TUGAS AKHIR - SS141501

KLASIFIKASI EMOSI UNTUK TEKS BERBAHASA INDONESIA PADA PENGGUNA TWITTER MENGENAI PRESIDEN JOKO WIDODO

FAZLUR RAHMAN
NRP 062114 4000 0072

Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

TUGAS AKHIR – SS141501

KLASIFIKASI EMOSI UNTUK TEKS BERBAHASA INDONESIA PADA PENGGUNA TWITTER MENGENAI PRESIDEN JOKO WIDODO

FAZLUR RAHMAN
NRP 062114 4000 0072

Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

FINAL PROJECT – SS 141501

EMOTIONS CLASSIFICATION FOR INDONESIAN TEXT ON TWITTER USER ABOUT PRESIDENT JOKO WIDODO

FAZLUR RAHMAN
SN 062114 4000 0072

Supervisor
Dr. Kartika Fithriasari, M.Si.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

LEMBAR PENGESAHAN

KLASIFIKASI EMOSI UNTUK TEKS BERBAHASA INDONESIA PADA PENGGUNA *TWITTER* MENGENAI PRESIDEN JOKO WIDODO

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh :
Fazlur Rahman
NRP. 062114 4000 0072

Disetujui oleh Pembimbing :
Dr. Kartika Fithriasari, M.Si. ( )
NIP. 19691212 199303 2 002

Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, AGUSTUS 2018

(*Halaman ini sengaja dikosongkan*)

KLASIFIKASI EMOSI UNTUK TEKS BERBAHASA INDONESIA PADA PENGGUNA TWITTER MENGENAI PRESIDEN JOKO WIDODO

Nama Mahasiswa	**: Fazlur Rahman**
NRP	**: 062114 4000 0072**
Departemen	**: Statistika**
Dosen Pembimbing	**: Dr. Kartika Fithriasari, M.Si.**

Abstrak

Presiden Jokowi jelas selalu menjadi sorotan di mata masyarakat. Beliau bertugas memenuhi harapan rakyatnya, dan akan baik kinerjanya apabila mendapat dukungan penuh pula dari rakyatnya. Hal tersebut membuat Presiden Jokowi kerap kali diperbincangkan oleh masyarakat. Pada perkembangan teknologi seperti saat ini, masyarakat dapat menyuarakan pendapat dan emosinya melalui teks pada media social seperti Twitter. Twitter adalah sebuah jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan dalam bentuk teks. Namun, emosi yang diungkapkan oleh masyarakat sangat beragam jenisnya. Mengetahui tanggapan masyarakat berdasarkan emosinya akan membuat proses evaluasi semakin efek-tif. Oleh karena itu, diperlukannya sebuah penelitian untuk mengetahui opini dan emosi dari masyarakat. Tanggapan publik mengenai Presiden Jokowi didapat dari Application Programming Interface (API). Sebelum dilakukan klasifikasi teks, akan dilakukan praproses teks. Praproses teks yang digunakan adalah case folding, stopwords, dan tokenizing. Sedangkan pada analisis klasifikasi data teks tersebut digunakan metode Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN). Klasifikasi menggunakan SVM menghasilkan akurasi prediksi sebesar 95.2%. Sedangkan KNN menghasilkan akurasi 87.2%.

Kata Kunci : Jokowi, , K-Nearest Neighbor, Klasifikasi Emosi, Support Vector Machine, Twitter

(*Halaman ini sengaja dikosongkan*)

EMOTIONS CLASSIFICATION FOR INDONESIAN TEXT ON TWITTER USER ABOUT PRESIDENT JOKO WIDODO

Name : **Fazlur Rahman**
SN : **062114 4000 0072**
Department : **Statistics**
Supervisor : **Dr. Kartika Fithriasari, M.Si.**

Abstract

President Jokowi obviously always be the spotlight in the eyes of the public. He is in charge of fulfilling the expectations of Indonesian people, and will perform well if he gets full support from the people. This makes President Jokowi often being the topic of people's talk about. Today, people can say their opinions and emotions through text on social media such as Twitter. Twitter is a microblog social network that allows users to send and read text messages. However, the emotions expressed by Twitter user are very diverse. Knowing people's responses based on their emotions will make President Jokowi knows what is must being evaluated more effectively. Therefore, are being needed a research to find opinions and emotions from the Twitter user. Public response about President Jokowi is obtained from the Application Programming Interface (API). Before classification process, preprocess text will be performed. The text praprocess tha is used are case folding, stopwords, and tokenizing. While in the text classification analysis is used Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) method. Predictive accuracy of Classification using SVM yields 95.2%. While for method KNN yield accuracy is 87.2%

Keyword : Emotion Classification, Jokowi, K-Nearest Neighbor, Support Vector Machine, Twitter

(*Halaman ini sengaja dikosongkan*)

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah, puji syukur penulis panjatkan atas rahmat, taufik, dan hidayah yang diberikan oleh Allah SWT sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **"Klasifikasi Emosi Untuk Teks Berbahasa Indonesia Pada Pengguna Twitter Mengenai Presiden Joko Widodo"** dengan lancar.

Penulis menyadari bahwa Tugas Akhir ini dapat terselesaikan tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Bapak, Ibu dan keluarga penulis yang selalu memberikan doa, kasih sayang, dukungan serta bimbingannya.
2. Dr. Kartika Fithriasari, M.Si. selaku dosen pembimbing Tugas Akhir dan yang telah meluangkan waktu dan dengan sangat sabar memberikan bimbingan, saran, dukungan serta motivasi selama penyusunan Tugas Akhir.
3. Dra. Wiwiek Setya Winahju, M.S dan Pratnya Paramitha Oktaviana, S.Si., M.Si. selaku dosen penguji yang telah ba-nyak memberi masukan kepada penulis.
4. Dr. Suhartono selaku Kepala Departemen Statistika Institut Teknologi Sepuluh Nopember dan Dr. Sutikno, M.Si. selaku Kepala Program Studi Sarjana yang telah memberikan fasilitas, sarana, dan prasarana.
5. Dra. Madu Ratna, M.Si. selaku dosen wali yang telah banyak memberikan saran dan arahan dalam proses belajar di Departemen Statistika.
6. Mas Taufik Kurniawan, S.Si yang telah memberi ilmu dan bimbingan mengenai segala hambatan yang penulis temui pada Tugas Akhir ini.
7. Teman-teman penulis di kos Garuda, yang selalu menemani dan memotivasi disaat penulis merasa malas pada proses penulisan laporan Tugas Akhir.

8. Sahabat-sahabat penulis seperantauan dari Lumajang yang selama perkuliahan selalu berkegiatan dan berdiskusi bersama sehingga meningkatkan *soft skill* penulis.
9. Teman-teman Statistika ITS angkatan 2014, Respect, yang selalu memberikan dukungan kepada penulis selama ini.
10. Semua pihak yang turut membantu dalam pelaksanaan Tugas Akhir yang tidak bisa penulis sebutkan satu persatu.

Besar harapan penulis untuk mendapatkan kritik dan saran yang membangun sehingga Tugas Akhir ini dapat memberikan manfaat bagi semua pihak yang terkait.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

(*Halaman ini sengaja dikosongkan*)

DAFTAR TABEL

Halaman

(*Halaman ini sengaja dikosongkan*)

DAFTAR LAMPIRAN

Halaman

(*Halaman ini sengaja dikosongkan)*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dari masa ke masa sangat pesat dan mempunyai peran yang sangat penting dalam kehidupan masyarakat baik individu maupun kelompok organisasi. Teknologi informasi memegang peranan diberbagai sendi kehidupan manusia karena dapat menghubungkan dan manyajikan berbagai informasi melalui *web*. *Web* atau *website* merupakan halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses seluruh dunia selama terkoneksi dengan jaringan internet. *Web* memuat dua jenis tekstual yaitu fakta dan opini (Buntoro, 2014). Fakta merupakan suatu pernyataan obyektif mengenai entitas dan kejadian di dunia, sedang opini merupakan pernyataan subyektif yang menggambarkan sentimen atau persepsi seseorang mengenai entitas atau kejadian di dunia. *Web* telah menyediakan berbagai fakta dan opini dari banyak hal melalui blog pribadi, situs jejaring sosial, dan *microblog* lainnya sehingga jika seseorang atau suatu kelompok organisasi maupun perusahaan yang ingin memperoleh opini publik mengenai suatu produk atau layanan, maka penggunaan data yang terdapat dalam *web* dapat menjadi alternatif yang efisien selain menggunakan survei konvensional.

Twitter adalah salah satu *microblog* yang memiliki banyak pengguna didunia. Pengguna *twitter* di Indonesia menempati peringkat 5 terbesar di dunia dibawah USA, Brazil, Jepang, dan Inggris yaitu mencapai angka 19,5 juta pengguna *twitter* dari total 300 juta pengguna global (Kemenkominfo, 2016). Semenjak tahun 2010 hingga kuartal ketiga tahun 2016, pengguna *twitter* mengalami pertumbuhan yang signifikan hingga mencapai 313 juta akun. Pengguna twitter dapat mengemukakan pendapatnya terhadap suatu produk atau suatu program melalui *tweet*. *Tweet* pada setiap pengguna *twitter* dapat berpengaruh dalam pembentukan citra suatu produk atau program karena semakin banyak suatu topik tertentu diulas dalam *tweet* pengguna maka topik tersebut dapat

menjadi trending topic di *twitter*. *Twitter* telah menyediakan *Application Programming Interface* (API) yaitu sekumpulan fungsi atau protokol yang disediakan dalam rangka mengembangkan sebuah aplikasi (Blanchette, 2008). *Twitter* API memungkinkan pengguna untuk mengakses dan mendapatkan informasi mengenai *tweet*, profil pengguna, data *follower*, dan lainnya. Hal tersebut menjadikan *Twitter* sebagai *microblog* yang banyak diminati perusahaan, organisasi, maupun individu dalam mendapatkan opini publik mengenai suatu topik tertentu.

Kerap kali terjadi kontorversi masyarakat terhadap sosok pemimpin bangsa. Seorang pemimpin bangsa di Indonesia yang dinamakan Presiden jelas selalu menjadi sorotan di mata masyarakat. Presiden Republik Indonesia adalah kepala negara sekaligus kepala pemerintahan Indonesia. Sebagai kepala negara, Presiden merupakan simbol resmi negara Indonesia di dunia. Menurut Undang-Undang Dasar (UUD) 1945 ayat 22, Presiden dibantu oleh wakil presiden beserta menteri-menteri dalam kabinet, memegang kekuasaan eksekutif untuk melaksanakan tugas-tugas pemerintah sehari-hari. Presiden (dan Wakil Presiden) menjabat selama 5 tahun, dan sesu-dahnya dapat dipilih kembali dalam jabatan yang sama untuk satu kali masa jabatan. Presiden Republik Indonesia telah berganti seba-nyak tujuh kali, mulai dari bapak bangsa Ir. Soekarno dan sekarang adalah Joko Widodo atau yang kerap dipanggil Jokowi (Republik Indonesia, 1945).

Di masa dengan media bersuara telah bebas seperti sekarang ini, tidak jarang orang melontarkan pendapat dan emosinya mengenai Presiden Jokowi. Seorang pemimpin memang harus selalu mengevaluasi diri. Karena Presiden bertugas memenuhi harapan rakyatnya, dan akan baik kinerjanya apabila mendapat dukungan penuh pula dari rakyatnya. Mengetahui emosi dari masyarakat penting dalam evaluasi kinerja dari seorang tokoh. Mengenali emosi dapat meningkatkan kualitas interaksi antar manusia (Lopatovska, 2010). Jika sebagai presiden semua yang dilakukan tidak mendapat respon yang baik oleh masyarakat, maka apa yang dilakukannya selama ini akan menjadi tak bermakna. Mengetahui emosi dari ma-

syarakatnya dapat membantu kinerjanya. Disaat masyarakat tidak suka dengan tindakan yang dilakukan oleh Presiden Jokowi, maka beliau dapat memperbaiki cara kerjanya sehingga masyarakat akan memberikan pendapat yang positif terhadap beliau. Dan seiring berjalannya waktu, mayoritas masyarakat akan selalu mendukung program-program yang akan dijalankannya. Mengetahui opini dan emosi masyarakat secara keseluruhan memang tidak bisa dilakukan. Namun, pada perkembangan teknologi seperti sekarang ini, masyarakat dapat menyuarakan pendapat dan emosinya melalui teks pada media sosial. Teks tidak hanya memuat informasi, teks juga dapat mengekspresikan emosi (Hirat, 2015). Namun, emosi yang diungkapkan oleh masyarakat sangat beragam jenisnya. Mengetahui tanggapan masyarakat berdasarkan emosinya akan membuat proses evaluasi semakin efektif. Oleh karena itu, diperlukannya sebuah penelitian untuk mengetahui opini dan emosi masyarakat dan mengklasifikasikannya berdasarkan emosi melalui media sosial.

Sebelum melakukan klasifikasi emosi, diperlukan metode praproses teks dengan metode *text mining* untuk mengolah data teks agar siap untuk dianalisis. Praproses teks meliputi *case folding*, *stopwords*, dan *tokenizing*. *Case folding* merupakan praproses untuk merubah semua teks menjadi huruf kecil. *Tokenizing* adalah proses memecah teks yang berasal dari kalimat menjadi kata per kata. *Stopwords* merupakan kosakata yang tidak termasuk kata unik atau ciri dari sebuah dokumen sehingga perlu dihilangkan. Terdapat banyak metode klasifikasi dalam ilmu statistika yang digunakan untuk analisis sentimen namun metode yang sering digunakan dalam klasifikasi teks adalah metode *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN). Metode SVM digunakan karena metode ini sangat cepat dan efektif pada klasifikasi data teks (Feldman & Sanger, 2007). Sedangkan kelebihan KNN adalah metode tersebut tangguh terhadap training data yang besar dan memiliki banyak noise (Larose, 2005):

Penelitian yang pernah dilakukan mengenai kalsifikasi teks *adalah Twitter Used by Indonesian President: An Sentiment Ana-*

lysis of Timeline. Penelitian tersebut membahas hasil analisis sentimen pada akun twiter milik Presiden. Akurasi yang didapat dari penelitian tersebut sebesar 79,42%. Namun masih terdapat kekurangan dalam memisahkan data *text* yang tidak mengandung sentimen (Aliandu, 2013). Arifin dan Purnama (2012) melakukan penelitian yang berjudul *Classification of Emotions in Indonesian Texts Using KNN Method*. Penelitian tersebut menggunakan dokumen teks dari website berita yang diambil secara manual dan mendapatkan hasil akurasi klasifikasi 58%. Selain itu, penelitian oleh Bata (2015) mengenai leksikon emosi berbahasa Indonesia yang memuat hasil awal pengembangan leksikon emosi untuk bahasa Indonesia. Lalu penelitian lain pernah dilakukan Kurniawan (2017) tentang Klasifikasi Media Mainstream Menggunakan *Naïve Bayes Classifier* dan *Support Vector Machine*. Penelitian tersebut menghasilkan bahwa akurasi dengan *Support Vector Machine* lebih baik secara keseluruhan.

