TUGAS AKHIR – SS141501

IMPLEMENTASI *TEXT MINING* PADA ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP MEDIA *MAINSTREAM* MENGGUNAKAN *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE*

TAUFIK KURNIAWAN
NRP 1313 100 075

Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si.
Dr. Irhamah, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017

TUGAS AKHIR – SS141501

IMPLEMENTASI *TEXT MINING* PADA ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP MEDIA *MAINSTREAM* MENGGUNAKAN *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE*

TAUFIK KURNIAWAN
NRP 1313 100 075

Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si.
Dr. Irhamah, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017

FINAL PROJECT – SS 141501

TEXT MINING IMPLEMENTATION IN TWIITER USER SENTIMENT ANALYSIS OF MAINSTREAM MEDIA USING NAÏVE BAYES CLASSIFIER AND SUPPORT VECTOR MACHINE

TAUFIK KURNIAWAN
NRP 1313 100 075

Supervisors
Dr. Kartika Fithriasari, M.Si.
Dr. Irhamah, S.Si., M.Si.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017

LEMBAR PENGESAHAN

IMPLEMENTASI *TEXT MINING* PADA ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP MEDIA *MAINSTREAM* MENGGUNAKAN *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember

Oleh :
Taufik Kurniawan
NRP. 1313 100 075

Disetujui oleh Pembimbing Tugas Akhir:

Dr. Kartika Fithriasari, M.Si. ( )
NIP. 19691212 199303 2 001

Dr. Irhamah, S.Si., M.Si. ( )
NIP. 19780406 200112 2 002

Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2017

IMPLEMENTASI *TEXT MINING* PADA ANALISIS SENTIMEN PENGGUNA TWITTER MENGGUNAKAN *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE*

Nama : **Taufik Kurniawan**
NRP : **1313 100 075**
Departemen : **Statistika**
Pembimbing : **Dr. Kartika Fithriasari, M.Si.**
Dr. Irhamah, S.Si., M.Si.

Abstrak

Keresahan masyarakat terhadap pemberitaan media mainstream sempat menjadi trending topic di media sosial Twitter akibat ketidakpuasan terhadap media yang dinilai tidak representatif dan independen dalam memuat berita. Bahkan pemerintah membahas secara khusus terkait bahaya berita palsu yang sering beredar. Beberapa media mainstream yang fokus sebagai media berita dan banyak mendapat tanggapan masyarakat di media sosial adalah TV One, Metro TV, dan Kompas TV. Sehingga perlu dilakukan penelitian guna mengetahui bagaimana sentimen publik terhadap ketiga media tersebut, apakah mayoritas publik menilai positif atau negatif. Tanggapan publik mengenai media mainstream didapat dari Application Programming Interface (API) pada Twitter karena media sosial tersebut memiliki pengguna yang sangat banyak di Indonesia bahkan hingga mencapai 19,5 juta pengguna dari total 300 juta pengguna global. Pada penelitian ini, praproses teks yang digunakan adalah case folding, tokenizing, stopwords, dan stemming. Untuk praproses stemming digunakan algoritma confix-stripping stemmer Sedangkan pada analisis klasifikasi data teks tersebut digunakan metode Naïve Bayes Classifier dan Support Vector Machine. Klasifikasi menggunakan NBC pada data media TV One dan Kompas TV menghasilkan akurasi sebesar 95,6% dan 97,8%, sedangkan pada media Metro TV menghasilkan nilai G-mean dan AUC berturut-turut sebesar 81,3% and 82,36%. Klasifikasi menggunakan SVM pada data media TV One dan Kompas TV menghasilkan akurasi sebesar 97,9% dan 99,3%,, sedangkan pada media Metro TV menghasilkan nilai G-mean dan AUC berturut-turut sebesar 97,35% and 97,38%.

Kata Kunci : Media *mainstream*, *Naïve Bayes Classifier*, *Support Vector Machine*, *Text mining*, dan Twitter

(Halaman sengaja dikosongkan)

TEXT MINING IMPLEMENTATION ON TWITTER USER SENTIMENT ANALYSIS OF MAINSTREAM MEDIA USING NAÏVE BAYES CLASSIFIER AND SUPPORT VECTOR MACHINE

Name : **Taufik Kurniawan**
NRP : **1313 100 075**
Department : **Statistics**
Supervisors : **Dr. Kartika Fithriasari, M.Si.**
Dr. Irhamah, S.Si., M.Si.

Abstract

Public unrest on mainstream media had become a trending topic on Twitter. This situation happened due to the public dissatisfaction on the media that is not representative and independent in presenting news. Even the government discussed specifically related to the danger of false news that is often circulated. Some mainstream media that focus as news media and get a lot of public response in social media is TV One, Metro TV, and Kompas TV. So it is necessary to do research to find out how public sentiments to these media, whether the majority of public rate positive or negative. The public response to mainstream media is derived from the Application Programming Interface (API) on Twitter because the social media has a very large number of users in Indonesia even up to 19.5 million users out of a total of 300 million global users. In this research, the text preprocess used is case folding, tokenizing, stopwords, and stemming. For stemming process used confix-stripping stemmer algorithm, while in the text data classification analysis used Naïve Bayes Classifier and Support Vector Machine method. Classification using NBC on TV One and Kompas TV data resulted accuracy about 95,6% and 97,8%, whereas on Metro TV data yielded G-mean and AUC respectively about 81,3% and 82,36%. Classification using SVM on TV One and Kompas TV data resulted in accuracy about 97,9% and 99,3%, whereas in Metro TV data yielded G-mean and AUC value respectively about 97,35% and 97,38%.

Keywords: *Mainstream media, Naïve Bayes Classifier, Support Vector Machine, Text Mining, and Twitter.*

(Halaman sengaja dikosongkan)

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah, sehingga penulis bisa menyelesaikan Tugas Akhir yang berjudul **"Implementasi *Text Mining* pada Analisis Sentimen Pengguna Twitter Terhadap Media *Mainstream* Menggunakan *Naïve Bayes Classifier* dan *Support Vector Machine*"** yang merupakan salah satu syarat yang harus ditempuh untuk menyelesaikan pendidikan sarjana sesuai kurikulum di Departemen Statistika FMIPA-ITS dengan sebaik-baiknya dan tepat waktu.

Proses penyusunan laporan Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Bapak, Ibu dan keluarga penulis yang selalu memberikan doa, kasih sayang, dukungan serta bimbingannya.
2. Ibu Dr. Kartika Fithriasari, M.Si. dan Ibu Dr. Irhamah, S.Si., M.Si. selaku dosen pembimbing serta Ibu Dra Wiwiek S.W., M.S. dan Bapak Dr. Bambang Wijanarko Otok, S.Si., M.Si. selaku dosen penguji yang senantiasa memberi arahan, bimbingan, waktu, dan semangat sehingga Tugas Akhir ini dapat diselesaikan dengan baik
3. Mas M. Idrus Fachruddin, Mas Doni Rubiagatra dan teman-teman Surabaya Python yang senantiasa memberikan arahan dan bimbingan materi maupun teknis mengenai Tugas Akhir penulis.
4. Semua pihak yang sudah membantu dalam penyelesaian Tugas Akhir ini.

Penulis berharap laporan Tugas Akhir ini dapat memberikan manfaat bagi masyarakat dan mengembangkan ilmu pengetahuan.

Wassalamualaikum Wr. Wb.

Surabaya, Juli 2017

Penulis

(Halaman sengaja dikosongkan)

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

(Halaman sengaja dikosongkan)

DAFTAR GAMBAR

Halaman

(Halaman sengaja dikosongkan)

DAFTAR LAMPIRAN

Halaman

(Halaman 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 tipe tekstual yaitu fakta dan opini (Buntoro, Adji, & Purnamasari, 2014). Fakta merupakan suatu pernyataan obyektif mengenai entitas dan kejadian di dunia, sedangkan 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). Sejak 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 mengomentari 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 sekum-

pulan fungsi atau protokol yang disediakan untuk pengguna 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.

Akhir-akhir ini, timbul keresahan masyarakat yang sempat menjadi *trending topic* di Twitter mengenai ketidakpuasan masyarakat terhadap media yang dianggap kurang representatif dalam menyajikan berita di berbagai media *mainstream*. Bahkan pada tanggal 29 Desember 2016, Presiden Joko Widodo secara khusus menggelar rapat terbatas guna membahas bahaya informasi palsu yang banyak beredar di media sosial (Sa'diyah & Fadhilah, 2017). Beberapa media *mainstream* yang terfokus pada acara berita yaitu TV One, Metro TV, dan Kompas TV. Ketiga media tersebut juga sering mendapat berbagai tanggapan publik di media sosial. Sehingga perlu dilakukan penelitian untuk mengetahui tanggapan masyarakat berdasarkan sentimen di media sosial (twitter) terhadap ketiga media *mainstream* tersebut.

