40%, *precision* sebesar 40%, *recall* sebesar 100%, dan *f-measure* sebesar 57,14%. Tingkat akurasi tersebut didapatkan dengan jumlah iterasi maksimum sebanyak 50 kali dengan diimplementasikannya fitur *lexicon based*. Dalam perhitungan akurasi digunakan parameter-parameter optimal yang telah disebutkan sebelumnya. Hasil klasifikasi dipengaruhi parameter yang optimal dan fitur yang digunakan.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan, masih ditemukan beberapa kekurangan. Sehingga diberikan saran untuk penelitian selanjutnya agar dapat memperbaiki. Saran yang dapat diberikan adalah sebagai berikut.

1. Data dapat diklasifikasikan menjadi tiga kelas, yaitu sentimen positif, negatif, dan netral.
2. Dapat diimplementasikan suatu metode tambahan seperti metode optimasi agar dapat lebih mudah dalam mengetahui sebuah teks berdasarkan makna tiap kata di dalam kalimat.
3. Dapat menambah dan mengurangi tahap pada *pre-processing*, misal: menggunakan formalisasi (sebagai contoh, seperti pujangga) karena banyak kata yang tidak baku atau tidak formal. Atau dapat mencoba tanpa melakukan tahap *stopword removal*, yang di mana saat ini sedang diperdebatkan penggunaannya terhadap analisis sentimen.
4. Dapat dilakukan inovasi pada sistem, misalnya selain sistem dapat menganalisis sentimen dari sebuah opini, sistem juga dapat merangkum opini dari keseluruhan opini yang ada.



DAFTAR REFERENSI

- Adiwijaya, I., 2006. Text Mining dan Knowledge Discovery. *Kolokium bersama komunitas datamining Indonesia & soft-computing Indonesia*, [online] pp.1–9. Available at: <<http://web.ipb.ac.id/~ir-lab/pdf/tm> (text summarization).pdf>.
- Afuan, L., 2013. Stemming Dokumen Teks Bahasa Indonesia. *Telematika*, 6(2), pp.34–40.
- Aribowo, A.S., Mining, O. and Produk, M., 2015. Arsitektur Aplikasi Twitter Opinion Mining Untuk Mengetahui Sentimen Publik Terhadap Merek. 2015(November), pp.14–20.
- Bhadane, C., Dalal, H. and Doshi, H., 2015. Sentiment analysis: Measuring opinions. *Procedia Computer Science*, [online] 45(C), pp.808–814. Available at: <<http://dx.doi.org/10.1016/j.procs.2015.03.159>>.
- Chalothorn, T. and Ellman, J., 2013. Sentiment Analysis: State of the Art. *Proceedings of the International Conference on Advances In Computer and Electronics Technology (ACET 2013)*, [online] pp.21–25. Available at: <http://dx.doi.org/10.3850/978-981-07-7227-7_05>.
- Ciravegna, F. and Harabagiu, S., 2003. Improving Sentiment Analysis in Twitter Using Multilingual Machine Translated Data. *IEEE Intelligent Systems*, [online] 18(1), pp.12–13. Available at: <<http://ieeexplore.ieee.org/document/1179188/>>.
- Fathan Hidayatullah, A., Sn, A., Teknik, J., Fakultas, I. and Industri, T., 2014. ISSN: 1979-2328 UPN "Veteran. *Seminar Nasional Informatika*, [online] 2014(semnasIF), pp.115–122. Available at: <<http://www.situs.com>>.
- Hamdan, H., Bellot, P. and Bechet, F., 2015. Lsislif: Feature extraction and label weighting for sentiment analysis in Twitter. *Proceedings of International Workshop on Semantic Evaluation (SemEval-2015)*, (SemEval), pp.568–573.
- Han, Q. and Guo, J., 2013. CodeX : Combining an SVM Classifier and Character N-gram Language Models for Sentiment Analysis on Twitter Text. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2(SemEval), pp.520–524.
- Informasi, F.T. and Susanti, N.D., 2016. Uji Perbandingan Akurasi Analisis Sentimen Pariwisata menggunakan Algoritma Support Vektor Machine dan Naive Bayes.
- Jain, A.P. and Katkar, V.D., 2015. Sentiments analysis of Twitter data using data mining. *2015 International Conference on Information Processing (ICIP)*, [online] pp.807–810. Available at: <<http://ieeexplore.ieee.org/document/7489492/>>.

- Junaedi, H., Budianto, H., Maryati, I. and Melani, Y., 2011. Data transformation pada data mining. *Prosiding Konferensi Nasional Inovasi dalam Desain dan Teknologi-IDEaTech*, 7, pp.93–99.
- Kaur, H. and Mangat, V., 2017. A Survey of Sentiment Analysis techniques. pp.921–925.
- Nanda, M.A., Seminar, K.B., Nandika, D. and Maddu, A., 2018. A comparison study of kernel functions in the support vector machine and its application for termite detection. *Information (Switzerland)*, 9(1).
- Studi, P., Sisteminformasi, M., Informasi, F.T., Kristen, U. and Wacana, S., 2017. Sentiment Analysis Model Based On Youtube Comment Using Support Vector Machine.
- Tiara, Sabariah, M.K. and 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. *2015 3rd International Conference on Information and Communication Technology, ICoICT 2015*, pp.386–390.
- Windasari, I.P., Uzzi, F.N. and Satoto, K.I., 2017. Sentiment Analysis on Twitter Posts : An analysis of Positive or Negative Opinion on GoJek. pp.266–269.
- Zhao, J., 2015. Pre-processing boosting twitter sentiment analysis? *Proceedings - 2015 IEEE International Conference on Smart City, SmartCity 2015, Held Jointly with 8th IEEE International Conference on Social Computing and Networking, SocialCom 2015, 5th IEEE International Conference on Sustainable Computing and Communic*, pp.748–753.
- Desai, M. and Mehta, M.A., 2017. Techniques for sentiment analysis of Twitter data: A comprehensive survey. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016, (March)*, pp.149–154.
- Rofiqoh, U., Perdana, R.S. and Fauzi, M.A., 2017. Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexion Based Feature. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, [online] 1(12), pp.1725–1732. Available at: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/628>>.
- Peng, W., 2011. Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization. s.l.:s.n.
- Darma, I.M.B.S., Perdana, R.S. and Indriati, 2018. Penerapan Sentimen Analisis Acara Televisi Pada Twitter Menggunakan Support Vector Machine dan Algoritma Genetika sebagai Metode Seleksi Fitur. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, [online] 2(3), pp.998–1007. Available at: <<http://j-ptiik.ub.ac.id>>.
- Turchi, M., 2013. Improving sentiment analysis in twitter using multilingual machine translated data Improving Sentiment Analysis in Twitter Using Multilingual Machine Translated Data. (January).