Pada penelitian ini, struktur data yang digunakan terdiri dari variabel independen yaitu kata dasar tweet yang telah dilakukan praproses *text* dan variabel dependen yaitu klasifikasi emosi tweet. Penelitian ini bertujuan melakukan klasifikasi emosi mengenai tanggapan masyarakat terhadap Presiden Jokowi berbasis *Text Mining* data *twitter*. Melalui penelitian ini diharapkan dapat memberikan informasi maupun saran kepada pihak terkait dan pemerintah mengenai tanggapan berdasarkan emosi masyarakat mengenai pemimpin bangsa.

1.2 Rumusan Masalah

Mengetahui emosi dari masyarakat pengguna *Twitter* terhadap Presiden Joko Widodo merupakan hal yang tidak bisa dikesampingkan. Namun, emosi yang diungkapkan oleh masyarakat sangat beragam jenisnya. Mengetahui tanggapan masyarakat berdasarkan emosinya akan membuat proses evaluasi semakin efektif. Oleh karena itu, diperlukannya sebuah penelitian untuk mengetahui opini dan emosi masyarakat dan mengklasifikasikannya berdasarkan emosi melalui media sosial. SVM adalah metode klasifi-

kasi dengan mencari nilai pemisah antar kategori yang optimum atau *optimum separating hyperplane*. Metode SVM mempunyai akurasi yang tinggi untuk klasifikasi data teks. Sedangakan KNN suatu metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari label *class* pada KNN. Tujuan dari algoritma KNN adalah mengklasifikasikan objek baru berdasarkan atribut dan training data. Kedua metode tersebut akan dihitung dengan harapan menghasilkan error yang kecil. Berdasarkan penjelasan tersebut, maka permasalahan utama yang akan dibahas dalam penelitian ini adalah berapa tingkat ketepatan klasifikasi emosi pengguna *twitter* terhadap Presiden Joko Widodo menggunakan metode *Support Vector Machine* dan K-*Nearest Neighbor*? Serta apa kata yang sering muncul berdasarkan masing-masing emosi?

1.3 Tujuan

Berdasarkan rumusan masalah tersebut, maka penelitian ini dibuat dengan tujuan sebagai berikut.

1. Mendapatkan hasil klasifikasi emosi pengguna twitter terhadap presiden Joko Widodo dengan menggunakan metode *Support Vector Machine* (SVM)
2. Mendapatkan hasil klasifikasi emosi pengguna twitter terhadap presiden Joko Widodo dengan menggunakan metode K-*Nearest Neighbor* (KNN)
3. Mendapatkan kata-kata yang sering muncul pada setiap klasifikasi menggunakan visualisasi *word cloud*.

1.4 Manfaat

Hasil penelitian ini diharapkan dapat bermanfaat dalam beberapa aspek sebagai berikut.

1. Menunjukkan hasil klasifikasi emosi masyarakat pengguna twitter mengenai Presiden Republik Indonesia untuk kepada Presiden Jokowi, pihak pemerintah, dan pihak lain yang terkait evaluasi kinerja ke depannya.

2. Memberikan tambahan informasi kepada publik melalui emosi pengguna *twitter* terhadap presiden Jokowi melalui hasil klasifikasi.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Penelitian menggunakan studi kasus dengan kata kunci yang berkaitan dengan Presiden Joko Widodo di *Twitter*.
2. Klasifikasi sentimen data awal ditentukan secara subyektif peneliti dengan refrensi pada leksikon emosi

BAB II

TINJAUAN PUSTAKA

Bab ini membahas mengenai *text mining*, *leksikon* emosi, praproses teks, *Term Frequency Inverse Document Frequency*, *Support Vector Machine* (SVM), *K-Nearest Neighbors*, evaluasi kebaikan model, visualisasi *wordcloud*, serta *Twitter*.

2.1 *Text Mining*

Text Mining adalah penggalian data untuk menyelesaikan masalah kebutuhan informasi dengan menerapkan teknik data mining, machine learning, natural language processing, pencarian informasi, dan manajemen pengetahuan. *Text mining* melibatkan praproses dokumen seperti kategorisasi teks, ekstraksi informasi, dan ekstraksi kata. Metode ini digunakan untuk mengekstrak informasi dari sumber data melalui identifikasi dan eksplorasi pola yang menarik (Feldman & Sanger, 2007).

Text Mining merupakan teknik yang digunakan untuk menangani permasalahan klasifikasi, clustering, information extraction dan information retrival (Berry, 2010). Pada dasarnya proses kerja dari *Text Mining* banyak mengapdopsi dari penelitian data mining namun yang menjadi perbedaan adalah pola yang digunakan oleh Text Mining diambil dari sekumpulan Bahasa alami yang tidak terstruktur sedangkan dalam data mining pola yang diambil dari database yang terstruktur. Tahap-tahap *Text Mining* secara umum adalah praproses teks dan *feature selection*

2.2 Leksikon Emosi

Emosi dapat digambarkan sebagai keadaan yang pada umumnya disebabkan oleh suatu kejadian penting sebuah subyek yang meliputi keadaan mental sadar yang dinyatakan dengan kemampuan mengenali, kualitas perasaan dan diarah untuk beberapa subyek, gangguan jasmani pada beberapa organ tubuh, pengenalan emosi pada wajah, suara dan isyarat tubuh, kesiapan untuk melakukan tindakan tertentu. Karenanya emosi dalam sosiobiologi ada-

lah kecenderungan mental, keadaan, proses dan model komputasi harus spesifikasi semirip mungkin (Oatley, 1996). Sejumlah penelitian tentang emosi manusia telah dilakukan sehingga ada kesepakatan tentang emosi dasar (Power, 1997)

1. Takut sebagai ancaman fisik atau sosial untuk diri sendiri
2. Marah sebagai ganjalan atau frustasi dari peran atau tujuan yang di rasakan orang lain
3. Jijik menggambarkan penghapusan atau jarak dari seseorang, obyek, atau menolak ide untuk diri sendiri dan menghargai peran dan tujuan
4. Sedih digambarkan sebagai kegagalan atau kerugian tentang peran dan tujuan
5. Senang digambarkan sebagai berhasil atau bergerak menuju selesainya peran yang bernilai atau tujuan.

Namun dalam penelitian kali ini, peneliti hanya menggunakan 3 emosi yakni marah, sedih, dan senang.

Dalam linguistik, leksikon adalah koleksi leksem pada suatu bahasa. Kajian terhadap leksikon mencakup apa yang dimaksud dengan kata, strukturisasi kosakata, penggunaan dan penyimpanan kata, pembelajaran kata, sejarah dan evolusi kata (etimologi), hubungan antar kata, serta proses pembentukan kata pada suatu bahasa. Dalam penggunaan sehari-hari, leksikon dianggap sebagai sinonim kamus atau kosakata (Pusat Bahasa, 2008). Bisa diartikan bahwa leksikon emsoi adalah kamus kosakata yang digolongkan terhadap emosi manusia

2.3 Metode Analisis

Metode klasifikasi yang akan digunakan adalah SVM dan KNN. Namun diperlukan praproses teks untuk mengolah data teks agar siap untuk diklasifikasikan. Pada proses stemmer akan menggunakan algoritma *Confix-Stripping Stemmer* yang merupakan al-

goritma untuk *stemming* berdasarkan aturan Bahasa Indonesia. Setelah itu dilakukan pembobotan untuk merubah data berbentuk kata (tweet) menjadi bentuk numerik agar dapat diklasifikasikan dan dihitung ketepatan akurasinya.

2.3.1 Praproses Teks

Praproses teks merupakan tahapan awal dalam pengolahan teks yang digunakan untuk pengubahan bentuk dokumen menjadi data yang terstruktur sesuai kebutuhannya agar dapat diolah lebih lanjut dalam proses *text mining*. Tahapan praproses teks dalam klasifikasi bertujuan untuk meningkatkan akurasi klasifikasi data. Tahapan dalam praproses teks adalah sebagai berikut.

a *Cleaning*, yaitu proses menghapus URL, simbol *retweet* (RT), simbol *hashtag* (#), serta symbol-simbol lain yang tidak diperlukan (Mujilahwati, 2016)
b *Case Folding*, merupakan proses untuk mengubah semua karakter teks menjadi huruf kecil serta menghilangkan tanda baca dan angka. (Weiss, 2010).
c *Stopwords*, merupakan kosakata yang bukan termasuk kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat (Dragut, 2009).
d *Stemming*, yakni proses untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* atau kombinasi dari awalan dan akhiran.
e. *Tokenizing*, merupakan proses memecah yang semula kalimat menjadi kata-kata

2.3.2 *Term Frequency Inverse Document Frequency*

Term Frequency Inverse Document Frequency (TFIDF) merupakan metode pembobotan yang dilakukan untuk ekstraksi data teks. Tujuan TFIDF adalah menemukan jumlah kata yang di-

ketahui (TF) setelah dikalikan dengan frekuensi *tweet* dimana suatu kata tersebut muncul (IDF). Metode TFIDF dilakukan dengan menghitung bobot dengan cara integrasi antara *term frequency* (TF) dan *inverse document frequency* (IDF). Berikut merupakan rumus untuk menemukan pembobot dengan TFIDF.

$$TFIDF_{ij} = TF_{ij} \times IDF_j \tag{2.1}$$

$$IDF_i = \log\left(\frac{N}{DF_i}\right) \tag{2.2}$$

dimana $_i$ adalah kata ke-i dari urutan kata yang muncul, $_j$ adalah *tweet* ke-*j*, $TFIDF_{ij}$ adalah bobot dari kata *i* pada artikel ke *j*, *N* merupakan jumlah seluruh *tweet*, *TF* adalah frekuensi kata *i* pada *tweet j*, dan *DF* adalah jumlah *tweet* yang mengandung kata *i*. TFIDF bertjuan merubah data teks menjadi bentuk numerik. Algoritma TFIDF berjalan dengan langkah sebagai berikut.

1. Memecah semua *tweet* menjadi kata untuk mengetahui semua kata yang muncul
2. Menghitung TF, yakni jumlah kata dalam setiap *tweet*
3. Menghitung DF atau banyaknya *tweet* dari suatu kata
4. Menghitung IDF dengan menggunakan rumus 2.2 untuk setiap kata yang muncul
5. Menghitung TFIDF setiap kata yang muncul per *tweet* dengan rumus 2.1

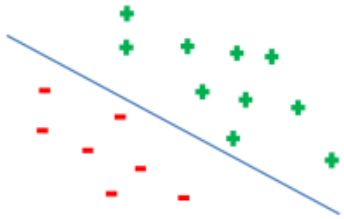
2.3.3 *Support Vector Machine*

Support Vector Machine (SVM) adalah metode yang mempelajari area yang memisahkan antar kategori dalam sebuah observasi. Dalam terminology SVM, kita membahas jarak atau margin antar kategori. Setiap kategori memiliki observasi dimana nilai variabel targetnya sama (Williams, 2011). SVM juga dikenal sebagai

sistem pembelajaran yang menggunakan hipotesis fungsi linear dalam ruang dimensi tinggi dan dilatih dengan algoritma berdasarkan toeri optimasi dengan menerapkan *learning bias* yang berasal dari teori statistik. Tujuan dari metode ini adalah membangun pemisah optimum yang disebut OSH (*Optimal Separating Hyperplane*) sehingga dapat digunakan untuk klasifikasi.

a. *SVM* pada *Linearly Separable Data*

SVM pada *linearly separable data* adalah penerapan metode SVM pada data yang dapat dipisahkan secara linier. Misalkan $x_i = \{x_i, x_{i+1}, \dots, x_n\}$ adalah dataset dan $y_i = \{+1, -1\}$ adalah label kategori untuk dataset. Penggambaran *linearly separable data* dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 Penggambaran *Linearly Separable Data*
(Sumber : google.com)

Tool pertama untuk mengimplementasi SVM adalah *hyperplane*. *Hyperplane* merupakan generalisasi dari *line* (garis lurus) pada ruang 2 dimensi atau *plane* (bidang datar) pada ruang 3 dimensi. Pada ruang *cartesius* 2 dimensi, persamaan garis $ax + by + c = 0$ sudah sangat familiar. Untuk membuat persamaan yang lebih umum dari persamaan tersebut yang dapat mencakup ruang berdimensi banyak, pertama-tama harus mengubah notasi variabel dan konstanta dari persamaan garis tersebut. x menjadi x_1,

y menjadi x_2, a menjadi w_1, b menjadi w_2, dan c menjadi b atau biasa disebut bias. Sehingga persamaannya menjadi

$$w_1 x_1 + w_2 x_2 + b = 0 \tag{2.3}$$

jika data yang diperoleh berdimensi $k > 1$, maka 2.3 menjadi

$$w_1 x_1 + \cdots + w_k x_k + b = 0 \tag{2.4}$$

atau

$$\sum_{k=1}^{K} w_k x_k + b = 0 \tag{2.5}$$

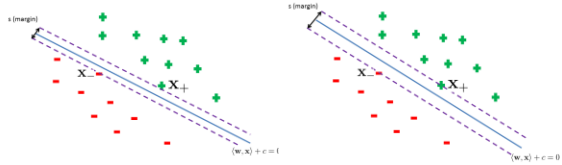
Cara lain menuliskan persamaan 2.5 adalah dengan notasi vektor dengan menyatakan variabel-variabel w_1 dan x_1 dengan vektor $\mathbf{w} = [w_1, \ldots, w_k]^{\mathrm{T}}$ dan $\mathbf{x} = [x_1, \ldots, x_d]^{\mathrm{T}}$.