Analisis sentimen atau *opinion mining* merupakan metode analisis berbasis komputasi mengenai pendapat, sentimen, dan emosi (Liu, 2010). Analisis sentimen digunakan untuk melihat kecenderungan suatu sentimen atau pendapat, apakah pendapat tersebut cenderung beropini positif atau negatif. Sebelum melakukan analisis sentimen, diperlukan praproses teks dengan metode *text mining* untuk mengolah data teks agar siap untuk dianalisis. Praproses teks tersebut meliputi *case folding*, *tokenizing*, *stopwords*, dan *stemming*. *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. S*temming* adalah proses untuk mendapatkan kata dasar dengan menghilangkan imbuhan pada kata (Hemalatha, Varma, & Govardhan, 2012). Pada penelitian ini, proses *stemmer* menggunakan algoritma

Confix-Stripping Stemmer yang merupakan pengembangan dari algoritma *Nazief and Adriani's Stemmer*. Kedua algoritma tersebut merupakan algoritma yang dikembangkan untuk *stemming* berdasarkan aturan Bahasa Indonesia yaitu mendapatkan kata dasar dengan menghilangkan imbuhan kata meliputi awalan (*prefix*), sisipan *(infix*), akhiran (*suffixes*), dan kombinasi antara awalan dan akhiran (*confixes*). Sedangkan untuk klasifikasi sentimen digunakan metode klasifikasi teks.

Terdapat banyak metode klasifikasi dalam ilmu statistika yang digunakan untuk analisis sentimen namun metode yang sering digunakan dalam klasifikasi teks adalah metode *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM). Metode NBC telah banyak digunakan dalam penelitian mengenai *Text Mining* karena memiliki kelebihan yaitu algoritma sederhana tapi memiliki akurasi yang tinggi (Rish, 2006). Sedangkan metode SVM digunakan karena metode ini sangat cepat dan efektif pada klasifikasi data teks (Feldman & Sanger, 2007). Penelitian yang pernah dilakukan mengenai *sentiment analysis* adalah *Twitter Used by Indonesian President: An Sentiment Analysis of Timeline.* Penelitian tersebut membahas hasil analisis sentimen pada akun twiter milik Presiden Indonesia. Akurasi yang didapat dari penelitian tersebut sebesar 79,42%. Namun masih terdapat kekurangan dalam memisahkan data text yang tidak mengandung sentimen (Aliandu, 2013). Falahah dan Nur (2015) melakukan penelitian mengenai Pengembangan Aplikasi *Sentiment Analysis* Menggunakan Metode *Naïve Bayes*. Penelitian tersebut menggunakan praproses teks *case folding*, *parsing*, dan *transformasi* sehingga mendapatkan hasil akurasi klasifikasi 73%. Selain itu, penelitian dengan metode serupa pernah dilakukan Ariadi (2015) tentang Klasifikasi Berita Indonesia Menggunakan NBC dan SVM dengan *Confix Stripping Stemmer*. Penelitian tersebut menyimpulkan bahwa akurasi dengan NBC dan SVM berturut-turut sebesar 82,2% dan 88,1%. Penelitian lain dilakukan Widhianingsih (2016) yang berjudul Aplikasi *Text Mining* untuk Automasi Klasifikasi Artikel dalam Majalah Online Wanita Menggunakan *Naïve Bayes Classifier* (NBC) dan Artificial Neural Network (ANN). Penelitian tersebut mengklasifikasi teks artikel majalah

wanita. Tingkat akurasi model NBC sebesar 80,71%, model ANN sebesar 75%, dan model Regresi Logistik Multinomial sebesar 57,86%.

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 sentimen *tweet* (positif dan negatif). Penelitian ini bertujuan melakukan analisis sentimen mengenai tanggapan masyarakat terhadap acara pada media televisi berbasis *Text Mining* data twitter. Melalui penelitian ini diharapkan dapat memberikan saran kepada penyedia stasiun televisi terkait dalam menyajikan acara di televisi serta kepada masyarakat sebagai pertimbangan dalam menentukan stasiun televisi yang akan dilihat.

1.2 Rumusan Permasalahan

Mengetahui tanggapan dari masyarakat terhadap layanan media merupakan hal yang tidak bisa dikesampingkan. Sehingga akurasi klasifikasi yang tinggi pada analisis sentimen dengan metode klasifikasi yang digunakan menjadi suatu yang penting. NBC merupakan metode klasifikasi yang mengacu pada teorema probabilitas bersyarat. NBC memiliki algoritma yang sederhana namun mempunyai akurasi yang tinggi. Sedangkan SVM adalah metode klasifikasi dengan mencari nilai pemisah antar kategori yang optimum atau *optimum separating hyperplane*. Metode SVM mempunyai akurasi yang tinggi untuk klasifikasi data teks. Kedua metode tersebut akan dibandingkan mana metode yang menghasilkan *error* yang paling kecil. Berdasarkan penjelasan tersebut, maka permasalahan utama yang akan dibahas dalam penelitian ini adalah berapa tingkat ketepatan klasifikasi sentimen pengguna twitter terhadap media *mainstream* TV One, Metro TV, dan Kompas TV menggunakan metode Naïve Bayes Classifier dan Support Vector Machine serta apa kata yang sering muncul berdasarkan masing-masing sentimen?

1.3 Tujuan

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

1. Mendapatkan hasil ketepatan klasifikasi sentimen pengguna twitter terhadap media *mainstream* TV One, Metro TV, dan Kompas TV menggunakan metode *Naïve Bayes Classifier*.
2. Mendapatkan hasil ketepatan klasifikasi sentimen pengguna twitter terhadap media *mainstream* TV One, Metro TV, dan Kompas TV menggunakan metode *Support Vector Machine*.
3. Membandingkan hasil ketepatan klasifikasi metode *Naïve Bayes Classifier* dengan metode *Support Vector Machine*
4. Mendapatkan kata-kata yang sering muncul berdasarkan masing-masing sentimen menggunakan visualisasi word cloud.

1.4 Manfaat Penelitian

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

1. Membantu pihak penyedia media *mainstream* yang bersangkutan dalam memahami tanggapan masyarakat mengenai program acara yang dijalankan serta mempercepat proses klasifikasi tanggapan masyarakat (sentimen) karena telah mendapatkan model dari data *training* sehingga pihak penyedia media *mainstream* tidak perlu mengklasifikasi ulang data secara manual.
2. Memberikan tambahan informasi kepada publik terhadap media *mainstream* pilihan melalui hasil klasifikasi sentimen.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Penelitian menggunakan studi kasus media *mainstream* TV One, Metro TV, dan Kompas TV.
2. Penelitian tidak memperhatikan latar belakang atau demografi dari pemilik akun twitter.
3. Klasifikasi sentimen data awal ditentukan secara subyektif peneliti.
4. Data twitter yang yang digunakan merupakan tweet yang diunggah pada tanggal 12 Februari hingga 2 April 2017. Sehingga substansi *tweet* dapat dipengaruhi isu yang sedang banyak diba-

has pada waktu tersebut seperti isu politik, isu Pilkada, dan isu SARA (Suku, agama, ras, dan antar golongan).

BAB II
TINJAUAN PUSTAKA

2.1 *Sentiment Analysis*

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *Text Mining* yang bertujuan menganalisis pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang pembicara atau penulis berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu lainnya (Liu, 2010). Tugas dasar dalam analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudia menentukan pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah bersifat positif atau negatif. *Sentiment analysis* juga dapat menyatakan perasaan emosional sedih, gembira, atau marah. Kita dapat mencari pendapat tentang produk-produk, merek atau orang-orang dan menentukan apakah mereka dilihat positif atau negatif di *web*.

Ekspresi atau *sentiment* mengacu pada fokus topik tertentu, pernyataan pada satu topik mungkin akan berbeda makna dengan pernyataan yang sama pada *subject* yang berbeda. Oleh karena itu pada beberapa penelitian, pekerjaan didahului dengan menentukan elemen dari sebuah produk yang sedang dibicarakan sebelum memulai proses *opinion mining*.

2.2 *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 mengekstraksi 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 & Kogan, 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*.

2.3 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. Praproses dalam *text mining* cukup rumit karena dalam Bahasa Indonesia terdapat berbagai aturan penulisan kalimat maupun pembentukan kata berimbuhan. Terdapat empat aturan pemben-tukan kata berimbuhan (afiks) untuk merubah makna kata dasar yaitu sebagai berikut.

a. Awalan (prefiks), imbuhan yang dapat ditambahkan pada awal kata dasar. Imbuhan ini ter-bagi dalam dua jenis.
   - Standar, yang mencakup imbuhan ‘di-‘, ‘ke-‘, dan ‘se-‘.
   - Kompleks, yang mencakup imbuhan ‘me-‘, ‘be-‘, ‘pe-‘, dan ‘te-‘.