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{2.6}$$

atau

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0 \tag{2.7}$$

Pada SVM, *margin* merupakan jarak antara titik-titk positif dan negatif terdekat di sekitar *hyperplane*. Pada Gambar 2.2, *margin* diilustrasikan dengan jarak antara 2 garis ungu. Titik sampel yang tepat berada di garis ungu disebut sebagai *support vektors.*



Gambar 2.2 Ilustrasi *Non-maximum Margin* (kiri) dan *Maximum Margin* (kanan) (Sumber : google.com)

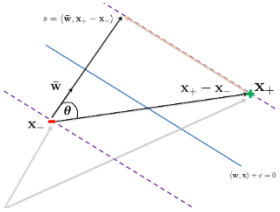
Sebelum penjelasan algoritma, *margin* (s) harus dinyatakan secara matematis terlebih dahulu. Misalkan terdapat sebuah sampel x dan *hyperplane* $y(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b$. *Decision rule* dari SVM didefinisikan sebagai berikut.

$$f(x) = \begin{cases} +1, \text{jika } y(x) \geq 1 \\ -1, \text{jika } y(x) \leq -1 \end{cases} \tag{2.8}$$

dan persamaan 2.8 tersebut dapat disederhanakan menjadi

$$y(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1 \tag{2.9}$$

dimana $y = 1$ apabila $\mathbf{x} \coloneqq \mathbf{x}_+$ (positif) dan $y = -1$ apabila $\mathbf{x} \coloneqq \mathbf{x}_-$ (negatif). Agar lebih jelas, maka dapat dilihat pada gambar 2.2 *Maximum margin* (kanan). Gambar 2.3 berikut adalah hasil *zoom in* dari gambar 2.2 dengan *Maximum margin* agar lebih terlihat 2 titik positif dan negatif yang tepat berada di garis ungu, yakni yang disebut *support vektor*.



Gambar 2.3 *Zoom in* Ilustrasi *Maximum Margin*
(Sumber : google.com)

Misalkan $\mathbf{x}_+ - \mathbf{x}_-$ merupakan vektor yang menghubungkan titik $\mathbf{x}_+$ ke titik $\mathbf{x}_-$, dan $\bar{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$ merupakan vektor satuan dengan $\|\bar{\mathbf{w}}\| = 1$, yang searah dengan vektor normal $\mathbf{w}$. *Margin* merupakan proyeksi orthogonal dari vektor $\mathbf{x}_+ - \mathbf{x}_-$ ke vektor $\bar{\mathbf{w}}$

$$s = \langle \overline{\mathbf{w}}, \mathbf{x}_+ - \mathbf{x}_- \rangle \tag{2.10}$$

Dibutuhkan penyederhanaan dari persamaan 2.10 agar lebih mudah dan praktis untuk memaksimumkan margin nantinya. Mulanya, persamaan 2.10 akan dijabarkan terlebih dahulu.

$$s = \langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_+ - \mathbf{x}_- \rangle = \langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_+ \rangle - \langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_- \rangle \tag{2.11}$$

bentuk $\langle \mathbf{w}, \mathbf{x}_+ \rangle$ dan $\langle \mathbf{w}, \mathbf{x}_- \rangle$ dapat dijabarkan lagi menjadi

$$\langle \mathbf{w}, \mathbf{x}_+ \rangle + b = 1, \text{menjadi } \langle \mathbf{w}, \mathbf{x}_+ \rangle = 1 - b \tag{2.12}$$

dan

$$-\langle \mathbf{w}, \mathbf{x}_- \rangle + b = 1, \text{menjadi } \langle \mathbf{w}, \mathbf{x}_- \rangle = -(1 + b) \tag{2.13}$$

dengan mensubsitusikan persamaan 2.12 dan 2.13, maka persamaan 2.11 akan menjadi

$$s = \frac{1 - b}{\|\mathbf{w}\|} + \frac{1 + \mathrm{b}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{2.14}$$

Untuk memaksimumkan *margin*, maka dibutuhkan algoritma yang dapat memaksimalkan persamaan 2.14.

$$\max \frac{2}{\|\mathbf{w}\|} = \min \frac{\|\mathbf{w}\|}{\mathbf{2}} = \min \frac{1}{2} \|\mathbf{w}\|^{\mathbf{2}} \tag{2.15}$$

dengan kata lain, memaksimumkan *margin s* ekuivalen dengan meminimumkan vektor normal $\mathbf{w}$ dari *hyperplane* $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$. Algoritma SVM menyelesaikan problem optimasi berdasarkan persamaan 2.9 dan 2.15. Masalah tersebut dapat dibentuk persamaan dalam bentuk *Lagrangian* seperti berikut.

$$L(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{k=1}^{K} a_k[y_k(\mathbf{w}.\mathbf{x} + b) - 1] \quad (2.16)$$

dimana a_k merupakan *Lagrange multipliers.* Persamaan $L(\mathbf{w}, b)$ digunakan untuk mencari nilai a_k (*support vektor)* dengan membuat $L(\mathbf{w}, b)$optimum. $L(\mathbf{w}, b)$ optimum didapat dengan cara mencari turunan parsial $L(\mathbf{w}, b)$ terhadap $\mathbf{w}$ dan b. Penurunan secara parsial dari fungsi $L(\mathbf{w}, b)$ terhadap $\mathbf{w}$ akan menghasilkan

$$\mathbf{w} = \sum_{k=1}^{K} a_k y_k \mathbf{x} \quad (2.17)$$

lalu penurunan $L(\mathbf{w}, b)$ terhadap b menghasilkan persamaan 2.18.

$$-\sum_{k=1}^{K} a_k y_k = 0 \text{ dan } b = 1 - \mathbf{w}^T\mathbf{x} \quad (2.18)$$

Dengan mensubstitusi persamaan 2.17 ke persamaan 2.16, maka akan memperoleh fungsi *hyperplane* seperti berikut

$$L(\boldsymbol{a}) = \sum_{k=1}^{K} a_k - \frac{1}{2}\sum_{k=1}^{K}\sum_{l=1}^{L} a_k a_l y_k y_l \langle \mathbf{x}_+, \mathbf{x}_- \rangle \quad (2.19)$$

sehingga akan dicari vektor $\boldsymbol{a}$ untuk fungsi pemisah *hyperplane* yang optimum. Mencari vektor $\boldsymbol{a}$ dengan algoritma *Quadratic Programming* yang hanya bisa diselesaikan dengan komputasi.

b. *SVM Non Linearly Separable Data dengan Metode Kernel*

Klasifikasi data yang tidak dapat dipisahkan secara linier memerlukan modifikasi pada formula SVM agar dapat menemukan

solusinya. Pencarian fungsi *hyperplane* optimal akan memperhatikan data-data yang tidak berada pada kelasnya (*missclassification error*) yang dilambangkan dengan ξ. Sehingga persamaan 2.9 menjadi sebagai berikut.

$$y(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1 - \xi \tag{2.20}$$

Persamaan *Lagrange Multiplier* pada data yang tidak dapat dipisahkan secara linier adalah sebagai berikut

$$L(\boldsymbol{a}) = \sum_{k=1}^{K} a_k - \frac{1}{2} \sum_{k=1}^{K} \sum_{l=1}^{L} a_k a_l y_k y_l K\langle \mathbf{x}_+, \mathbf{x}_- \rangle \tag{2.21}$$

Penyelesaian data jenis ini dapat dilakukan dengan metode kernel. Metode kernel berkerja dengan mentransformasi data ke dalam dimensi ruang fitur sehingga dapat dipisahkan secara linier pada *feature space*. Sebagai contoh, terdapat suatu data *x* di input *space* pada *feature space* dengan menggunakan fungsi transformasi $\mathbf{x} \rightarrow \emptyset(\mathbf{x})$. Sehinga nilai $\mathbf{w} = \sum_{k=1}^{K} a_k y_k \emptyset(\mathbf{x})$ dan fungsi hasil data latih yang dihasilkan adalah

$$f(\mathbf{x}_k) = \sum_{k=1}^{K} a_k y_k K\langle \mathbf{x}_+, \mathbf{x}_- \rangle + b \tag{2.22}$$

Pada praktiknya, *feature space* dapat memiliki dimensi yang tinggi dari vektor *input space*. (Gunn, 1998) Kernel memperhatikan parameter tertentu yakni parameter *C*. *C* adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Sehingga peran dari *C* adalah meminimalkan kesalahan pelatihan dan mengurangi kompleksitas model. Berikut adalah fungsi kernel untuk metode SVM.

Tabel 2.1 Fungsi Kernel SVM

Fungsi Kernel	Rumus $K(\mathbf{x}_+, \mathbf{x}_-)$	Parameter
Linier	$\mathbf{x}_+^{\mathrm{T}}\mathbf{x}_-$	C
RBF	$exp(-\gamma\|(\mathbf{x}_+ - \mathbf{x}_-)^2\|)$	γ dan C

dimana $\mathbf{x}_+$ adalah *support vector* sebagai kategori positif, $\mathbf{x}_+$ adalah *support vector* kategori negatif. Untuk kernel *Radial Basis Function* (RBF), terdapat tambahan parameter yakni parameter γ (*gamma*) dalam fungsi kernelnya. Dimana secara teknis, parameter γ adalah kebalikan dari standar deviasi dari kernel RBF, yang digunakan sebagai ukuran kesamaan antara dua titik. Secara intuitif, nilai γ kecil mendefinisikan fungsi RBF dengan varians yang besar. Dalam hal ini, dua poin dapat dianggap sama meskipun jauh dari satu sama lain. Di sisi lain, nilai γ yang besar berarti mendefinisikan fungsi RBF dengan varians yang kecil dan dalam hal ini, dua poin dianggap sama hanya jika keduanya berdekatan satu sama lain. (Pinto, 2016).

2.3.4 *K-Nearest Neighbor* (KNN)

Nearest Neighbor (NN) adalah suatu metode yang banyak digunakan dalam *Data Mining*. Metode NN diklasifikasikan sebagai *lazy learner,* karena metode ini menunda proses pelatihan atau tidak melakukan sama sekali sampai ada data uji yang ingin diketahui label kelasnya. Ketika ada data uji yang ingin diketahui kelasnya, maka metode NN baru akan menjalankan algoritmanya (Prasetyo, 2014). *K-Nearest Neighbor* merupakan salah satu metode paling tua yang berasal dari konsep NN, paling sederhana dan popular namun memiliki kinerja yang mampu menyamai kinerja metode lain yang lebih rumit.

Pemilihan nilai K merupakan hal yang sangat mempengaruhi kinerja metode KNN (Prasetyo, 2014). Nilai K yang terbaik

tergantung pada data. Secara umum, nilai K yang tinggi akan mengurangi efek noise pada klsifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Isu lain yang berkaitan dengan pemilihan nilai K adalah apakah nilai tersebut ganjil atau genap. Ketika nilai K ganjil, resiko dua kelas atau lebih memperoleh jumlah suara sama sangat kecil. Sedangkan ketika K bernilai genap, ada resiko dua kelas atau lebih memproleh suara yang sama. KNN biasanya diaplikasikan dengan 10-*fold validation*, menggunakan aturan jarak terdekat untuk mengatasi pemilihan kelas karena jumlah suara yang sama kuat, dan pemberian bobot pengaruh berdasarkan jarak untuk memberikan pengaruh yang lebih kecil terhadap data yang terletak jauh dari data uji (Arianto, 2017).

Metode KNN tidak membutuhkan pembangunan model dikarenakan model yang akan diujikan yaitu keseluruhan data latih. Dalam melakukan prediksi terhadap data uji, maka data latih mulai digunakan untuk mencari kemiripan data sesuai jumlah K yang ditentukan terlebih dahulu. Algoritma KNN mempunyai kelebihan yaitu mudah dimengerti dan diterapkan, proses pelatihan berlangsung sangat cepat, tangguh terhadap data latihan yang mempunyai *noise*, serta bisa diimplementasikan pada kasus banyak kelas (*multiclass*) (Bhavsar & Ganatra, 2014). Sedangkan kelemahan yang dimiliki KNN yaitu sensitif terhadap struktur data atau urutan data. (Duda & Hart, 1973).

Algoritma K-*Nearest Neighbor* bekerja dengan langkah pengklasifikasian sebagai berikut.

1. Menentukan nilai K
   Penentuan nilai K dimulai dari K=1 saat menggunakan data uji untuk tingkat kesalahan pengklasifikasian. Nilai K ditambah sampai mendapatkan nilai kesalahan yang paling minimum, atau didapatkan didapatkan nilai ketepatan akurasi yang paling tinggi (Han & Kamber, 2012)

2. Menghitung jarak antara data latih dengan data uji
   Data yang digunakan dalam pengklasifikasian metode KNN adalah dengan frekuensi kata dari setiap kata yang muncul pada suatu *tweet*. Data yang berupa data numerik tersebut dapat dihtung jarak antar data latih dengan data uji dengan menggunakan jarak *eucliden*, dengan rumus sebagai berikut.

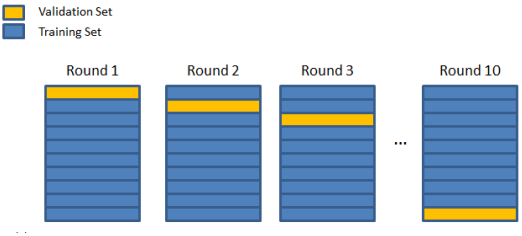
$$\left(\sum_{m=1}^{M}(x_{1m}-x_{2m})^2\right)^{1/2} \tag{2.23}$$

   dengan x_{1m} adalah nilai pada latih data, x_{2m} adalah nilai x pada data uji, dan M adalah batas jumlah banyaknya data.
3. Mengurutkan jarak terdekat
   Data dari semua perhitungan nilai jarak terdekat akan diurutkan berdasarkan nilai yang terkecil, sehingga akan diperoleh jarak yang paling dekat.
4. Mengklasifikasikan kelas (klasifikasi *Nearest Neighbor*)
   Pengklasifikasian kelas ditinjau dari jarak antara data latih dan data uji sebanyak nilai K. Setelah itu dilihat frekuensi suara tiap kelas dan data latih akan diklasifikasikan ke kelas dengan suara mayoritas.

2.3.5 *K-fold Cross Validation*

K-fold cross validation adalah salah satu metode yang digunakan untuk mempartisi data menjadi data *training* dan data *testing*. Metode ini banyak digunakan peneliti karena dapat mengurangi bias yang terjadi dalam pengambilan sampel. *K-fold cross validation* secara berulang membagi data menjadi data *training* dan data *testing*, dimana setiap data mendapat kesempatan menjadi data *testing* (Gokgoz, 2015). K merupakan besar angka partisi data yang

digunakan untuk pembagian *training testing.* Berikut merupakan ilustrasi pembagian data menggunakan *K-fold cross validation*.



Gambar 2.4 Ilustrasi Pembagian Data
(Sumber : codesachin.wordpress.com)

2.3.6 Ketepatan Klasifikasi

Pengukuran ketepatan klasifikasi dilakukan untuk melihat performa klasifikasi yang telah dilakukan. Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah pada setiap kelas prediksi dan kelas aktual. Pengukuran yang sering digunakan untuk menghitung ketepatan klasifikasi adalah *precision, recall,* dan akurasi (Hotho, 2005). Akurasi merupakan persentase *tweet* yang teridentifikasi secara tepat dari total dokumen dalam proses klasifikasi. Akurasi digunakan untuk menghitung ketepatan klasifikasi sebuah dokumen yang mempunyai data yang *balanced* pada tiap kategorinya. *Precision* adalah rasio dari prediksi terkategori benar terhadap pada kelas emosi tersebut. *Recall* adalh rasio prediksi terkategori benar terhadap total prediksi dari kelas emosi awal..