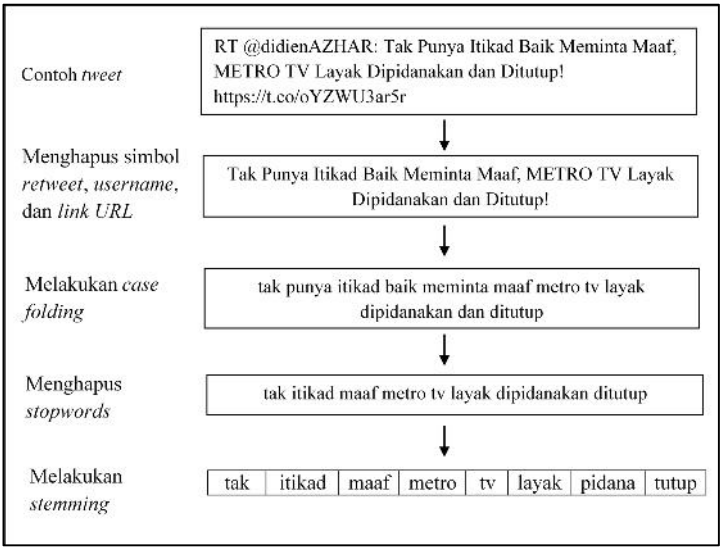
   Perbedaan atara kedua jenis imbuhan awalan tersebut yaitu penambahan imbuhan awalan standar pada suatu kata dasar tidak merubah kata dasar tersebut, sedangkan imbuhan awalan kompleks pada suatu kata dasar dapat merubah struktur kata dasar tersebut.
b. Akhiran (sufiks), imbuhan yang ditambahkan di belakang kata dasar. Sufiks yang sering digunakan yaitu ‘-i’, ‘-kan’, dan ‘-an’. Selain itu, imbuhan kata yang menunjukkan keterangan kepemilikan (‘-ku’, ‘-mu’, dan ‘-nya’) dan partikel (‘-lah’, ‘-kah’, ‘-tah’, dan ‘-pun’) juga dapat dikategorikan sebagai sufiks.
c. Awalan dan akhiran (konfiks), imbuhan yang ditambahkan di depan dan belakang kata dasar (prefiks dan sufiks) secara bersama-sama.

d. Sisipan (infiks), imbuhan yang ditambahkan di tengah kata dasar.

Aturan pembentukan kata dalam Bahasa Indonesia berkaitan dengan praproses teks karena hasil akhir praproses teks diharapkan mendapatkan kata dasar yang sesuai dengan Kamus Besar Bahasa Indonesia. Tahapan dalam praproses teks adalah sebagai berikut.

a. *Case Folding*, merupakan proses untuk mengubah semua karakter teks menjadi huruf kecil serta menghilangkan tanda baca dan angka. Cara kerja *case folding* adalah memproses huruf alphabet dari "a" hingga "z" saja sehingga karakter selain huruf tersebut akan dihapus (Weiss, 2010).
b. *Tokenizing*, merupakan proses memecah yang semula kalimat menjadi kata-kata atau memutus urutan string menjadi potongan-potongan, seperti kata-kata berdasarkan tiap kata yang menyusunnya.
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, Fang, Sitla, Yu, & Meng, 2009). Kosakata yang dimaksud yaitu seperti kata penghubung dan kata keterangan yang bukan merupakan kata unik, misalnya "dari", "akan", "seorang", dan sebagainya.
d. *Stemming*, yakni proses untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran). Pada penelitian ini algoritma *stemming* yang digunakan adalah *Confix Striping Stemmer* yang merupakan pengembangan dari algoritma *Nazief and Adriani's Stemmer*.
e. *Removal URL*, yaitu proses menghapus URL atau alamat *website* yang ada pada *tweet* (Mujilahwati, 2016).

Penjelasan mengenai hasil dari setiap tahap praproses teks akan dijabarkan pada simulasi praproses teks pada sebuah data *tweet*. *Tweet* yang akan digunakan sebagai contoh adalah *tweet* "RT @didienAZHAR: Tak Punya Itikad Baik Meminta Maaf, METRO TV Layak Dipidanakan dan Ditutup! https://t.co/oYZ WU3ar5r"



Gambar 2.1 Simulasi Praproses Teks

Untuk *tweet* berikutnya, seperti *tweet* "Pdhl kompas tv netral.. https://t.co/0skFwltm4Z" akan dilakukan praproses teks dengan langkah-langkah yang sama sehingga menghasilkan hasil praproses terakhir sebagai berikut.

kompas	tv	netral	ya

Gambar 2.2 Contoh Hasil Praproses Teks

Dari kedua contoh hasil praproses teks pada *tweet* diatas, maka didapat struktur data setelah praproses teks sebagai berikut.

Tabel 2.1 Contoh Struktur Data Setelah Praproses Teks

Tweet ke	Variabel Prediktor										
	tak	itikad	maaf	metro	tv	layak	pidana	tutup	kompas	netral	ya
1	1	1	1	1	1	1	1	1	0	0	0
2	0	0	0	0	1	0	0	0	1	1	1

Pembentukan struktur data setelah dilakukan praproses teks seperti pada Tabel 2.1, yaitu menjadikan setiap kata menjadi variabel prediktor dan meletakkannya pada satu baris. Jika terdapat

tambahan kata (variabel prediktor) dari *tweet* baru, maka kata tersebut diletakkan pada baris yang sama dan di kolom berikutnya. Namun jika terdapat kata yang sama atau kata yang telah ada pada struktur data, maka kata tersebut tidak dimasukkan lagi pada struktur data. Sehingga tidak terdapat kata atau variabel prediktor yang sama dalam struktur data. Nilai dari setiap kata tersebut merupakan jumlah kemunculan kata dalam *tweet* ke-i seperti yang terdapat pada Tabel 2.1 mengenai contoh struktur data setelah praproses teks.

2.4 *Nazief and Adriani's Stemmer*

Algoritma *stemming* dengan Bahasa Indonesia telah dikembangkan sejak tahun 1996 oleh Nazief dan Adriani yang kemudian dikenal dengan *Nazief and Adriani's Stemmer*. Algoritma ini dikembangkan untuk mendapatkan kata dasar dengan menghilangkan imbuhan kata berdasarkan aturan pada Bahasa Indonesia yakni imbuhan awalan (prefiks), akhiran (sufiks), awalan dan akhiran (konfiks), dan sisipan (infiks) seperti yang telah dijelaskan pada subbab 2.3. Algoritma ini juga dapat digunakan untuk *recoding*, sebuah pendekatan untuk me-ngembalikan huruf awal kata yang terhapus akibat penghilangan prefiks. Selain itu, algoritma ini juga menggunakan daftar kata dasar yang dipakai pada tahap pemeriksaan ketika proses *stemming* telah menemukan kata dasar yang diduga. Pengelompokan beberapa kategori aturan imbuhan dimodelkan pada algoritma *nazief and adriani's Stemmer* sebagai berikut.

$$\left[\left[\left[\mathrm{AW}+\right]\mathrm{AW}+\right]\mathrm{AW}+\right]\ \text{Kata dasar}\ \left[\left[+\mathrm{AK}\right]\left[+\mathrm{KK}\right]\left[+\mathrm{P}\right]\right]$$

dimana:

AW	= Awalan	KK	= Kata ganti kepunyaan
AK	= Akhiran	P	= Partikel

Langkah-langkah *nazief and adriani's stemmer* adalah sebagai berikut (Asian, 2007).

1. Kata yang belum dilakukan proses *stemming*, dicari pada kamus kata dasar. Jika ditemukan, maka kata tersebut dianggap sebagai kata dasar dan proses berhenti. Jika tidak ditemukan, maka dilanjutkan pada tahap kedua.

2. Menghilangkan *inflection particle* (‘-lah’, ‘-kah’, ‘-tah’, ‘-pun’) dan dilanjutkan menghilangkan *passive pronoun* (‘-ku’, ‘-mu’, ‘-nya’).
3. Menghilangkan *derivation suffixes* (‘-i’, ‘-kan’, ‘-an’).
4. Menghilangkan *derivation prefixes* (‘di-‘, ‘ke-‘, ‘se-‘, ‘me-‘, ‘be-‘, ‘te-‘, ‘pe-‘) dengan iterasi maksimum tiga kali dengan langkah-langkah sebagai berikut.
   a. Langkah pertama berhenti jika:
      - Terjadi kombinasi awalan dan akhiran terlarang seperti pada Tabel 2.2.
      - Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
      - Tiga awalan telah dihilangkan.

Tabel 2.2 Kombinasi Awalan dan Akhiran yang Dilarang

Awalan	Akhiran yang Tidak Diperbolehkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-ans

   b. Identifikasi tipe awalan kemudian hilangkan. Awalan terbagi menjadi dua tipe sebagai berikut.
      - Standar (‘di-‘, ‘ke-‘, ‘se-‘) merupakan awalan yang dapat dihilangkan langsing dari kata.
      - Kompleks (‘me-‘, ‘be-‘, ‘pe-‘, ‘te-‘) merupakan awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya.
   c. Mencari kata yang telah dihilangkan awalannya dalam kamus kata dasar. Apabila tidak ditemukan, maka seluruh tahap dihentikan.
5. Jika kata dasar belum ditemukan, maka proses selanjutnya adalah *recoding*. *Recoding* dilakukan dengan menambah atau mengganti huruf awal kata yang terpenggal proses *stemming*. Contoh, kata ‘menangkis’ dimana setelah dihilangkan awalan

‘me-‘ menjadi ‘nangkis’. Kata ‘nangkis’ tidak terdapat pada kamus kata dasar sehingga dilakukan *recoding* dengan mengganti karakter ‘n’ menjadi ‘t’ dan didapat kata dasar ‘tangkis’.

6. Jika semua langkah gagal, maka input kata dianggap sebagai kata dasar.

2.5 *Confix Stripping Stemmer*

Algoritma *Confix-Stripping Stemmer* merupakan pengembangan dari algoritma *Nazief and Adriani’s Stemmer*. Algoritma ini dikembangkan dengan perbaikan algoritma menyesuaikan kaidah Bahasa Indonesia dengan tujuan untuk meningkatkan hasil *stemming* yang diperoleh. Perbaikan dalam algoritma *Confix-Stripping Stemmer* ini adalah sebagai berikut.