Tabel 2.2 *Confusion Matrix* Tiga Kelas

F_{ap}		Kelas Prediksi (p)		
		Senang	**Sedih**	**Marah**
Kelas Asli (a)	**Senang**	F_{11}	F_{12}	F_{13}
	Sedih	F_{21}	F_{22}	F_{23}
	Marah	F_{31}	F_{32}	F_{33}

dengan :
F_{11} = jumlah *tweet* kelas senang terprediksi tepat di kelas senang.
F_{12} = jumlah *tweet* kelas senang terprediksi di kelas sedih.
F_{13} = jumlah *tweet* kelas senang terprediksi di kelas marah.
F_{21} = jumlah *tweet* kelas sedih terprediksi di kelas senang.
F_{22} = jumlah *tweet* kelas sedih terprediksi tepat di kelas sedih.
F_{23} = jumlah *tweet* kelas sedih terprediksi di kelas marah.
F_{31} = jumlah *tweet* kelas marah terprediksi di kelas senang.
F_{32} = jumlah *tweet* kelas marah terprediksi di kelas sedih.
F_{33} = jumlah *tweet* kelas marah terprediksi tepat di kelas marah.

dan berikut merupakan rumusnya dari pengukuran ketepatan klasifikasinya.

$$Recall = \frac{1}{3}\left(\frac{F_{11}}{F_{11}+F_{21}+F_{31}} + \frac{F_{22}}{F_{12}+F_{22}+F_{32}} + \frac{F_{33}}{F_{13}+F_{23}+F_{33}}\right) \quad (2.24)$$

$$Precision = \frac{1}{3}\left(\frac{F_{11}}{F_{11}+F_{21}+F_{31}} + \frac{F_{22}}{F_{12}+F_{22}+F_{32}} + \frac{F_{33}}{F_{13}+F_{23}+F_{33}}\right) \quad (2.25)$$

$$\text{Akurasi} = \frac{F_{11}+F_{22}+F_{33}}{F_{11}+F_{21}+F_{31}+F_{12}+F_{22}+F_{32}+F_{13}+F_{23}+F_{33}} \quad (2.26)$$

2.3.7 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang sering digunakan.

Gambar 2.5 Contoh *Word Cloud*
(Sumber : google.com)

Word cloud merupakan representasi grafis dari sebuah dokumen yang dilakukan dengan *plotting* kata-kata yang sering mencul pada sebuah dokumen pada ruang dua dimensi. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut. (Castella, 2014).

2.4 Twitter

Twitter adalah sebuah situs *web* yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan tweet (Twitter, 2016).. *Tweet* adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna. *Tweet* bisa dilihat secara publik, namun pemilik *twitter* dapat membatasi pengiriman pesan ke daftar teman-teman mereka saja. Pengguna dapat melihat *tweet* pengguna lain yang dikenal dengan sebutan pengikut *(follower)*.

BAB III

METODOLOGI PENELITIAN

3.1 Sumber Data dan Variabel Penelitian

Data yang akan digunakan adalah kumpulan *tweet* mengenai Presiden Joko Widodo. Data didapat dari Twitter API (*Application Programming Interface*) dan akan diambil sebanyak 450 *tweet* dengan interval seminggu selama 2 bulan dimulai dari tanggal 2 April 2018 hingga 28 Mei 2018.

Tabel 3.1 Contoh Data *Tweet*

No	Tweet	Emosi
1	@jokowi Aamiinnn....terima kasih pa Jokowi atas kepeduliannya…	Senang
2	Ya allah Kabulkan doa hamba mu ganti presiden…	Sedih
3	@jokowi munafik nih, jilat ludah sendiri …	Marah
.	…	…

Variabel penelitian yang digunakan dalam penelitian ini setelah dilakukan praproses pada data teks *tweet* terdiri dari variabel predictor (x) yaitu kata dasar setiap *tweet* dan variabel respon (y) yaitu klasifikasi emosi *tweet* (senang, sedih, dan marah).

Tabel 3.2 Variabel Penelitian

Variabel	Keterangan	Skala
y	Kelas emosi 1. Senang 2. Sedih 3. Marah	Nominal
x	Kata	Nominal

3.2 Struktur Data

Data yang berjumlah 450 dengan masing-masing *tweet* emosi berjumlah 150 dibagi menjadi data *training* dan data *testing* dengan menggunakan pembagian data 10-*fold cross validation*. Struktur setelah dilakukan praproses ditunjukkan pada Tabel 3.3.

Tabel 3.3 Contoh Struktur Data Penelitian

Tweet	***y***	x_1	x_2	x_3	**…**	x_n
1	*1*	x_{11}	x_{12}	x_{13}	…	x_{1n}
2	*1*	x_{21}	x_{22}	x_{23}	…	x_{2n}
.	.	.	.	.	.	.
.	.	.	.	.	.	.
k	*2*	x_{k1}	x_{k2}	xk_3	…	xk_n

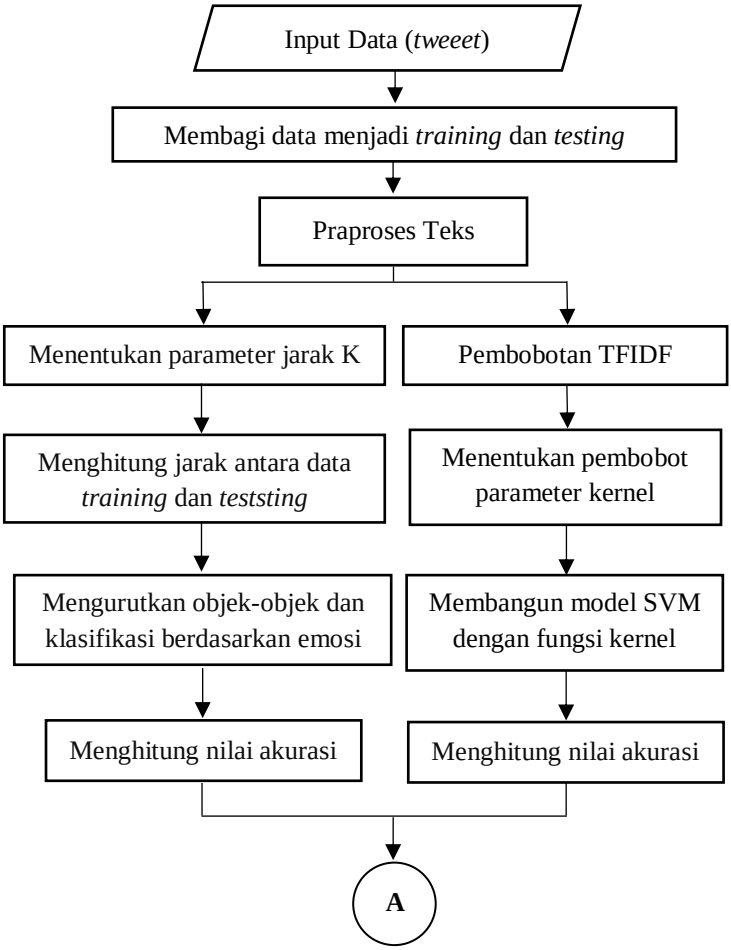
3.3 Langkah Penelitian

Langkah-langkah analisis yang dilakukan dalam penelitian ini adalah sebagai berikut.

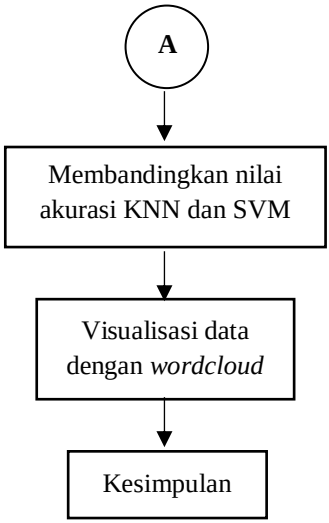
1. Mengambil data *tweet* dengan Twitter API.
   a. Memasukkan *keyword* yang berhubungan dengan Presiden Jokowi dengan kata kunci “jokowi” dan“joko Widodo”
   b. Menyimpan hasil *searching* ke database.
2. Menyiapkan data *tweet* dan daftar *stopwords.*
   a. Data *tweet* dibagi menjadi data latih dan data uji menggunakan *10-fold cross validation* dengan perbandingan sebesar 90%:10%.
   b. Daftar *stopwords* didapat dari F. Z. Tala yang berjudul “*A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*” (Tala, 2003).
3. Praproses Teks
   a. Menghapus *tweet* yang tidak mengandung salah satu dari ketiga emosi

   b. Menghapus *link* URL, simbol *retweet* (*response tweet*) "RT", simbol *hashtag* # serta simbol-simbol lain yang tidak diperlukan
   c. Melakukan *case folding*, yaitu mengubah semua teks dengan huruf kecil (non kapital)
   d. Melakukan *steeming* untuk mendapatkan kata dasar
   e. Menghapus kata pada *tweet* berdasarkan daftar *stopwords*.
   f. Melakukan *tokenizing* untuk memecah *tweet* menjadi kata
   g. Mengubah data *tweet* kedalam bentuk frekuensi kemunculan kata
4. Klasifikasi data menggunakan *Support Vector Machine*
   a. Melakukan pembobotan TFIDF untuk merubah teks kedalam bentuk numerik
   b. Menentukan pembobot parameter pada SVM tiap kernel
   c. Mengklasifikasikan hasil TFIDF untuk tiap jenis kernel
   d. Membangun fungsi SVM berdasarkan hasil kernel dengan ketepatan klasifikasi terbaik.
   e. Menghitung hasil ketepatan klasifikasi
5. Klasifikasi data menggunakan *K-Nearest Neighbor*
   a. Menentukan parameter jarak.
   b. Menghitung jarak *euclidean.*
   c. Klasifikasi *Nearest Neighbor* berdasarkan kelas emosi
   d. Menghitung hasil ketepatan klasifikasi
6. Evaluasi hasil klasifikasi dari metode *Support Vector Machine* menggunakan hasil akurasi kernel terbaik dengan metode *K-Nearest Neighbor*
7. Melakukan visualisasi *word cloud* berdasarkan kata-kata yang muncul pada tiap emosi
8. Interpretasi dan menarik kesimpulan

3.4 Diagram Alir



Gambar 3.1 Diagram Alir Penelitian



Gambar 3.1 Diagram Alir Penelitian (Lanjutan)

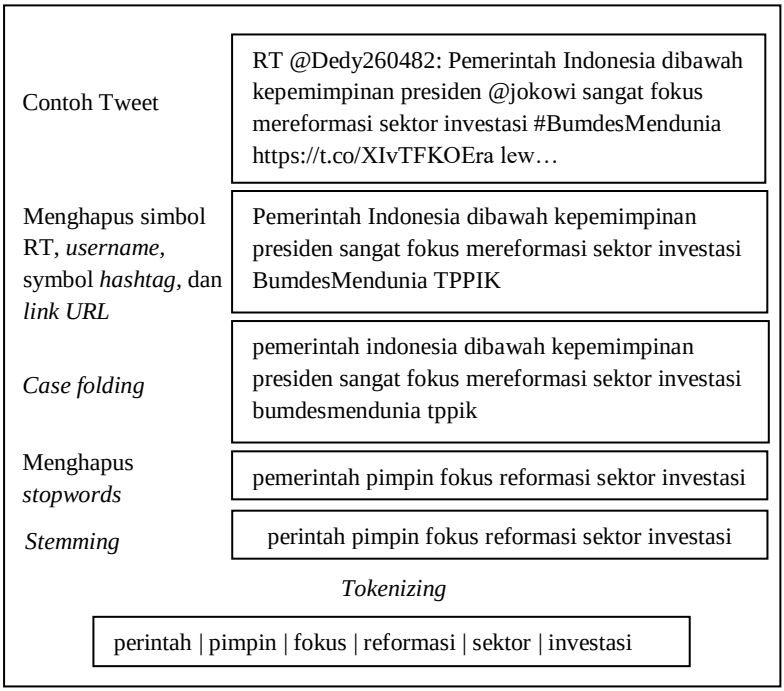
(*Halaman ini sengaja dikosongkan*)

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Praproses dan Karakteristik Data

Data *tweet* yang dikumpulkan masih berupa data kotor. Artinya data tersebut masih memuat *username*, *link URL*, kata-kata yang dianggap bukan merupakan kata, dan simbol-simbol lainnya dan tanda baca, sehingga perlu dilakukan praproses guna mendapatkan data *tweet* yang tidak memuat hal-hal tersebut. Data *tweet* tersebut dilakukan praproses teks yang meliputi *case folding*, *stopword*, *stemming*, dan *tokenizing* dengan langkah-langkah seperti pada gambar 4.1 berikut.



Gambar 4.1 Contoh Praproses Teks

Setelah dilakukan praproses, akan terbentuk struktur data berisi data *tweet* yang telah dilakukan pengahupasan *stopwords* terhadap data kata-kata dari hasil proses *stemming* yang menjadi variabelnya. Sehingga akan terlihat pada suatu *tweet* berapa frekuensi variable dari kata-kata tersebut. Berikut merupakan struktur data *tweet* mengenai Presiden Jokowi sesudah dilakukan praproses data.

Tabel 4.1 Struktur Data Setelah Praproses

Tweet	**Emosi**	**…**	**apresiasi**	**…**	**dua**	**…**	**rezim**	**…**
..reformasi sektor..	1	…	0	…	0	…	0	…
…dua periode	1	…	0	…	1	…	1	…
… piawai	1	…	0	…	0	…	0	…
…rakyat cinta	1	…	0	…	0	…	0	…
.rezim ini…	1	…	0	…	0	…	1	…
..bangga puji..	1	…	0	…	0	…	0	…
…kasih peduli..	1	…	0	…	0	…	0	…
..wakil terbaik..	1	…	0	…	0	…	0	…
..semoga dua periode..	1	…	1	…	1	…	0	…
…	.	…	…	…	…	…	…	…
…	.	…	.	…	…	…	…	…
…rezim sadar…	3	…	0	…	0	…	1	…

Data yang telah berbentuk *document term matrix* seperti pada Tabel 4.1 tersebut, dapat dilakukan perhitungan jumlah kata yang selanjutnya akan menjadi jumlah variabel. Data *tweet* memiliki jumlah kata sebanyak 1069 kata. Setelah terbentuk struktur data, dilakukan perhitungan frekuensi kemunculan kata yang ditampilkan pada Tabel 4.2.