1. Kamus kata dasar yang digunakan lebih lengkap.
2. Modifikasi dan penambahan aturan pemenggalan untuk tipe awalan yang kompleks.
3. Penambahan aturan *stemming* untuk kata ulang dan bentuk jamak.
   Contohnya kata ‘kemerah-merahan’ menjadi kata ‘merah’. Algoritma ini bekerja dengan melakukan pemisahan kata tersebut menjadi dua kata yang masing-masing di-*stemming*.
4. Mengubah urutan *stemming* untuk beberapa kasus tertentu.
   Pada algoritma *Nazief and Adriani’s Stemmer*, penghilangan imbuhan dilakukan dari menghilangkan akhiran terlebih dahulu kemudian baru menghilangkan awalan. Sedangkan pada algoritma *Confix-Stripping Stemmer*, terdapat kasus dimana penghilangan imbuhan dimulai dari awalan terlebih dahulu kemudian diikuti penghilangan akhiran yang disebut *rule precedence*. Aturan ini berlaku jika terdapat kombinasi awalan dan akhiran ‘be-lah’, ‘be-an’, ‘me-i’, ‘di-i’, ‘pe-i’, atau ‘te-i’. Contohnya ‘berterbangan’, ‘memiliki’, ‘ditangisi’, dan ‘terampuni’.

2.6 Naïve Bayes Classifier

Teorema Bayes merupakan teorema yang mengacu konsep probabilitas bersyarat (Siang, 2005). Secara umum teorema Bayes dapat dinotasikan pada persamaan berikut.

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)} \quad (2.1)$$

Algoritma *Naïve Bayes Classifier* (NBC) merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Feldman & Sanger, 2007). Metode *Naive Bayes Classification* merupakan salah satu metode yang dapat mengklasifikasikan teks. Kelebihan NBC adalah algoritmanya sederhana tetapi memiliki akurasi yang tinggi. Terdapat dua tahap dalam klasifikasi *tweet*. Tahap pertama adalah pelatihan terhadap *tweet* yang telah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi *tweet* yang belum diketahui kategorinya (Falahah & Nur, 2015). Dalam algoritma NBC setiap dokumen direpresentasikan dengan pasangan atribut "a_1, a_3,..., a_n" dimana a_1 adalah kata pertama, a_2 adalah kata kedua dan seterusnya. Sedangkan V adalah himpunan kategori *tweet*. Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan (V_{MAP}). Adapun persamaan V_{MAP} adalah sebagai berikut.

$$V_{MAP} = \arg\max_{v_j = V} P(v_j) \prod_i P(a_i \mid v_j) \quad (2.2)$$

Nilai P(vj) dihitung pada saat *training*, didapat dengan rumus sebagai berikut:

$$P(v_j) = \frac{|doc\ j|}{|training|} \quad (2.3)$$

dimana |*doc j*| merupakan jumlah *tweet* yang memiliki kategori *j* dalam *training*. Sedangkan |*training*| merupakan jumlah *tweet* dalam contoh yang digunakan untuk *training*. Untuk setiap probabilitas kata a_i untuk setiap kategori $P(a_i|v_j)$, dihitung pada saat *training*.

$$P(a_i \mid v_j) = \frac{n_i + 1}{|n + kosakata|} \quad (2.4)$$

dimana n_i adalah jumlah kemunculan kata a_i dalam *tweet* yang berkategori v_j, sedangkan n adalah banyaknya seluruh kata dalam *tweet* dengan kategori v_j dan |kosakata| adalah banyaknya kata dalam data *training*.

Berikut merupakan ilustrasi perhitungan untuk melakukan klasifikasi sentimen menggunakan metode *Naïve Bayes Classifier*

dengan contoh *tweet* pada Tabel 2.1 sebagai data *training* dan contoh *tweet* "Maaf menurut saya Kompas tv netral" sebagai data *testing*. Per-hitungan dilakukan untuk mengklasifikasikan apakah contoh *tweet* sebagai data *testing* tersebut memiliki sentimen positif atau negatif. Hal pertama yang dilakukan adalah meng-hitung probabilitas setiap kelas sentimen dengan persamaan (2.3) sebagai berikut.

$$P(v_1) = \frac{|doc1|}{|training|} = \frac{1}{2} = 0,5$$

$$P(v_2) = \frac{|doc2|}{|training|} = \frac{1}{2} = 0,5$$

dimana $P(v_1)$ adalah probabilitas sentimen positif dan $P(v_2)$ adalah probabilitas sentimen negatif. Kemudian dilakukan perhitungan probabilitas kemunculan setiap kata pada masing-masing kategori dengan persamaan (2.4).

P(tak | v_1) = (1+1)/(8/11)=0,1053
P(tak | v_2) = (0+1)/(4/11)=0,0667
P(itikad | v_1) = (1+1)/(8/11)=0,1053
P(itikad | v_2) = (0+1)/(4/11)=0,0667
P(maaf | v_1) = (1+1)/(8/11)=0,1053
P(maaf | v_2) = (0+1)/(4/11)=0,0667
P(metro | v_1) = (1+1)/(8/11)=0,1053
P(metro | v_2) = (0+1)/(4/11)=0,0667
P(tv | v_1) = (1+1)/(8/11)=0,1053
P(tv | v_2) = (1+1)/(4/11)=0,1333
P(layak | v_1) = (1+1)/(8/11)=0,1053
P(layak | v_2) = (0+1)/(4/11)=0,0667
P(pidana | v_1) = (1+1)/(8/11)=0,1053
P(pidana | v_2) = (0+1)/(4/11)=0,0667
P(tutup | v_1) = (1+1)/(8/11)=0,1053
P(tutup | v_2) = (0+1)/(4/11)=0,0667
P(kompas | v_1) = (0+1)/(8/11)=0,0526
P(kompas | v_2) = (1+1)/(4/11)=0,1333
P(netral| v_1) = (0+1)/(8/11)=0,0526
P(netral | v_2) = (1+1)/(4/11)=0,1333
P(ya | v_1) = (0+1)/(8/11)=0,0526

$P(\text{ya} \mid v_2)$ = (1+1)/(4/11)=0,1333

Selanjutnya adalah mencari probabilitas tertinggi dari *tweet* yang diujikan. *Tweet testing* setelah dilakukan praproses teks, maka terdiri dari kata “maaf”, “kompas”, “tv”, dan “netral”. Sehingga dicari probabilitas tertinggi dari setiap kata pada *tweet* tersebut menggunakan persamaan (2.2).

$$P(v_1)\prod_i P(a_i \mid v_1) = (0,5)\left(P(\text{maaf} \mid \text{v}_1)\times P(\text{kompas} \mid \text{v}_1)\times P(\text{tv} \mid \text{v}_1)\times P(\text{netral} \mid \text{v}_1)\right)$$
$$= (0,5)(0,1053\times 0,0526\times 0,1053\times 0,0526)$$
$$= 0,000015339$$

$$P(v_2)\prod_i P(a_i \mid v_2) = (0,5)\left(P(\text{maaf} \mid \text{v}_2)\times P(\text{kompas} \mid \text{v}_2)\times P(\text{tv} \mid \text{v}_2)\times P(\text{netral} \mid \text{v}_2)\right)$$
$$= (0,5)(0,0667\times 0,1333\times 0,1333\times 0,1333)$$
$$= 0,000078993$$

$$V_{MAP} = \arg\max_{v_j = V} P(v_j)\prod\nolimits_i P(a_i \mid v_j) = v_2$$

Nilai probabilitas kata setiap *tweet testing* yang terbesar adalah probabilitas setiap kata pada sentimen negatif sehingga *tweet testing* tersebut diklasifikasikan sebagai *tweet* dengan sentimen negatif.

2.7 *Term Frequency Inverse Document Frequency*

Term Frequency Inverse Document Frequency (TF-IDF) merupakan sebuah metode pembobotan yang dilakukan untuk ekstraksi data teks. Tujuan dari TF-IDF adalah untuk menemukan jumlah kata yang diketahui (tf) setelah dikalikan dengan beberapa banyak tweet dimana suatu kata tersebut muncul (idf). Metode TF-IDF dilakukan dengan menghitung bobot dengan cara integrasi antara *term frequency* (tf) dan *inverse document frequency* (idf). Berikut merupakan rumus untuk menemukan pembobot dengan TF-IDF.

$$w_{ij} = tf_{ij} \times idf$$
$$idf = \log\left(\frac{N}{df_j}\right) \qquad (2.5)$$

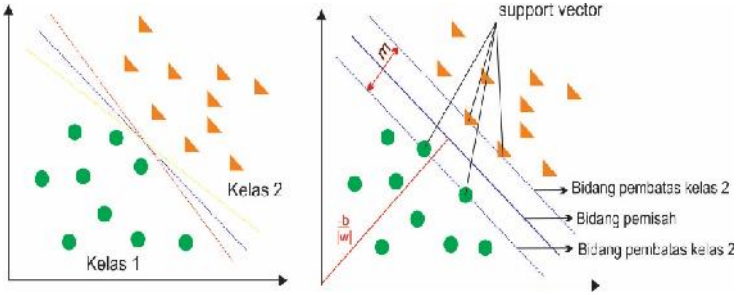
dimana w_{ij} adalah bobot dari kata *i* pada artikel ke *j*, N merupakan jumlah seluruh *tweet*, *tf* adalah jumlah kemunculan kata *i* pada tweet *j*, dan *df* adalah jumlah tweet *j* yang mengandung kata *i*. TF-IDF dilakukan agar data dapat dianalisis dengan menggunakan *Support Vector Machine*.