Hasil frekuensi data dapat digunakan untuk mencari informasi lain mengenai kata-kata yang tercantum. Misalkan ingin diketahui seberapa banyak suatu kata muncul dalam suatu kelas tertentu, contoh frekuensi kemunculan kata pada suatu kelas emosi ditunjukkan pada Tabel 4.3.

. Tabel 4.2 Frekuensi Kemunculan Kata Tertinggi

Kata	Jumlah	Kata	Jumlah
tipu	36	dukung	25
ganti	35	perintah	24
rakyat	30	periode	20
sikap	30	suka	20
nyinyir	29	korupsi	19
temu	27	tukang	18
paham	27	bikin	17
negarawan	27	gua	17
pol	27	doang	17
rezim	26	rapat	17

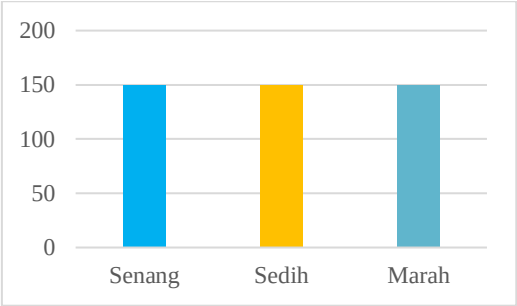
. Tabel 4.3 Frekuensi Kemunculan Kata pada Setiap Kelas

Kata	Senang	Sedih	Marah
…	…	…	…
bangga	10	0	0
bahagia	8	0	0
baper	0	5	0
israel	0	9	2
tukang	0	0	18
konsolidasi	0	0	17
janji	0	8	2
…	…	…	…

Dari table 4.3 dapat dilihat kata “janji” muncul dalam 8 *tweet* pada kelas Sedih dan 2 *tweet* pada kelas Marah, dan tidak muncul sama sekali pada kelas senang. Contoh lain adalah kata “bangga” yang hanya muncul pada *tweet* kelas Senang

Dala penelitian ini, jumlah *tweet* setiap kelas emosi dibuat sama. Hal tersebut dikarenakan penelitian ini tidak bertujuan untuk mencari tahu bagaimana respon pengguna *twitter* terhadap Presiden Jokowi, namun ingin mengetahui apakah *classifier* yang di-

gunakan dapat direkomendasikan untuk penelitian selanjutnya dengan lebih melihat respon pengguna *twitter*.



Gambar 4.2 *Bar Chart* Frekuensi Data Tiap Emosi

Frekuensi kategori data *training* pada Gambar 4.2 menunjukkan bahwa ketiga kelas emosi mempunyai jumlah data yang persis sama, yakni sebesar 150 *tweet* pada setiap emosi. Dikarenakan data yang *balance*, maka perhitungan ketepatan klasifikasi bisa menggunakan akurasi, *recall*, dan *precision*.

4.2 Klasifikasi Menggunakan *Support Vector Machine*

4.2.1 Pembobotan Kata dengan TFIDF

Sebelum dapat diklasifikasikan menggunakan SVM, bentuk data *tweet* yang semula adalah teks harus diubah terlebih dahulu menjadi numerik. Mulanya, akan dicari *Term frequency* (TF), dimana TF merupakan frekuensi kemunculan kata pada *tweet*. Agar lebih mudah dipahami, berikut akan dicontohkan ilustrasi dalam penghitungan TF. Namun dikarenakan data yang terlalu banyak, yakni sejumlah 1069 kata dan 450 *tweet*, maka hanya akan diperlihatkan beberapa contoh kata dan *tweet* saja.

Diambil sebanyak 5 kata dan 5 *tweet* dalam pengilustrasian di Tabel 4.4. Dapat dilihat kata “apresiasi” muncul sebanyak satu kali pada *tweet* pertama, ketiga, dan kelima namun tidak muncul sama sekali pada *tweet* kedua dan keempat. Kata “dua” muncul sebanyak

satu kali pada *tweet* pertama dan keempat, dan dua kali pada *tweet* kedua, begitu seterusnya untuk setiap kata yang muncul.

Tabel 4.4 Ilustrasi Sampel Data pada Perhitungan TF

Kata	*Term Frequency* (TF)				
	T1	T2	T3	T4	T5
apresiasi	1	0	1	0	1
dua	1	2	0	1	0
dukung	0	1	0	0	1
janji	0	0	0	1	0
rezim	0	1	0	0	0

Setelah ditemukan nilai TF per kata untuk tiap *tweet*, selanjutnya adalah mencari *Document Frequency* (DF), dimana DF adalah jumlah kata yang muncul dalam keseluruhan *tweet.* Berikut ditunjukkan ilustrasi dari perhitungan DF.

Tabel 4.5 Ilustrasi Sampel Data pada Perhitungan DF

Kata	*Term Frequency* (TF)					DF
	T1	T2	T3	T4	T5	
apresiasi	1	0	1	0	1	3
dua	1	2	0	1	0	4
dukung	0	1	0	0	1	2
janji	0	0	0	1	0	1
rezim	0	1	0	0	0	1

Pada Tabel 4.5 ditunjukkan kata “apresiasi” mempunyai nilai DF sebesar 3. Hal tersebut dikarenakan kata “apresiasi” muncul sebanyak tiga kali, yakni satu kali pada *tweet* pertama, ketiga, dan kelima. Lalu kata “dua” muncul sebanyak satu kali pada *tweet* pertama dan keempat, dan muncul dua kali pada *tweet* kedua, sehingga nilai DF dari kata “dua” adalah 4, begitu pula untuk kata yang lainnya. Selanjutnya adalah menghitung nilai *Inverse Document Frequency* (IDF) dengan menggunakan rumus 2.2 untuk setiap kata. Nilai *N* adalah jumlah keseluruhan kata yang ditemukan, dalam ilustrasi yakni sebanyak 5, dan DF_iadalah nilai DF dari setiap kata. Berikut adalah ilustrasi penghitungan IDF.

Tabel 4.6 Ilustrasi Sampel Data pada Perhitungan IDF

Kata	*Term Frequency (TF)*					*DF*	*IDF*
	T1	*T2*	*T3*	*T4*	*T5*		
apresiasi	1	0	1	0	1	3	$\log\left(\frac{5}{3}\right)=0.2218$
dua	1	2	0	1	0	4	$\log\left(\frac{5}{4}\right)=0.0969$
dukung	0	1	0	0	1	2	$\log\left(\frac{5}{2}\right)=0.3979$
janji	0	0	0	1	0	1	$\log\left(\frac{5}{1}\right)=0.6989$
rezim	0	1	0	0	0	1	$\log\left(\frac{5}{1}\right)=0.6989$

Setelah ditemukan nilai TF dan IDF, maka pembobotan kata dalam *tweet* dapat dilakukan dengan mengkalikan nilai TF kata pada suatu *tweet* dan nilai IDF setiap kata. Sehingga setiap kata mempunyai bobot yang berbeda-beda antara *tweet* satu dengan *tweet* lainnya. Berikut adalah ilustrasi hasil perhitungan antara TF dan IDF.

Tabel 4.7 Ilustrasi Sampel Data pada Perhitungan TFIDF

Kata	*TFIDF*				
	T1	*T2*	*T3*	*T4*	*T5*
apresiasi	0.2218	0	0.2218	0	0.2218
dua	0.0969	1.3979	0	0.0969	0
dukung	0	0.3979	0	0	0.3979
janji	0	0	0	0.6989	0
rezim	0	0.6989	0	0	0

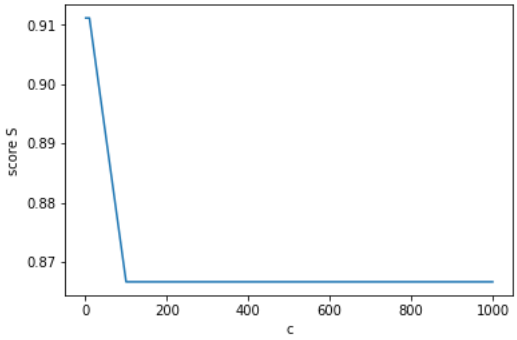
Kata “apresiasi” mempunyai IDF sebesar 0.2218. Karena kata tersebut muncul sebanyak satu kali pada *tweet* pertama, ketiga, dan kelima, maka nilai TFIDF pada *tweet-tweet* tersebut adalah sebesar 0.2218. Kata “dua” mempunyai nilai IDF 0.0969, dan muncul sebanyak dua kali pada *tweet* kedua, sehingga nilai TFIDFnya adalah sebesar $2 \times 0.0969 = 1.3979$. Begitu seterusnya untuk kata lain pada *tweet-tweet* yang mengandung kata tersebut.

Setelah dilakukan perhitungan pembobot kata setiap *tweet*, maka klasifikasi menggunakan SVM dapat dilakukan. Pembobotan

pada kata menyebabkan adanya sebaran data dan tidak dapat dilihat dengan jelas apakah data dapat dipisahkan secara linier atau tidak. Sehingga pada klasifikasi menggunakan SVM akan dilakukan dengan SVM kernel *linear* dan SVM RBF.

4.2.2 Klasifikasi Menggunakan Kernel *Linear*

Klasifikasi SVM dengan kernel *linear*, mempertimbangkan parameter *C*. Penentuan parameter ini sangat penting karena nilai *C* akan menghasilkan hasil akurasi yang berbeda pula. Untuk mencari nilai parameter *C* yang baik dapat dilakukan dengan menggunakan metode *trial error*, yaitu mencoba nilai parameter *C* yang berbeda-beda hingga mendapatkan hasil ketepatan klasifikasi yang terbaik yang paling optimum. Nilai *C* yang dicoba yaitu mulai dari $C = 10^0$ hingga $C = 10^3$. Gambar 4.4 menunjukkan hasil pengukuran performa klasifikasinya.



Gambar 4.3 Grafik Hasil Pengukuran Performa Klasifikasi SVM Kernel *Linear*

Gambar 4.3 menunjukkan nilai *C* mulai dari $C = 10^0$ hingga $C = 10^3$ serta hasil akurasi dari tiap *C*-nya, semakin tinggi grafiknya maka semakin tinggi nilai akurasinya. Dapat dilihat bahwa grafik dengan nilai akurasi paling optimum adalah saat $C = 10^0$, dengan akurasi melewati 90%. Hasil tersebut merupakan rata-rata hasil ketepatan akurasi dari sepuluh kali pembagian data uji dan data latih menggunakan 10-*fold cross validation*. Untuk jelasnya, hasil ketepatan klasifikasi ditampilkan pada Tabel 4.8.

Tabel 4.8. Ketepatan Klasifikasi dengan SVM *Linear* C=1

Fold ke-	***Precision***	***Recall***	**Akurasi**
1	0,91	0,91	0,91
2	0,87	0,84	0,84
3	0,98	0,98	0,98
4	0,98	0,98	0,98
5	0,98	0,98	0,98
6	0,96	0,96	0,96
7	0,93	1	0,93
8	0,96	0,96	0,96
9	0,98	0,98	0,98
10	0,96	0,96	0,96
Rata-rata	**0.951**	**0.955**	**0.948**

Tabel 4.8 menunjukkan hasil ketepatan klasifikasi prediksi SVM kernel *linear* menggunakan *C*=1untuk tiap *fold*, dimana tiap *fold* terdiri dari data uji sebanyak 45 dan data latih sebanyak 405. Dapat dilihat bahwa dengan menggunakan nilai parameter $C = 1$ menghasilakan ketepatan klasifikasi berupa rata-rata nilai *precision* sebesar 95.1%, rata-rata nilai *recall* sebesar 95.5%, dan nilai akurasi sebesar 94.8%.

Precision adalah rasio dari prediksi terkategori benar terhadap pada kelas emosi tersebut. Dari hasil pada Tabel 4.8 diketahui bahwa hasil rata-rata *precision* adalah sebesar 95.1%, artinya adalah sistem mampu mengklasifikasikan dengan ketepatan hingga hasil prediksi yang tepat berada dikelas emosinya rata-rata sebesar 95.1%. Lalu nilai *Recall* adalah rasio prediksi terkategori benar terhadap total prediksi dari kelas emosi awal. Hasil *recall* sebesar 95.5% menunjukkan bahwa sistem mampu memprediksi benar kedalam kelas emosinya rata-rata hingga 95.5%. Dan nilai akurasi sebesar 94.8% menunjukkan bahwa dengan menggunakan SVM kernel RBF, dari 45 data uji *tweet* terprediksi benar terdapat di dalam kelas emosinya adalah rata-rata sebanyak 41 *tweet*, dan 4 *tweet* lainnya terjadi *misclassification* atau salah prediksi. *Confusion matrix* dari tiap *fold* pada metode SVM kernel *linear* dapat dilihat pada Lampiran 5.

4.2.3 Klasifikasi Menggunakan Kernel RBF

Untuk kernel RBF, ada tambahan parameter yang harus diperhatikan selain parameter *C*, yakni parameter γ (gamma). Penentuan kombinasi kedua parameter ini sangat penting dalam kernel RBF, karena nilai *C* dan γ yang berbeda akan menghasilkan hasil akurasi yang berbeda pula. Untuk mencari nilai parameter *C* dan γ yang baik dapat dilakukan dengan menggunakan metode *trial error*, yaitu mencoba nilai parameter *C* dan γ yang berbeda-beda hingga mendapatkan hasil ketepatan klasifikasi yang terbaik. Dan didapatkan hasil ketepatan klasifikasi terbaik dengan kombinasi parameter *C* dan γ beruturt-berturut sebesar 10 dan 0.1.

Selanjutnya parameter tersebut akan digunakan untuk memprediksi data uji untuk dilihat akurasi, *precision*, dan *recall*. Hasil ketepatan klasifikasi prediksi merupakan rata-rata dari sepuluh kali pembagian data uji dan data latih menggunakan 10-*fold cross validation*. Untuk lebih jelasnya, hasil ketepatan klasifikasi ditampilkan pada Tabel 4.9.

Tabel 4.9. Ketepatan Klasifikasi dengan SVM RBF C=100, γ =1

Fold ke-	***Precision***	***Recall***	**Akurasi**
1	0,91	0,91	0,91
2	0,89	0,87	0,87
3	1	1	1
4	0,98	0,98	0,98
5	0,98	0,98	0,98
6	0,96	0,96	0,96
7	0,93	0,93	0,93
8	0,94	0,93	0,93
9	0,98	0,98	0,98
10	0,98	0,98	0,98
Rata-rata	**0,955**	**0,952**	**0,952**

Tabel 4.9 menunjukkan hasil ketepatan klasifikasi prediksi SVM kernel RBF menggunakan *C*=100 dan γ=1 untuk tiap *fold*, dimana tiap *fold* terdiri dari data uji sebanyak 45 dan data latih sebanyak 405. Dapat dilihat bahwa dengan kombinasi nilai para-

meter C=100 dan γ=1 menghasilakan ketepatan klasifikasi berupa rata-rata nilai *precision* sebesar 95.5%, rata-rata nilai *recall* sebesar 95.2%, dan nilai akurasi sebesar 95.2%.