2.8 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.

2.8.1 SVM pada *Linearly Separable Data*

SVM pada *linearly separable data* adalah penerapan metode SVM pada data yang dapat dipisahkan secara linier. Misalkan $x_i = \{x_i, x_{i+1},\ldots,x_n\}$ adalah *dataset* dan $y_i = \{+1, -1\}$ adalah label kategori untuk *dataset*. Penggambaran *linearly separable data* dapat dilihat pada Gambar 2.3 berikut.



Gambar 2.3 Alternatif Bidang Pemisah (kiri) dan Bidang Pemisah Terbaik dengan Margin (m) Terbesar (kanan)

Pada Gambar 2.3, kedua kelas data dapat dipisahkan oleh sepasang bidang pembatas yang sejajar (linier). Data yang berada

pada bidang pembatas disebut dengan *support vector*. |b|/||w|| merupakan jarak bidang pemisah yang tegak lurus dari titik pusat koordinat dan ||w|| adalah jarak *Euclidean* dari w. Bidang pembatas pertama membatasi kelas pertama, sedangkan bidang kedua membatasi kelas kedua. Persamaan *hyperplane* dapat ditulis sebagai berikut.

$$x_i w + b = 0 \qquad (2.6)$$

w adalah normal bidang dan *b* adalah suatu konstanta yang biasa disebut bias. Nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) yaitu:

$$\frac{2}{|| w ||} \qquad (2.7)$$

Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan (2.6). Dengan mengalikan *b* dan *w* dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstanta yang sama (Gunn, 1998). Oleh karena itu, *constraint* pada persamaan (2.6) merupakan *scaling constraint* yang dapat dipenuhi dengan *rescaling b* dan *w*. Selain itu karena memaksimalkan 1/||w|| sama dengan meminimumkan $||w||^2$. Jika kedua bidang pembatas direpresentasikan dalam pertidaksamaan, maka akan menjadi seba-gai berikut.

$$y_i (x_i w + b) - 1 \geq 0 \qquad (2.7)$$

maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu:

$$\min \frac{1}{2} || w ||^2 \qquad (2.8)$$

dengan $y_i (x_i w + b) - 1 \geq 0$ dan fungsi batasan sebagai berikut.

$$\sum_{m=1}^{M} \sim_m \left[y_m \left(W.X_m + b \right) - 1 \right] \qquad (2.9)$$

Kemudian menggunakan *Lagrange Multiplier* didapatkan persamaan berikut.

$$L_d = \sum_{m=1}^{M} \sim_m - \frac{1}{2} \sum_{m_1=1}^{M} \sum_{m_2=1}^{M} \sim_{m_1} \sim_{m_2} y_{m_1} y_{m_2} \left(X_{m_1}^T X_{m_2} \right) \qquad (2.10)$$

Persamaan L_d didapat dengan mensubtitusikan nilai y_{m1}, y_{m2}, X_{m1}, dan X_{m2} ke persamaan (2.10). Persamaan L_d digunakan untuk mencari nilai-nilai α_m (*support vector*) dengan membuat L_d optimum. L_d optimum didapat dengan cara mencari turunan parsial L_d terhadap α. Setelah mendapatkan nilai α, langkah selanjutnya adalah mencari nilai ω dan b dengan persamaan sebagai berikut.

$$w = \sum_{i=1}^{I} \sim_i y_i x_i \quad \text{dan} \quad b = 1 - w^T x \qquad (2.11)$$

Berikut disajikan ilustrasi perhitungan dalam mengklasifikasi data *tweet* menggunakan metode *Support Vector Machine*. Ilustrasi menggunakan contoh data hasil praproses pada Tabel 2.1 yang telah diberi pembobotan dengan *term frequency-inverse document frequency* dengan hasil sebagai berikut.

Tabel 2.3 Ilustrasi Pembobotan TF-IDF

Simbol	Kata	Bobot	Sentimen
x11	Tak	0.30103	-1
x21	Itikad	0.30103	-1
x31	Maaf	0.30103	-1
x41	Metro	0.30103	-1
x51	Tv	0.30103	-1
x61	Layak	0.30103	-1
x71	Pidana	0.30103	-1
x81	Tutup	0.30103	-1
x12	kompas	0.30103	1
x22	Tv	0.30103	1
x32	Netral	0.30103	1
x42	Ya	0.30103	1

Setelah didapatkan nilai pembobot masing-masing kata pada setiap kategori *tweet*, kemudian mensubtitusi nilai bobot sebagai variabel *x* dan nilai sentimen sebagai variabel *y* pada persamaan (2.10) sehingga akan didapat persamaan L_d. Persamaan tersebut digunakan untuk mendapatkan α. Kemudian dilanjutkan pada persamaan (2.11) untuk mendapatkan nilai *w* dan nilai b. Nilai-nilai

tersebut selanjutnya akan digunakan untuk membangun persamaan *hyperplane*.

2.8.2 SVM pada *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 *hyperplane* yang optimal akan memperhatikan data-data yang tidak berada pada kelasnya (*misclassification error*) yang dilambangkan dengan ξ. Sehingga persamaan (2.7) menjadi sebagai berikut.

$$y_i(x_i + b) \geq 1 - \P_i \qquad (2.12)$$

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

$$L_d = \sum_{m=1}^{M} \sim_m - \frac{1}{2} \sum_{m_1=1}^{M} \sum_{m_2=1}^{M} \sim_{m_1} \sim_{m_2} y_{m_1} y_{m_2} K\left(X_{m_1} X_{m_2}\right) \qquad (2.13)$$

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 $x_k \rightarrow ¤(x_k)$. Sehingga nilai $w = \sum_{i=1}^{n} \sim_i y_i ¤(x_i)$ dan fungsi hasil *training* yang dihasilkan adalah:

$$f(x_k) = \sum_{i=1}^{n} \sim_i y_i K\left(x_{m_1}, x_{m_2}\right) + b \qquad (2.14)$$

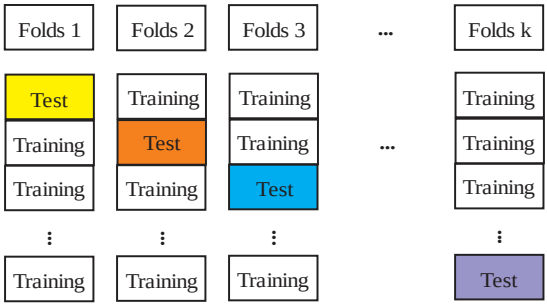
Pada praktiknya, *feature space* dapat memiliki dimensi yang tinggi dari vektor *input space*. (Gunn, 1998) Kernel RBF direkomendasikan untuk diuji pertama kali karena fungsi kernel RBF memiliki performansi yang sama dengan kernel linier pada parameter tertentu seperti parameter *C*. *C* adalah parameter 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.

Tabel 2.4 Fungsi Kernel pada SVM

Fungsi Kernel	**Rumus K (x_{m1}, x_{m2})**	**Parameter**
Linier	$x_{m1}^{T} x_{m2}$	C
RBF	$\exp\left(-\frac{\left(x_{m_1}-x_{m_2}\right)^{T}\left(x_{m_1}-x_{m_2}\right)}{2¥^{2}}\right)$	¥ dan C

2.9 *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-ulang membagi data menjadi data *training* dan data *testing*, dimana setiap data mendapat kesempatan menjadi data *testing* (Gokgoz & Subasi, 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

2.10 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 yang terdiri dari *TP* (*True Positive*) yaitu

jumlah *tweet* bersentimen positif yang tepat terprediksi dalam kelas positif, *TN* (*True Negative*) yaitu *tweet* bersentimen negatif yang tepat terprediksi dalam kelas negatif, *FP* (*False Positive*) yaitu *tweet* bersentimen negatif yang terprediksi dalam kelas positif, dan *FN* (*False Negative*) yaitu *tweet* bersentimen positif yang terprediksi dalam kelas negatif. Berikut merupakan *confusion matrix* yang memuat keempat nilai tersebut.