Precision adalah rasio dari prediksi terkategori benar terhadap pada kelas emosi tersebut. Dari hasil pada Tabel 4.9 diketahui bahwa hasil rata-rata *precision* adalah sebesar 95.5%, artinya adalah sistem mampu mengklasifikasikan data dengan ketepatan hingga hasil prediksi yang tepat berada dikelas emosinya rata-rata sebesar 95.5%. Lalu nilai *Recall* adalah rasio prediksi terkategori benar terhadap total prediksi dari kelas emosi awal. Hasil *recall* sebesar 95.2% menunjukkan bahwa sistem mampu memprediksi benar kedalam kelas emosinya rata-rata hingga 95.2%. Dan nilai akurasi sebesar 95.2% menunjukkan bahwa dengan menggunakan SVM kernel RBF, dari 45 data uji *tweet* terprediksi benar terdapat di dalam kelas emosinya adalah rata-rata sebanyak 43 *tweet*, dan sisa 2 *tweet* lainnya terjadi *missclassification* atau salah prediksi. *Confusion matrix* dari tiap *fold* pada metode SVM kernel RBF dapat dilihat pada Lampiran 6.

4.2.4 Model *Support Vector Machine*

Pembahasan hasil ketepatan klasifikasi menggunakan SVM kernel *linear* dan SVM kernel RBF menunjukkan bahwa SVM kernel RBF mempunyai hasil ketepatan klasifikasi yang lebih baik. Nilai γ yang digunakan, yakni sebesar 1, disiubstitusikan pada persamaan kernel RBF yang terdapat pada Tabel 2.1 sehingga akan ditemukan fungsi kernel RBF sebagai berikut.

$$K\langle \mathbf{x}_+, \mathbf{x}_- \rangle = exp\left(-\left(1 \times (\mathbf{x}_+ - \mathbf{x}_-)^{\mathrm{T}}(\mathbf{x}_+ - \mathbf{x}_-)\right)\right)$$

Perlu diketahui pada tugas akhir ini, terdapat tiga label kelas yakni kelas emosi “Senang”, “Sedih”, dan “Senang”. Dan secara teoritis, klasifikasi *Support Vector Machine* adalah klasifikasi dengan dua kelas, dimana kelas tersebut dikategorikan menjadi positif ($\mathbf{x}_+$) dan negatif ($\mathbf{x}_-$). Oleh karena itu terdapat 3 fungsi pembentukan *hyperplane* yang digunakan, yaitu fungsi *hyperplane*

pemisah kelas "Senang" dan "Sedih", fungsi *hyperplane* pemisah kelas "Senang" dan "Marah", serta fungsi *hyperplane* pemisah kelas "Marah" dan "Sedih".

Fungsi kernel yang telah dihitung digunakan untuk membentuk fungsi *hyperplane* dengan cara mensubstitusikan nilai *support vector* kategori positif $(\mathbf{x}_+)$ dan *support vector* kategori negatif $(\mathbf{x}_-)$. Fungsi *hyperplane* dihitung dengan mensubstitusikan fungsi kernel ke persamaan 2.21. Sehingga didapat fungsi *hyperplane* pada setiap pemisah kelas emosi sebagai berikut.

Tabel 4.10 Fungsi *Hyperplane* SVM Kernel RBF

Emosi yang Dipisahkan	Persamaan *Hyperplane*
Senang dan Sedih	$f(\mathbf{x}) = \sum_{k=1}^{274} (0.5862 a_k \mathbf{x} + \cdots + 1.0346 a_k \mathbf{x}) - 0.0356$
Senang dan Marah	$f(\mathbf{x}) = \sum_{k=1}^{254} (0.6814 a_k \mathbf{x} + \cdots + 1.2688 a_k \mathbf{x}) - 0.3088$
Sedih dan Marah	$f(\mathbf{x}) = \sum_{k=1}^{232} (1.0986 a_k \mathbf{x} + \cdots + 1.218 a_k \mathbf{x}) - 0.2702$

Persamaan *hyperplane* yang telah didapat digunakan untuk mengklasifikasikan setiap dua kelas yang berbeda, sehingga terdapat tiga fungsi *hyperplane*. Pada persamaan *hyperplane* tersebut, a_k merupakan nilai koefisien dari *support vektor* dan $\mathbf{x}$ adalah nilai vektor observasi dari variable prediktor yang akan diklasifikasikan, yakni nilai TFIDF dari setiap kata.

4.3 Klasifikasi Menggunakan *K-Nearest Neighbor*

Metode *K-Nearest Neighbor* (KNN) adalah mencari klasifikasi berdasarkan jarak K terdekat. Pada penelitian ini, perhitungan jarak didasarkan pada nilai frekuensi kata. Untuk lebih memperje-

las cara kerja KNN, ilustrasi perhitungan akan ditampilkan pada Tabel 4.11

Tabel 4.11. Ilustrasi Sampel Data pada Jarak *Euclidean*

***Tweet* ke-**	**"semangat"**	**"kesal"**	**Emosi**
1	17	4	Sedih
2	3	3	Senang
3	8	2	Senang
4	2	0	Senang
5	3	0	Sedih
6	0	4	Marah
7	8	1	*unclassified*

Pada Tabel 4.11 diketahui bahwa *tweet* ketujuh tidak diketahui kelasnya. Untuk menemukan kelas yang belum diketahui tersebut, dengan frekuensi kata "semangat" sebanyak delapan, dan kata "kesal" sebanyak satu, maka dilakukan perhitungan jarak terdekat dengan rumus 2.21 yang akan ditunjukkan perhitungannya pada Tabel 4.12.

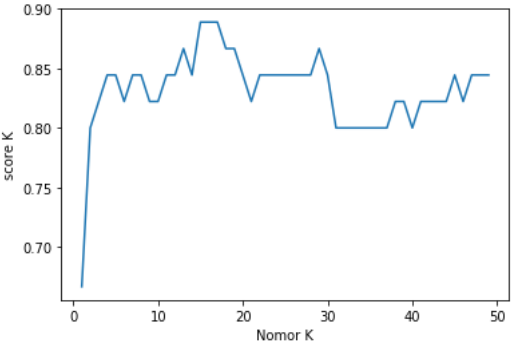
Tabel 4.12. Ilustrasi Perhitungan Jarak *Euclidean*

Tweet	**"keren"**	**"kesal"**	**Emosi**	**Jarak *Euclidean***
1	17	4	Senang	$\sqrt{(17-8)^2+(4-1)^2}=9.49$
2	**3**	**3**	**Sedih**	$\boldsymbol{\sqrt{(3-8)^2+(3-1)^2}=5.39}$
3	**8**	**2**	**Senang**	$\boldsymbol{\sqrt{(8-8)^2+(2-1)^2}=1}$
4	2	0	Senang	$\sqrt{(2-8)^2+(0-1)^2}=6.08$
5	**3**	**0**	**Senang**	$\boldsymbol{\sqrt{(3-8)^2+(0-1)^2}=5.09}$
6	0	4	Marah	$\sqrt{(0-8)^2+(4-1)^2}=9$

Dimisalkan nilai parameter K pada Ilustrasi perhitungan adalah sebesar 3. Dari perhitungan jarak *Euclidean* pada Tabel 4.12, maka didapatkan setiap sampel data dengan sebuah sampel yang belum diketahui kelasnya, yakni *tweet* ketujuh. Dengan nilai K tetangga terdekat adalah 3, maka dengan melihat 3 *tweet* dengan jarak *eculidean* terdekat, yakni *tweet* kedua, ketiga, dan kelima, de-

ngan kelas emosi berturut-turut adalah “Sedih”, “Senang”, dan “Senang”, maka ditentukan bahwa kelas *tweet* ketujuh adalah kelas emosi “Senang”. Penentuan ini berdasar kepada banyaknya kelas dengan jarak terdekat yang paling banyak mempengaruhi.

Tidak ada ketentuan untuk menentukan nilai K terbaik. Oleh karena itu, untuk menentuan jarak terdekat dengan metode KNN pada tugas akhir ini, akan digunakan beberapa nilai K yang berbeda untuk melihat pengaruh nilai K yang diberikan sehingga menghasilkan nilai akurasi yang paling optimum. Nilai K yang digunakan yaitu mulai dari K=1 hingga K=50. Berikut adalah grafik hasil pengukuran performa klasifikasinya.



Gambar 4.4 Grafik Hasil Pengukuran Performa Klasifikasi KNN

Gambar 4.4 menunjukkan nilai K mulai dari K=1 hingga K= 50 serta hasil akurasi dari tiap K-nya, semakin tinggi grafiknya maka semakin tinggi nilai akurasinya. Dapat dilihat bahwa grafik dengan nilai akurasi paling optimum adalah saat K=17, dengan akurasi hampir mencapai 90%. Hasil tersebut merupakan rata-rata dari hasil ketepatan akurasi dari sepuluh kali pembagian data uji dan data latih menggunakan 10-*fold cross validation*. Untuk lebih jelasnya, hasil ketepatan klasifikasi ditampilkan pada Tabel 4.13.

Tabel 4.13 menunjukkan hasil ketepatan klasifikasi KNN menggunakan K=17 untuk tiap *fold*, dimana untuk tiap *fold* terdiri dari data uji sebanyak 45 dan data latih sebanyak 405.Dapat dilihat

bahwa dengan nilai K=17 untuk metode KNN, menghasilakan ketepatan klasifikasi berupa rata-rata nilai *precision* sebesar 89.3%, *recall* sebesar 87.2%, dan nilai rata-rata akurasi se-besar 87.2%.

Tabel 4.13. Ketepatan Klasifikasi dengan KNN K=17

***Fold* ke-**	***Precision***	***Recall***	**Akurasi**
1	0,79	0,78	0,78
2	0,78	0,69	0,69
3	0,9	0,89	0,89
4	0,89	0,89	0,89
5	0,9	0,89	0,89
6	0,93	0,93	0,93
7	0,95	0,93	0,93
8	0,89	0,87	0,87
9	0,98	0,98	0,98
10	0,92	0,87	0,87
Rata-rata	**0,893**	**0,872**	**0,872**

Precision adalah rasio dari prediksi terkategori benar terhadap pada kelas emosi tersebut. Dari hasil pada Tabel 4.13 diketahui bahwa hasil rata-rata *precision* adalah sebesar 89.3%, artinya adalah sistem mampu mengklasifikasikan data dengan ketepatan hingga hasil prediksi kelas emosi sebesar 89.3%. *Recall* adalah rasio prediksi terkategori benar terhadap total prediksi dari kelas emosi awal. Hasil *recall* sebesar 87.2% menunjukkan metode mampu memprediksi benar kedalam kelas emosinya hingga 87.2%. Dan nilai akurasi 87.2% membuktikan dengan menggunakan KNN, dari 45 data uji terprediksi benar terdapat di dalam kelas emosinya adalah sebanyak 39 *tweet*, dan sisa 6 *tweet* lainnya terjadi *miss-classification*. *Confusion matrix* metode KNN ditampilkan pada Lampiran 7.

4.4 Perbandingan Antara SVM dan KNN

Setelah mendapatkan ketepatan klasifikasi dari masing-masing metode, selanjutnya adalah membandingkan hasil dari ke-

dua metode tersebut. Perbandingan metode SVM dan *KNN* pada penelitian ini mempertimbangkan hasil terbaik dari ketepatan klasifikasi terbaik dari setiap metode.

Tabel 4.14 Perbandingan Ketepatan Prediksi Klasifikasi

Metode	Precision	Recall	Akurasi
SVM	0,955	0,952	0,952
KNN	0,893	0,872	0,872

Ketepatan prediksi klasifikasi terbaik pada Tabel 4.14 menunjukkan perbandingan yang sangat jelas perbedaannya. Hal tersebut dikarenakan ketepatan prediksi klasifikasi menggunakan SVM baik dari *precision, recall,* maupun akurasi lebih besar daripada ketepatan prediksi klasifikasi menggunakan KNN.

4.5 Visualisasi *Word Cloud*

Visualisasi data teks menggunakan *word cloud* digunakan untuk mengetahui kata-kata yang paling sering muncul pada data. Pada penelitian ini, *word cloud* digunakan untuk visualisasi *tweet* berdasarkan kategori emosinya sehingga dapat diketahui kata-kata yang sering muncul pada setiap kelas emosi. Semakin besar ukuran *font* berarti semakin besar frekuensi kemunculan kata tersebut. Berikut merupakan word cloud data pada setiap media.

Gambar 4.5 *Wordcloud* Emosi Senang

Word cloud emosi senang pada Gambar 4.5 menunjukkan kata-kata yang sering muncul yaitu “perintah”, dari hasil *stemming* kata pemerintah, “periode”, “doa”, dan “dukung”. Kata-kata tersebut menunjukkan bahwa publik pengguna Twitter banyak yang sedang membahas mengenai dukungan kepada Presiden Jokowi untuk 2 periode. Publik pengguna *Twitter* banyak mendoakan Presiden Jokowi agar memenangkan pemilu tahun depan dan mendoakannya melalui *tweet* pada akun mereka.

Word cloud emosi sedih pada Gambar 4.6 menunjukkan kata-kata yang sering muncul yaitu “rakyat”, “ganti”, “dukung”, dan “dukung”.

Gambar 4.6 *Wordcloud* Emosi Sedih

Kata-kata tersebut menunjukkan bahwa publik pengguna Twitter banyak yang sedang membahas mengenai kesedihan rakyat akibat rezim Presiden Jokowi. Beberapa akun menyatakan bahwa Presiden Jokowi telah membodohi rakyat dengan pemikiran sesatnya. Ada juga yang menyatakan bahwa rak-yat Indonesia malu karena Presiden yang menjabat sampai di-buatkan lagu untuk segera lengser dari kedudukannya sebagai presiden. Untuk kata “dukung”, beda halnya dengan kata “dukung” pada emosi senang, arti “dukung” pada *Tweet* beremosi sedih menyatakan bahwa mereka mendukung tokoh-tokoh yang akan mencalonkan menjadi Presiden pada pemilu mendatang karena Lelah terhadap era

kepemimpinan Jokowi. Hal tersebut didukung de-ngan kata "ganti" yang berarti ganti Presiden yang banyak diung-kapkan rakyat melalui *tweet* pada akun mereka menggunakan tagar "#2019GantiPresiden".