Tabel 2.5 *Confusion Matrix*

Kelas Aktual	Kelas Prediksi	
	Positif	Negatif
Positif	*TP*	*FN*
Negatif	*FP*	*TN*

Pengukuran yang sering digunakan untuk menghitung ketepatan klasifikasi adalah akurasi, *specificity*, dan *sensitivity* (Hotho, Nurnberger, & Paass, 2005). Akurasi merupakan persentase dokumen 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. Berikut merupakan rumus dalam meng-hitung akurasi, *specivicity* dan *sensitivity*.

$$Akurasi = \frac{TN + TP}{TN + TP + FN + FP} \tag{2.15}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.16}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.17}$$

Sedangkan untuk data *imbalanced*, pengukuran ketepatan klasifikasi yang digunakan adalah *G-Mean*. *G-Mean* atau *geometric mean* merupakan rata-rata geometrik nilai *recall* dari data yang memiliki dua kategori (Sun, Kamel, & Wang, 2006). Dalam mengukur nilai performansi klasifikasi, *G-Mean* memiliki kelebihan yaitu nilai klasifikasi yang dihasilkan *robust*. Berikut merupakan rumus untuk mendapatkan nilai *G-Mean*. Selain *G-*

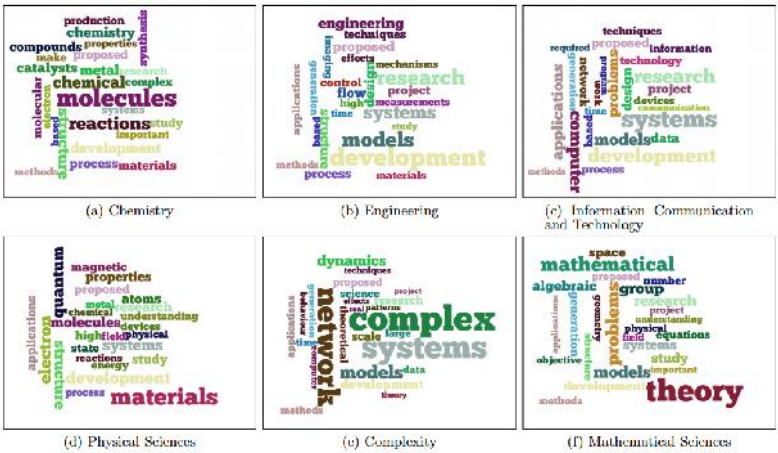
Mean juga digunakan nilai Area Under Curve (AUC). AUC merupakan indikator performansi kurva ROC (*Receiver Operating Characteristic*) yang dapat meringkas kinerja sebuah *classifier* menjadi satu nilai (Bekkar, Djemaa, & Alitouch, 2013).

$$G - mean = \sqrt{Sensitivity \times Specificity} \qquad (2.18)$$

$$AUC = \frac{1}{2}(Sensitivity + Specivicity) \qquad (2.19)$$

2.11 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang sering digunakan. *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. Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut muncul dalam dokumen. Berikut merupakan contoh dari visualisasi dokumen teks dengan *word cloud* (Castella & Sutton, 2014).



Gambar 2.4. Contoh Visualisasi Data dengan *Word Cloud*

2.12 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). Mikroblog adalah salah satu jenis alat komunikasi online dimana pengguna dapat memperbarui status tentang mereka yang sedang memikirkan dan melakukan sesuatu, apa pendapat mereka tentang suatu objek atau fenomena tertentu. *Tweet* adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna. *Tweet* bisa dilihat secara publik, namun pengirim dapat membatasi pengiriman pesan ke daftar teman-teman mereka saja. Pengguna dapat melihat *tweet* pengguna lain yang dikenal dengan sebutan pengikut (*follower)*.

Trending topic pada twitter merupakan isu yang sedang banyak diulas oleh pengguna Twitter yang dihitung dengan penanda *hashtag* (#). *Hashtag* digunakan untuk menandai suatu topik tertentu agar dapat seragam dengan pembahasan pengguna lain atau agar dapat dicari pengguna lain yang tertarik dengan topik yang sama.

2.13 Media *Mainstream*

Terdapat dua jenis media massa yaitu media *mainstream* dan media alternatif. Media *mainstream* adalah istilah yang digunakan untuk merujuk secara kolektif berbagai media massa besar yang mempengaruhi sebagian besar masyarakat dan membentuk suatu arus pemikiran masyarakat (Chomsky, 2014). Media *mainstream* dapat berupa media cetak, media televisi, maupun media online.

Beberapa media *mainstream* yang beroperasi diranah media televisi dan fokus pada program berita adalah TV One, Metro TV, dan Kompas TV. TV One merupakan stasiun televisi yang diresmikan pada 14 Februari 2008. TV One mempunyai fokusan untuk menginspirasi masyarakat Indonesia usia 15 tahun keatas untuk berpikiran maju dan melakukan perbaikan bagi diri sendiri serta masyarakat sekitar melalui berbagai program berita dan olahraga (TV One, 2017). Metro TV adalah stasiun televisi swasta berita yang didirikan oleh PT Media Televisi Indonesia. Metro TV resmi mengudara sejak 25 November 2000. Sebagai stasiun televisi berita, Metro TV setidaknya menyediakan 17 program berita yang

mendominasi program acara di stasiun TV ini (Metro TV, 2017). Kompas TV merupakan stasiun televisi berita yang didirikan Kompas Gramedia. Kompas TV resmi tayang pada September 2011 dan saat ini telah menjangkau lebih dari 100 kota di Indonesia. Kompas TV mempunyai 20 program berita unggulan yang membuat program berita tersebut mendominasi dari keseluruhan program acara Kompas TV.

(Halaman sengaja dikosongkan)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah kumpulan *tweet* mengenai media *mainstream* TV One, Metro TV, dan Kompas TV pada tanggal 12 Februari 2017 hingga 2 April 2017. Data didapat dari Twitter API (Application Programming Interface) sebanyak 3000 *tweet*.

3.2 Struktur Data

Data yang berjumlah 1000 pada setiap media dibagi menjadi data *training* dan data *testing* dengan perbandingan 90%:10% menggunakan 10-*fold cross validation*. Struktur data yang digunakan dalam penelitian ini setelah dilakukan praproses pada data teks *tweet* terdiri dari variabel prediktor yaitu kata dasar setiap *tweet* dan variabel respon yaitu klasifikasi sentimen *tweet* (positif dan negatif). Berikut merupakan contoh struktur data penelitian sebelum praproses.

Tabel 3.1 Contoh Struktur Data Penelitian

No	Tweet	Sentimen
1	saya lihat tv one beritanya bagus wartawan membaur dengan peserta	Positif
2	tayangan bagus metro tv pagi ini di awal tahun	Positif
3	acara kompas tv mencerdaskan	Positif
4	kecewa ilc batal tayang	Negatif
5	jitu metro tv telah sebarkan berita hoax	Negatif
6	berita kompas tv diulang ulang membosankan	Negatif
…	…	…

3.3 Langkah Analisis Data

Tahapan analisis data yang dilakukan dalam penelitian ini sebagai berikut.

1. Mengambil data *tweet* dengan Twitter API.

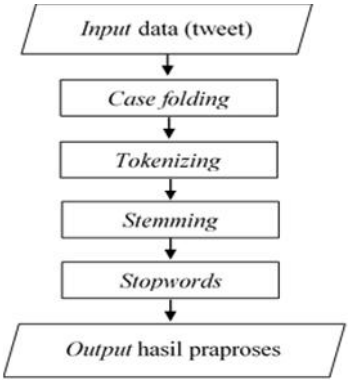
a) Memasukkan *keyword* yang berhubungan dengan stasiun televisi TV One, Metro TV, dan Kompas TV.
b) Menyimpan hasil *searching* ke database.

2. Menyiapkan data *tweet*, daftar *stopwords*, dan kata dasar.
   a) Data *tweet* dibagi menjadi data *training* dan data *testing* menggunakan *10-fold cross validation* dengan perbandingan *training-testing* sebesar 90%:10%.
   b) Daftar *stopwords*, didapatkan dari tesis F. Z. Tala yang berjudul "*A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*" (Tala, 2003).
   c) Kata dasar diambil dari Kamus Besar Bahasa Indonesia.
3. Praproses Teks
   a) Menghapus *tweet* yang mengandung dua sentimen, positif dan negatif.
   b) Menghapus simbol *retweet* (*response tweet*) "RT".
   c) Menghilangkan *link* URL.
   d) Melakukan *case folding*, yaitu mengubah semua teks dengan huruf kecil (non kapital) serta menghilangkan tanda baca.
   e) Menghapus kata pada *tweet* yang terdapat dalam daftar *stopwords*.
   f) Melakukan *tokenizing* untuk memecah *tweet* menjadi kata per kata.
   g) Melakukan *steeming* untuk menghilangkan kata imbuhan dan mendapatkan kata dasar.
   h) Mengubah data *tweet* kedalam bentuk frekuensi kemunculan kata seperti pada contoh struktur data pada Tabel 2.1
4. Klasifikasi data menggunakan *Naïve Bayes Classifier* untuk masing-masing media.
   a) Menghitung probabilitas dari V_j pada data *training* dengan persamaan (2.3), dimana V_j merupakan kategori sentimen, yaitu V_1 = negatif, dan V_2 = positif.
   b) Menghitung probabilitas kata a_i pada kategori V_j dengan persamaan (2.4).
   c) Model probabilitas NBC disimpan dan digunakan untuk tahap data *testing*.

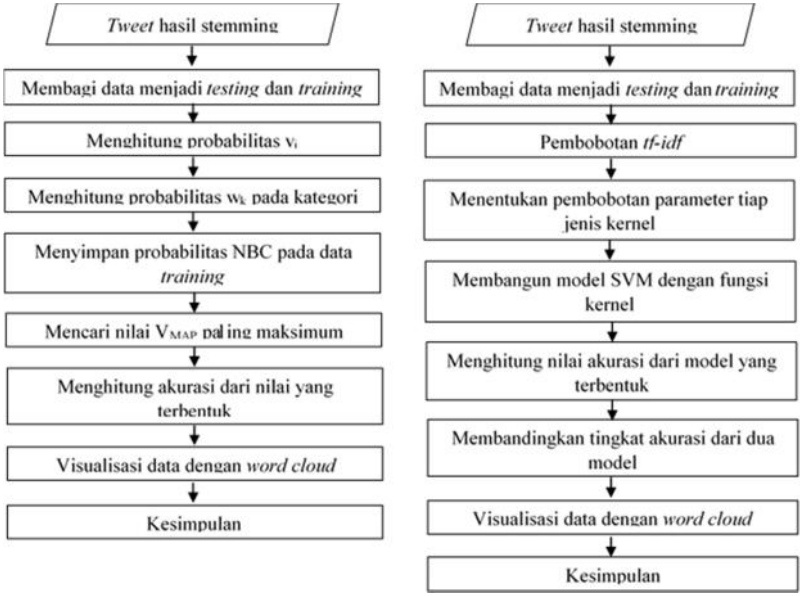
d) Menghitung probabilitas tertinggi dari kategori sentimen yang diujikan (V_{MAP}) dengan persamaan (2.1).
e) Mencari nilai V_{MAP} paling maksimum dan memasukkan tweet tersebut pada kategori dengan V_{MAP} maksimum.