Word cloud emosi marah pada Gambar 4.7 menunjukkan kata-kata yang sering muncul yaitu "nyinyir", "sikap", "temu", dan "paham".

Gambar 4.7 *Wordcloud* Emosi Marah

Kata "nyinyir" yang diungkapkan masyarakat pengguna *Twitter* adalah ungkapan ejekan antar pengguna *Twitter* lain yang mendukung Presiden Jokowi.Terdapat seorang tokoh politik yang bernama Fadli Zon dan Fahri Hamzah yang sering memberi kritik pedas terhadap Presiden Jokowi melalui akun *Twitter* mereka. Na-mun, terdapat suatu kejadian yang membuat kedua tokoh tersebut berhadapan dengan Presiden di Istana Negara, dan disana mereka memberi hormat kepada Presiden Jokowi. Hal tersebut membuat beberapa pendukung Presiden Jokowi menyindir, atau dalam is-tilah sekarang adalah "nyinyir" dan menghina kedua tokoh terse-but. Akibatnya, masyarakat yang kontra dengan Presiden dan pro dengan Fahri dan Fadli membuat pernyataan bahwa mereka yang "menyinyir" Fahri dan Fadli saat di Istana negara tidaklah paham mengenai arti negarawan dalam politik. Maksud dari pernyataan

tersebut adalah bahwa meskipun Fahri dan Fadli sering mengkritik sang presiden, bagi warga yang pro dengan mereka menganggap bahwa hormat di hadapan Presiden adalah hal yang wajar dan malah harus dilakukan. Karena hal tersebut merupakan hokum politik di negara. Jika tidak memberi hormat dan malah menantang di hadapan sang Presiden, jeruji besi lah yang akan menunggu mereka. Tweet tersebut banyak sekali di-*retweet* oleh kelompok kontra Presiden Jokowi. Hal tersebut bisa dilihat dari kata-kata yang bersinggungan dengan kejadian tersebut sepert "temu" yang berartti pertemuan di Istana negara, kata "paham" yang berarti bahwa yang "menyinyir" tidaklah paham.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut.

1. Klasifikasi yang didapat menggunakan SVM Kernel *Radial Basis Function* menghasilkan ketepatan prediksi klasi-fikasi lebih baik daripada menggunakan kernel Linear. Kernel RBF pada prediksi data *testing* mendapatkan rata-rata nilai *precision* sebesar 95.5%, rata-rata nilai *recall* sebesar 95.2%, dan nilai akurasi sebesar 95.2%.
2. Klasifikasi yang didapat menggunakan KNN k=17 pada prediksi data *testing* mendapatkan rata-rata nilai nilai *precision* sebesar 89.3%, rata-rata nilai *recall* sebesar 87.2%, dan nilai akurasi sebesar 87.2%.
3. Kata-kata yang sering muncul pada kelas emosi senang adalah "perintah", dari hasil stemming kata pemerintah, "periode", "doa", dan "dukung". Kata-kata tersebut menunjukkan bahwa publik pengguna Twitter banyak yang sedang membahas mengenai dukungan kepada Presiden Jokowi untuk 2 periode. Pada emosi sedih, menunjukkan kata-kata yang sering muncul yaitu "rakyat", "ganti", "dukung", dan "dukung". Kata-kata tersebut menunjukkan bahwa publik pengguna Twitter banyak yang sedang membahas mengenai kesedihan rakyat akibat rezim Presiden Jokowi. Sedangkan pada emosi menunjukkan kata yang sering muncul yaitu "nyinyir", "sikap", "temu", dan "paham" adalah ungkapkan masyarakat pengguna *Twitter* pendukung Fahri dan Fadli untuk menjelekkan pengguna *Twitter* lain.

5.2 Saran

Berdasarkan kesimpulan yang diperoleh, dapat dirumuskan saran sebagai pertimbangan penelitian selanjutnya adalah sebagai berikut.

1. Untuk Presiden Jokowi dan pihak yang terkait dalam membantunya, dapat melakukan klasifikasi emosi terhadap pengguna *Twitter* untuk mengevaluasi kinerjanya.
2. Untuk penelitian selanjutnya, penelitian serupa dapat dikembangkan dengan meninjau kembali tahap praproses, melihat kembali kata-kata yang harus dan tidak dihilangkan dalam kamus *stopwords*, karena hal tersebut sangat berpengaruh terhadap akurasi ketepatan prediksi.

DAFTAR PUSTAKA

Aliandu, P. (2013). Twitter Used by Indonesian President: An Sentiment Analysis of Timeline. *Information Systems International Conference*, 713-716.

Aman, S. &. (2007). Identifying Expressions of Emotion in Text. In Text. *Speech and Dialogue, Lencture Notes in Artificial Intelligence Vol 4629*, 196-205.

Ariadi, D. (2015). *Klasifikasi Berita Indonesia Menggunakan Naïve Bayes Classifier dan Support Vector Machine dengan Confix Stripping Stemmer.* Surabaya: Institut Teknologi Sepuluh Nopember.

Arifin, & E, P. K. (2012). Classification of Emotions in Indonesian Texts Using K-NN Method. *International Journal of Information and Electronics Engineering, Vol. 2, No. 6.*

Bata, J., Suyoto & Pranowo. (2015). Leksikon Untuk Deteksi Emosi dari Teks Berbahasa Indonesia. *Seminar Nasional Informatika 2015 (semnasIF 2015).* Yogyakarta: UPN Veteran Yogyakarta.

Bekkar, M. D. (2013). Evaluation Measure for Models Assesment over Imbalanced Data Sets. *Journal of Information Engineering and Aplications, 3*, 27-38.

Berry, M. W. (2010). *Text Mining Application and.* United Kingdom: WILEY.

Blanchette, J. (2008). *A Little Manual of API Design.* Oslo: Trolltech.

Buntoro, G. A. (2014). Sentiment Analysis Twitter dengan Kombinasi Lexicon Based dan Double Propagation. *Conference on Information Technology and Electrical Engineering*, (pp. 38-43).

Calvo, R. A. (2010). Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing 1(1)*, 18-37.

Castella, Q. &. (2014). *Word Storm : Multiples of Word Clouds for Visual Comparison of Documents.*

Djuroto, T. (2000). *Manajemen Penerbitan Pers.* Bandung: PT Remaja Rosda Karya.

Dragut, E. F. (2009). *Stop Word And Related Problem in Web Interface Integration.* VLDB Endowment.

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook "Advanced Approaches in Analyzing Unstructured Data".* Cambridge: University Press.

Gokgoz, E. &. (2015). Comparison of Decision Tree Algorithms for EMG Signal Classification Using DWT. *Biomedical Signal Processing and Control, 18*, 138-144.

Gunn, S. R. (1998). *Support Vector Machine for Classification and Regression.* Southampton: University of Southampton.

Hemalatha, I. V. (2012). Preprocessing The Informal Text for Efficient Sentiment Analysis. *International Journal of Emerging Trends & Technology in Computer Science, 1*, 58-61.

Hirat, R. &. (2015). A Survey On Emotion Detection Techniques using Text in Blogposts. *International Bulletin of Mathematical Research Vol 2, Issue 1*, 180-187.

Hotho, A. N. (2005). *A Brief Survey of Text Mining.* Kassel: University of Kassel.

Kemenkominfo. (2016). *Kominfo: Pengguna Internet di Indonesia 63 Juta Orang.* https://kominfo.go.id/index.php/. Diakses pada 3 Maert 2018

Larose, D. (2005). *Discovering Knowledge in Data.* USA: John Wiley's and Son.

Lopatovska, I. &. (2010). Theories, methods and current research on emotions in library and information science, information retrieval and human-computer interaction. *Information Processing and Management 47(4)*, 575-592.

Mohammad, S. &. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence 29(3)*, 301-326.

Mujilahwati, S. (2016). Pre-Processing Text Mining Pada Data. *Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA)*, 49-56.

Neviarouskaya, A. P. (2011). Affect Analysis Model: novel rule-based approach to affect sensing from text. *Natural Language Engineering 17*, 95-135.

Oatley, K. a. (1996). *Understanding Emotions.* Blackwell.

Poria, S. G. (2013). Fuzzy Clustering for Semi-supervised Learning – Case Study: Construction of an Emotion Lexicon. *MICAI 2012, Part I, LNAI 7629*, 73-86.

Power, M. a. (1997). *Cognition and Emotion.* LEA Press.

Pusat Bahasa. (2008). *Kamus Besar Bahasa Indonesia Edisi Keempat.* Jakarta: Gramedia Pustaka Utama.

Republik Indonesia. (1945). *Undang-Undang Dasar 1945 Pasal 22E.* Jakarta: Republik Indonesia.

Strapparava, C. &. (2004). WordNetAffect: an affective extension of WordNet. *Proc of the Conference on International Language Resources and Evaluation (LREC)*, (pp. 1083-1086).

Sun, Y. K. (2006). Boosting for Learning Multiple Classes with Im-balanced Class Distribution. Sixth International Conference on Data Mining (ICDM'06)., (pp. 421-431).

Tala, F. Z. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Amsterdam: M.S. thesis. M.Sc. Thesis. Master of Logic Project. Institute for Logic, Language and Computation. Universiteti van Amsterdam The Netherlands.

Twitter. (2016). *Twitter Support*. http:// support.twitter.com/. Diakses pada 20 Februari 2018

Weiss, S. M. (2010). *Text Mining: Predictive Methods for Analyzing*. New York: Springer.

Wicaksono, A. F. (2014). Automatically Building a Corpus for Sentiment Analysis on Indonesian Tweets. *Proc of the 28th Pacific Asia Conference on Language, Information and Computation*, 185-194.

Williams, G. (2011). *Data Mining with Ratle and R: The Art of Excavating Data for Knowledge Discovery.* New York: Springer

LAMPIRAN

Lampiran 1. ***Term Frequency*** **(TF)**

Tweet	aamin	abaikan	abis	agama	…	waspada	…	zionis
1	1	0	0	0	…	0	…	0
2	0	0	0	0	…	1	…	0
3	0	0	0	0	…	0	…	0
4	0	0	0	0	…	0	…	0
5	0	0	0	0	…	0	…	0
6	0	0	0	0	…	0	…	0
7	0	0	0	0	…	0	…	0
8	0	0	0	0	…	0	…	0
9	0	0	0	0	…	0	…	0
10	0	0	0	0	…	0	…	0
11	0	0	0	0	…	0	…	0
12	0	0	0	0	…	0	…	0
13	0	0	0	0	…	0	…	0
14	0	0	0	0	…	0	…	0
15	0	0	0	0	…	0	…	0
16	0	0	0	0	…	0	…	0
17	0	0	0	0	…	0	…	0
18	0	0	0	0	…	0	…	0
19	0	0	0	0	…	0	…	0
20	0	0	0	0	…	0	…	0
21	0	0	0	0	…	0	…	0
22	0	0	0	0	…	0	…	0
23	0	0	0	0	…	0	…	0
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…
444	0	0	0	0		0	…	
445	0	0	0	0		0	…	0
446	0	0	0	1		0	…	0
447	0	0	0	1		0	…	0
448	0	0	0	0		0	…	0
449	0	0	0	0		0	…	0
450	0	0	0	0		0	…	0

Lampiran 2. *Document Frequency* (DF)

Kata	DF	Kata	DF	Kata	DF
amin	1	laskar	1	…	…
abai	8	latar	1	…	…
abdi	1	lawan	11	ubah	2
abis	1	layan	4	udh	1
acara	1	lbh	6	uji	1
adem	2	lebar	1	ulama	7
adik	2	lebay	2	ummat	1
adlh	2	leceh	1	umpat	3
admin	10	legal	1	unskil	2
agama	1	lekat	1	untik	1
agenda	2	lelah	13	urus	3
agh	1	lemah	3	usung	2
agp	1	lembaga	1	video	1
agung	8	lengkap	2	viral	1
ahok	1	lengser	1	virus	1
ahoker	4	lg	1	wacana	1
aiiihhh	8	liat	2	wajah	1
air	2	liberal	1	wajar	2
ajak	2	listrik	2	wakil	3
ajar	6	lite	11	wang	1
akhlak	1	ludah	2	waras	1
akibat	11	luthfi	4	warga	4
aktor	1	maaf	2	warganet	5
akut	3	mafia	1	warteg	4
alas	2	mah	1	waspada	1
alat	1	mahal	3	web	14
alhamdulilah	1	maharani	1	wenang	1
alhamdulillah	1	mahasiswa	1	wes	2
all	8	mahfud	1	widodo	8
allah	1	mainstream	3	wkl	1
amanah	3	maju	1	wow	3
ambisi	1	makas	1	wujud	1
ambruk	2	makmur	5	yah	2
…	.	makna	1	yra	3
…	.	malu	1	zionis	3

Lampiran 3. ***Inverse Document Frequency*** **(IDF)**

Kata	IDF	Kata	IDF	Kata	IDF
amin	10,8322	laskar	6,109248	…	…
abai	6,109248	latar	6,109248	…	…
abdi	32,23845	lawan	40,82488	ubah	10,8322
abis	6,109248	layan	18,89181	udh	6,109248
acara	6,109248	lbh	25,90493	uji	3,912023
adem	6,109248	lebar	6,109248	ulama	29,14336
adik	8,999619	lebay	10,8322	ummat	6,109248
adlh	10,8322	leceh	6,109248	umpat	15,03191
admin	10,8322	legal	5,4161	unskil	10,8322
agama	38,06662	lekat	6,109248	untik	6,109248
agenda	6,109248	lelah	46,07588	urus	15,03191
agh	10,02127	lemah	15,03191	usung	10,8322
agp	6,109248	lembaga	6,109248	video	6,109248
agung	6,109248	lengkap	10,8322	viral	6,109248
ahok	33,3067	lengser	6,109248	virus	6,109248
ahoker	6,109248	lg	5,4161	wacana	6,109248
aiiihhh	18,89181	liat	10,8322	wajah	6,109248
air	32,23845	liberal	6,109248	wajar	10,8322
ajak	10,8322	listrik	10,8322	wakil	16,2483
ajar	9,445906	lite	39,86775	wang	6,109248
akhlak	25,90493	ludah	10,8322	waras	6,109248
akibat	6,109248	luthfi	18,89181	warga	15,64809
aktor	40,82488	maaf	10,8322	warganet	22,49905
akut	6,109248	mafia	6,109248	warteg	18,89181
alas	14,16886	mah	3,912023	waspada	6,109248
alat	10,8322	mahal	15,03191	web	48,58266
alhamdulilah	6,109248	maharani	6,109248	wenang	6,109248
alhamdulillah	6,109248	mahasiswa	6,109248	wes	10,8322
all	3,806662	mahfud	6,109248	widodo	31,29618
allah	31,29618	mainstream	15,03191	wkl	6,109248
amanah	6,109248	maju	6,109248	wow	15,03191
ambisi	15,03191	makas	4,163337	wujud	6,109248
ambruk	6,109248	makmur	22,49905	yah	10,02127
…	.	laskar	6,109248	yra	15,03191
…	.	latar	6,109248	zionis	15,03191