5. Klasifikasi data menggunakan *Support Vector Machine* untuk masing-masing media.
   a) Merubah teks menjadi vektor dan pembobotan kata dengan *tf-idf* menggunakan persamaan (2.5).
   b) Menentukan pembobot parameter pada SVM tiap jenis kernel.
   c) Membangun model SVM menggunakan fungsi *Radial Basis Function* dan linier.
6. Evaluasi hasil klasifikasi
   Menghitung ketepatan klasifikasi dan membandingkan performansi metode NBC dan SVM berdasarkan tingkat akurasi, *spesitivity*, dan *sensitivity* dengan persamaan (2.15), (2.16), dan (2.17) jika data *balance* atau berdasarkan nilai *G-mean* dan AUC dengan persamaan (2.18) dan (2.19) jika data *unbalance*.
7. Melakukan visualisasi *tweet* dengan *word cloud*
8. Interpretasi dan menarik kesimpulan

Diagram alir penelitian disajikan pada Gambar 3.1 dan Gambar 3.2 sebagai berikut.



Gambar 3.1 Diagram Alir Praproses Teks



(a) (b)

Gambar 3.2 Diagram Alir Klasifikasi NBC (a) dan SVM (b)

BAB IV
ANALISIS DAN PEMBAHASAN

4.1 Praproses dan Karakteristik Data

Data *tweet* mengenai ketiga media, TV One, Metro TV, dan Kompas TV yang telah terkumpul dilakukan praproses teks meliputi *case folding*, *stopword*, *stemming*, dan *tokenizing*. Praproses teks dilakukan dengan langkah-langkah seperti pada Gambar 3.1. Berikut merupakan struktur data *tweet* mengenai salah satu media *mainstream* (Kompas TV) sebelum dilakukan praproses data.

Tabel 4.1 Struktur Data Kompas TV Sebelum Praproses

Class	Tweet
Negt	RT @DhikkiE: Dari sekian banyak jurnalis kenapa cuma metro tv & kompas tv yg diusir massa? Itu artinya media tsb bermasalah, karena selalu…
Post	RT @PieterSeno: Mobil Kompas TV diusir? Ini pelecehan KEBEBASAN PERS. Di tmpt ibadah kok ada kekerasan n pengusiran? Sangat disayangkan...m…
Negt	RT @DhikkiE: Dari sekian banyak jurnalis kenapa cuma metro tv & kompas tv yg diusir massa? Itu artinya media tsb bermasalah, karena selalu…
Negt	RT @akhdanarif: metro tv tipu tipu kompas kafir tvone gajelas lu baca bobo aja sono tuh bona rongrong lagi kampanye
Negt	RT @progres_98: Sebaiknya Pemred Kompas & Metro TV minta maaf pd umat Islam. Stop NIPU! Pers pembela rakyat dihormati, klu kerjanya suka fi…
.	.
.	.
.	.
Negt	@spardaxyz @EDDYSANTRI @MariaAnnie5 Apalagi CNN, media asing pro liberal.. CNN tdk sama seperti Metro TV & Kompas yg suka mlintir" kenyataan

Data *tweet* yang belum dilakukan praproses masih tersusun dalam satu kolom seperti pada Tabel 4.1. Data tersebut masih memuat *username*, *link URL*, kata-kata yang dianggap bukan merupakan kata penting dalam *tweet* (*stopwords*), dan simbol-simbol lainnya yang tidak menggambarkan isi *tweet* seperti simbol *retweet* (RT) dan tanda baca, sehingga perlu dilakukan praproses guna mendapatkan data *tweet* yang tidak memuat hal-hal tersebut. Selain itu, praproses data juga bertujuan untuk meningkatkan ketepatan klasifikasi dan mengurangi kesalahan klasifikasi data. Berikut merupakan struktur data Kompas TV yang telah dilakukan praproses teks.

Tabel 4.2 Struktur Data Kompas TV Setelah Praproses

tweet	**absah**	**…**	**acara**	**…**	**aja**	**…**	**aksi**	**…**	**zara**
sekian jurnalis…	0	…	0	…	0	…	0	…	0
mobil usir…	0	…	0	…	0	…	0	…	0
sekian jurnalis…	0	…	0	…	0	…	0	…	0
...bobo aja…	0	…	0	…	1	…	0	…	0
pemred metro…	0	…	0	…	0	…	0	…	0
sekian jurnalis…	0	…	0	…	0	…	0	…	0
...liput acara…	0	…	1	…	0	…	0	…	0
malam mobil…	0	…	0	…	0	…	0	…	0
senang quote…	0	…	0	…	0	…	0	…	0
wartawan usir…	0	…	0	…	0	…	0	…	0
...massa aksi…	0	…	0	…	0	…	1	…	0
sampe gue…	0	…	0	…	0	…	0	…	0
wartawan usir…	0	…	0	…	0	…	0	…	0
...usir aksi…	0	…	0	…	0	…	1	…	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
cnn media…	0	…	0	…	0	…	0	…	0

Data yang telah berbentuk *document term matrix* seperti pada Tabel 4.2 tersebut, dapat dilakukan perhitungan jumlah kata yang selanjutnya akan menjadi jumlah variabel dari setiap media. Data *tweet* media TV One memiliki jumlah kata sebanyak 1324 kata. Data *tweet* media Metro TV memiliki jumlah kata sebanyak 1303 kata. Sedangkan data *tweet* media Kompas TV mempunyai jumlah kata sebanyak 976 kata. Setelah terbentuk struktur data yang diinginkan, dilakukan perhitungan frekuensi kemunculan kata pada setiap media. Berikut merupakan frekuensi kemunculan kata tertinggi dari setiap media.

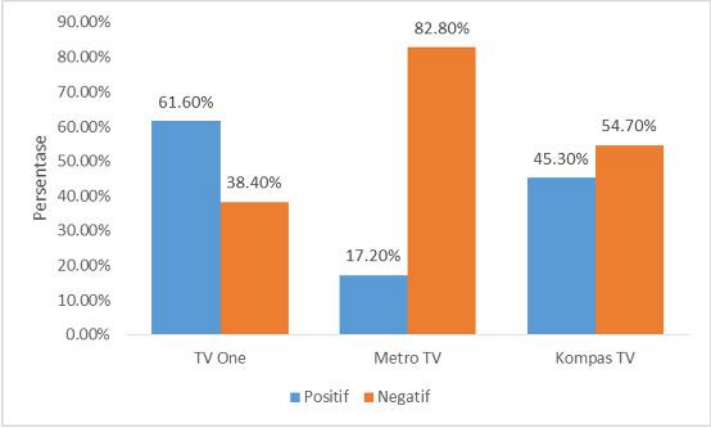
Tabel 4.3 Frekuensi Kemunculan Kata Tertinggi Setiap Media

TV One		**Metro TV**		**Kompas TV**	
Kata	**Jumlah**	**Kata**	**Jumlah**	**Kata**	**Jumlah**
selamat	172	ahok	376	metro	474
aksi	156	mendagri	252	media	341
aniaya	153	narasumber	251	kasih	170
wartawan	152	mahfud	250	terima	170
berita	126	jabat	250	net	164
beda	118	potong	245	selesai	163
indonesia	100	kata	240	marcuskevin	162
moga	78	media	199	safari	162
siar	72	berita	111	mengund…	161
ulang	71	banjir	89	konsisten	113
informasi	70	laku	70	nayangin	108
masyarakat	65	seru	68	bulutangkis	107
usia	64	tunggu	67	respek	104
tapak	60	licik	66	banget	97
bebas	58	htt	66	ahok	96

Daftar kata dengan frekuensi kemunculan tertinggi pada setiap data media *mainstream* pada Tabel 4.3 menunjukkan bahwa

kata-kata tersebut merupakan kata-kata yang mempunyai pengaruh signifikan dalam pembangunan model klasifikasi.

Data kategori sentimen sebagai variabel respon pada penelitian ini akan disajikan dalam *bar chart* guna mengetahui perbandingan frekuensi antar kategori.



Gambar 4.1 *Bar Chart* Kategori Data Setiap Media

Frekuensi kategori data *training* pada Gambar 4.1 menunjukkan bahwa kategori pada media TV One dan Kompas TV cenderung *balance* atau seimbang antara kategori sentimen positif dan negatif, sedangkan frekuensi antara kategori sentimen positif dan negatif pada media Metro TV cenderung *imbalance*. Keadaan *imbalance* pada kategori data tersebut akan berpengaruh pada perhitungan ketepatan klasifikasi. Ukuran ketepatan klasifikasi akurasi tidak sesuai untuk data *imbalance* sehingga data media Metro TV akan dilakukan pengukuran ketepatan klasifikasi dengan *G-mean* dan AUC, sedangkan data media TV One dan Kompas TV tetap menggunakan ukuran akurasi. Dari Gambar 4.1 tersebut juga dapat dilihat media TV One merupakan media yang paling banyak mendapatkan sentimen positif dari publik pengguna twitter yaitu sebesar 61,6%.