Lampiran 4. *Confusion Matrix* SVM Kernel *Linear*

***Confusion Matrix Fold* ke-1**			
	Senang	**Sedih**	**Marah**
Senang	7	1	1
Sedih	1	19	0
Marah	1	0	15
Total	9	20	16

***Confusion Matrix Fold* ke-2**			
	Senang	**Sedih**	**Marah**
Senang	16	1	0
Sedih	1	9	0
Marah	2	3	13
Total	19	13	13

***Confusion Matrix Fold* ke-3**			
	Senang	**Sedih**	**Marah**
Senang	19	0	0
Sedih	1	12	0
Marah	0	0	13
Total	20	12	13

***Confusion Matrix Fold* ke-4**			
	Senang	**Sedih**	**Marah**
Senang	15	0	0
Sedih	0	19	0
Marah	0	1	10
Total	15	20	10

***Confusion Matrix Fold* ke-5**			
	Senang	**Sedih**	**Marah**
Senang	17	0	0
Sedih	0	11	0
Marah	1	0	16
Total	18	11	16

Lampiran 4. *Confusion Matrix* SVM Kernel *Linear* (Lanjutan)

***Confusion Matrix Fold* ke-6**			
	Senang	**Sedih**	**Marah**
Senang	13	0	0
Sedih	0	16	0
Marah	0	2	14
Total	13	18	14

***Confusion Matrix Fold* ke-7**			
	Senang	**Sedih**	**Marah**
Senang	17	1	0
Sedih	1	12	1
Marah	0	0	13
Total	18	13	14

***Confusion Matrix Fold* ke-8**			
	Senang	**Sedih**	**Marah**
Senang	14	0	0
Sedih	2	17	0
Marah	0	0	12
Total	16	17	12

***Confusion Matrix Fold* ke-9**			
	Senang	**Sedih**	**Marah**
Senang	11	1	0
Sedih	0	18	0
Marah	0	0	15
Total	11	19	15

***Confusion Matrix Fold* ke-10**			
	Senang	**Sedih**	**Marah**
Senang	14	2	0
Sedih	0	10	0
Marah	0	0	19
Total	14	12	19

Lampiran 5. *Confusion Matrix* SVM Kernel RBF

***Confusion Matrix Fold* ke-1**			
	Senang	**Sedih**	**Marah**
Senang	7	1	1
Sedih	1	19	0
Marah	1	0	15
Total	9	20	15

***Confusion Matrix Fold* ke-2**			
	Senang	**Sedih**	**Marah**
Senang	16	1	0
Sedih	1	9	0
Marah	1	3	14
Total	18	13	14

***Confusion Matrix Fold* ke-3**			
	Senang	**Sedih**	**Marah**
Senang	19	0	0
Sedih	0	13	0
Marah	0	0	13
Total	19	13	13

***Confusion Matrix Fold* ke-4**			
	Senang	**Sedih**	**Marah**
Senang	15	0	0
Sedih	0	19	0
Marah	0	1	10
Total	15	29	10

***Confusion Matrix Fold* ke-5**			
	Senang	**Sedih**	**Marah**
Senang	17	0	0
Sedih	0	11	0
Marah	1	0	16
Total	18	11	16

Lampiran 5. *Confusion Matrix* SVM Kernel RBF (Lanjutan)

***Confusion Matrix Fold* ke-6**			
	Senang	**Sedih**	**Marah**
Senang	13	0	0
Sedih	0	16	0
Marah	0	2	14
Total	13	18	14

***Confusion Matrix Fold* ke-7**			
	Senang	**Sedih**	**Marah**
Senang	17	1	0
Sedih	1	12	1
Marah	0	0	13
Total	18	0	14

***Confusion Matrix Fold* ke-8**			
	Senang	**Sedih**	**Marah**
Senang	14	0	0
Sedih	2	16	1
Marah	0	0	12
Total	16	16	13

***Confusion Matrix Fold* ke-9**			
	Senang	**Sedih**	**Marah**
Senang	11	1	0
Sedih	0	18	0
Marah	0	0	15
Total	11	0	15

***Confusion Matrix Fold* ke-10**			
	Senang	**Sedih**	**Marah**
Senang	15	1	0
Sedih	0	10	0
Marah	0	0	19
Total	15	11	19

Lampiran 6. *Confusion Matrix* KNN

***Confusion Matrix Fold* ke-1**			
	Senang	**Sedih**	**Marah**
Senang	6	2	1
Sedih	3	16	1
Marah	2	1	13
Total	11	19	15

***Confusion Matrix Fold* ke-2**			
	Senang	**Sedih**	**Marah**
Senang	6	2	1
Sedih	3	16	1
Marah	2	1	13
Total	11	19	15

***Confusion Matrix Fold* ke-3**			
	Senang	**Sedih**	**Marah**
Senang	13	4	0
Sedih	1	8	1
Marah	1	7	10
Total	15	19	11

***Confusion Matrix Fold* ke-4**			
	Senang	**Sedih**	**Marah**
Senang	18	1	0
Sedih	2	11	0
Marah	1	1	11
Total	21	13	11

***Confusion Matrix Fold* ke-5**			
	Senang	**Sedih**	**Marah**
Senang	13	1	1
Sedih	0	18	1
Marah	1	1	9
Total	14	20	11

Lampiran 6. ***Confusion Matrix*** **KNN (Lanjutan)**

Confusion Matrix Fold **ke-6**			
	Senang	**Sedih**	**Marah**
Senang	17	0	0
Sedih	1	10	0
Marah	1	3	13
Total	19	13	13

Confusion Matrix Fold **ke-7**			
	Senang	**Sedih**	**Marah**
Senang	12	1	0
Sedih	0	15	1
Marah	1	0	15
Total	13	16	16

Confusion Matrix Fold **ke-8**			
	Senang	**Sedih**	**Marah**
Senang	17	1	0
Sedih	0	14	0
Marah	0	2	11
Total	17	17	11

Confusion Matrix Fold **ke-9**			
	Senang	**Sedih**	**Marah**
Senang	13	1	0
Sedih	1	18	0
Marah	0	4	8
Total	14	23	8

Confusion Matrix Fold **ke-10**			
	Senang	**Sedih**	**Marah**
Senang	12	0	0
Sedih	0	18	0
Marah	0	1	14
Total	12	19	14

Lampiran 7. Syntax Input Data Menggunakan Python 3.6

```
import pandas as pd
import numpy as np
import string
from string import punctuation
import nltk
from nltk.tokenize import wordpunct_tokenize
from nltk import word_tokenize
import re
import sys
import codecs
import collections
from collections import Counter
import matplotlib.pyplot as plt
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from sklearn.metrics import f1_score, precision_score,
  recall_score, accuracy_score
from sklearn.feature_extraction.text import CountVektorizer
from sklearn.feature_extraction.text import TFIDFTransformer
from sklearn.model_selection import KFold
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn import model_selection

df = pd.read_csv('E:/Data  Fix.csv',sep=';',encoding='latin-1')
df_train = df['Tweet'] #ambil kolom text
df_label = df['Emosi'] #ambil kolom class
df = pd.concat([df_train,df_label],axis=1)
stopword = open("E:/stopwords.txt","r").read()
```

Lampiran 8. *Syntax* Praproses Data Menggunakan Python 3.6

```
def preprocess_tweet(tweet):
    processed_tweet = []
    # Remove Punc
    tweet = re.sub(r'[^\w\s](\S+)','',tweet)
    # Remove \n
    tweet = re.sub(r"\\n", "", tweet)
    # Remove :
    tweet = re.sub(r":", "", tweet)
    # Remove %
    tweet = re.sub(r"%", "", tweet)
    # Remove URLs
    tweet = re.sub(r'((www\.[\S]+)|(https?://[\S]+))', '', tweet)
    # Remove Digit
    tweet = re.sub("\d+", "", tweet)
    # Remove @User_account
    tweet = re.sub(r"@\S+", "", tweet)
    # No URL
    tweet = re.sub(r"http\S+", "", tweet)
    # Remove #hashtag
    tweet = re.sub(r'#(\S+)', r' \1 ', tweet)
    # Remove RT (retweet)
    tweet = re.sub(r"RT(\S+)", "", tweet)
    # Replace 2+ dots with space
    tweet = re.sub(r'\.{2,}', ' ', tweet)
    # Strip space, " and ' from tweet
    tweet = tweet.strip(' "\'')
    # Replace multiple spaces with a single space
    tweet = re.sub(r'\s+', ' ', tweet)
    # Convert to lower case
    tweet = tweet.lower()
    return tweet
```

Lampiran 8. *Syntax* Praproses Data Menggunakan Python 3.6 (Lanjutan)

```
trainclean = []
for line in df_train:
    result = preprocess_tweet(line)
    trainclean.append(result)

factory = StemmerFactory()
stemmer = factory.create_stemmer()
train_stemmed = map(lambda x: stemmer.stem(x), trainclean)
train_no_punc = map(lambda x:
x.lower().translate(string.punctuation), train_stemmed)

trainfinal = []
for line in train_no_punc:
    word_token = nltk.word_tokenize(line)
    word_token = [word for word in word_token if not word in
stopword and not word[0].isdigit()]
    trainfinal.append(" ".join(word_token))

df1=pd.DataFrame(trainfinal)
df1.columns=['Tweet']
data = pd.concat([df1,df_label],axis=1)

data['Token'] = data.Tweet.str.strip().str.split('[\W_]+')
```

Lampiran 9. *Syntax* TFIDF Menggunakan Python 3.6

```
words = Counter()
for idx in data.index:
    words.update(data.loc[idx, "Token"])
words.most_common(20)

TF1 = (data['Tweet']).apply(lambda x: pd.value_counts(x.split("
  "))).sum(axis = 0).reset_index()
TF1.columns = ['Kata','TF']
for i,word in enumerate(TF1['Kata']):
    TF1.loc[i, 'IDF'] = np.log(train.shape[0]/(len(data
  [data['Tweet'].str.contains(word)])))
TF1['TFIDF'] = TF1['TF'] * TF1['IDF']
```

Lampiran 10. *Syntax* Akurasi Prediksi SVM Kernel *Linear* Menggunakan Python 3.6

```
kFoldCrossValidation = KFold(n_splits=10, random_state = 100,
shuffle=True)
for train_index, test_index in kFoldCrossValidation.split(x, b):
    x_train=x[train_index]
    x_test=x[test_index]
    y_train=b[train_index]
    y_test=b[test_index]
    sv = SVC(kernel='linear', C=1)
    svm = sv.fit(x_train, y_train)
    prediksi = svm.predict(x_test)
    akurasi.append(accuracy_score(y_test, prediksi))
    print("Akurasi Prediksi: ", accuracy_score(y_test, prediksi))
```

Lampiran 11. *Syntax* Akurasi Prediksi SVM Kernel RBF Menggunakan Python 3.6

```
kFoldCrossValidation = KFold(n_splits=10, random_state = 100,
shuffle=True)
for train_index, test_index in kFoldCrossValidation.split(x, b):
    x_train=x[train_index]
    x_test=x[test_index]
    y_train=b[train_index]
    y_test=b[test_index]
    sv = SVC(kernel='rbf', C=10, gamma=1)
    svm = sv.fit(x_train, y_train)
    prediksi = svm.predict(x_test)
    akurasi.append(accuracy_score(y_test, prediksi))
    print("Akurasi Prediksi: ", accuracy_score(y_test, prediksi))
```

Lampiran 12. *Syntax* Akurasi Prediksi KNN Menggunakan Python 3.6

```
kFoldCrossValidation = KFold(n_splits=10, random_state = 100,
shuffle=True)
for train_index, test_index in kFoldCrossValidation.split(x, b):
    x_train=x[train_index]
    x_test=x[test_index]
    y_train=b[train_index]
    y_test=b[test_index]
    sv = SVC(kernel='rbf', C=10, gamma=1)
    svm = sv.fit(x_train, y_train)
    prediksi = svm.predict(x_test)
    akurasi.append(accuracy_score(y_test, prediksi))
    print("Akurasi Prediksi: ", accuracy_score(y_test, prediksi))
```

BIODATA PENULIS

Penulis dengan nama lengkap Fazlur Rahman dilahirkan di Pekanbaru pada 5 April 1996. Penulis menempuh pendidikan formal di SDN Kutorenon 1, SMPN 1 Sukodono, dan SMAN 2 Lumajang. Penulis diterima sebagai Mahasiswa Departemen Statistika ITS melalui jalur SBMPTN pada tahun 2014. Pada tahun kedua perkuliahan, penulis mulai mengikuti organisasi dan kepanitiaan di dalam ITS sebagai *staff* Pengembangangan Sumber Daya Mahasiswa (PSDM) HIMASTA-ITS dan *staff* di Badan Eksekutif Mahassiswa Institut. Di PSDM HIMASTA-ITS penulis ditunjuk menjadi koordinator *Organizer Committee* Bina Cinta Statistika (BCS) 2015, yakni kaderisasi mahasiswa baru selama satu tahun. Di luar kampus, penulis ditunjuk menjadi *Staff Human Resource Development* (HRD) di organisasi sosial Duacare yang bertugas mengoordinir *volunteer*. Di tahun ketiga, penulis diamanahi menjadi *Head of* HRD Duacare serta menjadi *Steering Committee* pada BCS 2016. Selain itu penulis juga menjadi pengajar Teater anak-anak di kampung eks-dolly dalam Gerakan Melukis Harapan (GMH). Pada tahun keempat, penulis mulai megurangi kegiatan manajerial dan hanya menjadi *Head of* HRD di Duacare. Penulis mencoba fokus di bidang akademik dan mampu menjadi Juara II pada *Call for Paper* Universitas Budi Luhur tahun 2017. Penulis juga dapat meraih 2 Juara sekaligus pada Konferensi LOGIKA 2018 di Universitas Indonesia dengan meraih Juara I di bidang Ekonomi dan Juara III di bidang Industri. Penulis juga telah mengikuti bebrapa kegiatan *survey* sebagai pengaplikasian ilmu statistika. Segala kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini, dapat menghubungi penulis melalui email fazlur0504@gmail.com