4.2 Klasifikasi Menggunakan *Naïve Bayes Classifier*

Langkah pertama dalam mengklasifikasikan data *tweet* adalah melatih model menggunakan data *training*. Data *training* dari setiap media yang telah dilakukan praproses digunakan untuk melatih model menggunakan *software* Python 2.7 dan Jupyter Notebook. Model yang telah dilatih dengan data *training* kemudian digunakan untuk mengklasifikasikan data *testing* ke dalam dua kelas sentimen, positif dan negatif. Pembagian data *training* dan *testing* berdasarkan metode 10-*fold cross validation*. Klasifikasi dengan metode *Naïve Bayes Classifier* menghasilkan probabilitas yang digunakan untuk menentukan apakah *tweet* masuk ke dalam kategori sentimen positif atau negatif. Probabilitas tersebut diperoleh dengan persamaan (2.3) dan persamaan (2.4). Perhitungan probabilitas setiap kategori *tweet* seperti yang telah dijelaskan ilustrasi pada subbab 2.6. Berikut beberapa nilai probabilitas yang dihasilkan dari model pada salah satu data media yaitu media Kompas TV.

Tabel 4.4 Probabilitas Klasifikasi NBC Media Kompas TV

Probabilitas Negatif	Probabilitas Positif	Keputusan
0.9999191	8.09E-05	Negatif
0.0013797	0.9986203	Positif
0.9999191	8.09E-05	Negatif
1	3.21E-09	Negatif
1	3.73E-10	Negatif
0.9999191	8.09E-05	Negatif
0.9992907	0.0007093	Negatif
0.9997174	0.0002826	Negatif
0.0810289	0.9189711	Positif
.	.	.
.	.	.
.	.	.
0.9905825	0.0094175	Negatif

Nilai probabilitas tweet pada Tabel 4.4 tersebut menunjukkan bahwa *tweet* mempunyai peluang untuk masuk ke dalam kategori sentimen sebesar nilai yang ada pada kedua kolom sentimen. Suatu *tweet* akan masuk ke dalam salah satu kategori sentimen yang nilainya paling besar. Sehingga jika probabilitas *tweet* masuk ke dalam kategori sentimen positif lebih besar dari probabilitas *tweet* masuk ke dalam kategori sentimen negatif maka *tweet* tersebut masuk ke dalam kategori sentimen positif dan sebaliknya. Dari probabilitas tersebut didapat kategori prediksi dari setiap *tweet*. Langkah selanjutnya adalah mengukur ketepatan klasifikasi pada data *training* dan data *testing* dari setiap media. Pengukuran ketepatan klasifikasi dilakukan dengan membentuk *confusion matrix* berdasarkan hasil prediksi. Berikut merupakan contoh *confusion matrix* kategori aktual dan prediksi pada data *training* Kompas TV.

Tabel 4.5 *Confusion Matrix* Data *Training* Kompas TV

Kelas Aktual	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	437	16
Negatif	6	541

Setelah terbentuk *confusion matrix* seperti pada Tabel 4.5, langkah selanjutnya adalah melakukan perhitungan ketepatan klasifikasi menggunakan persamaan (2.15) hingga persamaan (2.19).

$$Akurasi = \frac{437+541}{437+541+16+6} = \frac{978}{1000} = 0,978$$

$$Sensitivity = \frac{437}{437+16} = \frac{437}{453} = 0,9647$$

$$Specificity = \frac{541}{6+541} = \frac{541}{547} = 0,989$$

$$G-mean = \sqrt{0,9647 \times 0,989} = 0,9768$$

$$AUC = \frac{1}{2}\left(0,9647 + 0,989\right) = 0,9769$$

Hasil perhitungan ketepatan klasifikasi setiap media selanjutnya dirangkum menjadi satu tabel berdasarkan data *training* dan data *testing.* Berikut merupakan hasil pengukuran ketepatan klasifikasi dari setiap media menggunakan algoritma Naïve Bayes Classifier.

Tabel 4.6 Ketepatan Klasifikasi NBC

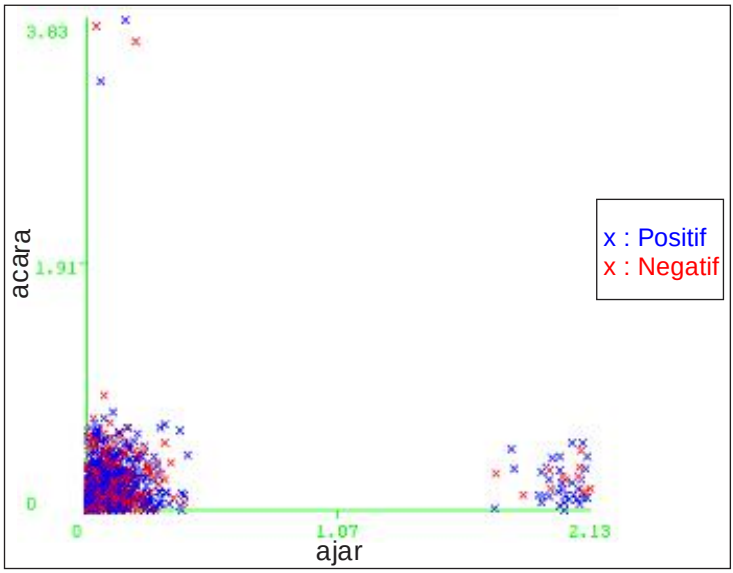
Data	Media	Akurasi	G-Mean	AUC
Training	TV One	0.9580	0.9518	0.9522
	Metro TV	0.9100	0.8130	0.8236
	Kompas TV	0.9780	0.9768	0.9769
Testing	TV One	0.7900	0.6984	0.7439
	Metro TV	0.9700	0.7559	0.7857
	Kompas TV	0.8700	0.8394	0.8523

Seperti pada pembahasan subbab 4.1 bahwa ukuran ketepatan klasifikasi yang akan digunakan untuk media TV One dan Kompas TV adalah akurasi karena data TV One dan Kompas TV cenderung seimbang (*balance*), sedangkan untuk media Metro TV menggunakan ukuran ketepatan klasifikasi G-Mean dan AUC. Klasifikasi data *training* TV One memiliki akurasi sebesar 95,8% yang berarti 95,8% dari data *training* media TV One tepat terklasifikasi pada kelas aktual, sedangkan pada data *testing* mendapat akurasi sebesar 79%. Klasifikasi data *training* data Metro TV menghasilkan ketepatan klasifikasi *G-mean* dan AUC sebesar 81,3% dan 82,36%, sedangkan pada data *testing* mendapat nilai *G-mean* dan AUC sebesar 75,59% dan 78,57%. Klasifikasi data Kompas TV menghasilkan ketepatan klasifikasi yang tinggi baik pada data *training* maupun data *testing* yaitu masing-masing mendapatkan akurasi sebesar 97,8% dan 87%.

4.3 Klasifikasi Menggunakan *Support Vector Machine*

Algoritma Support Vector Machine pada penelitian ini menggunakan dua macam kernel yaitu kernel *linear* dan kernel

radial basis function. Data yang akan dianalisis menggunakan *Support Vector Machine* dilakukan pembobotan *term frequency-inverse document frequency* (tf-idf) dulu pada setiap kata. Pembobotan pada data menyebabkan adanya sebaran data. Berikut sebaran data variabel ajar vs acara pada sentimen positif dan negatif.



Gambar 4.3 *Scatter Plot* Variabel Ajar dan Variabel Acara

Scatter plot antara variabel ajar dan variabel acara pada Gambar 4.3 menunjukkan bahwa terdapat kemungkinan data tidak terpisah secara linier, sehingga pada pembahasan klasifikasi menggunakan SVM akan dilakukan dengan SVM kernel *linear* dan SVM kernel *radial basis function* (RBF).

4.3.1 Klasifikasi Menggunakan SVM Kernel *Linear*

Pelatihan model menggunakan data *training* dengan SVM *linear* mempertimbangkan parameter C. Parameter C akan dicoba dari 10^{-3} hingga 10^{3}. Model yang telah dilatih dengan data *training* selanjutnya akan digunakan untuk mengklasifikasikan data

training dan *testing* pada masing-masing media. Hasil perhitungan ketepatan klasifikasi pada data *training* dan *testing* setiap media dapat dilihat pada Lampiran 1 hingga Lampiran 4. Berikut merupakan hasil ketepatan klasifikasi terbaik pada data ketiga media menggunakan SVM kernel *linear*.

Tabel 4.7 Ketepatan Klasifikasi Terbaik dengan SVM *Linear*

Data	Media	C	Akurasi	G-Mean	AUC
Training	TV One	10	0.9790	0.9750	0.9751
	Metro TV	10	0.9880	0.9717	0.9720
	Kompas TV	1000	0.9860	0.9870	0.9870
Testing	TV One	10	0.8400	0.8051	0.8160
	Metro TV	100	0.9600	0.7519	0.7803
	Kompas TV	1	0.8500	0.8547	0.8563

Klasifikasi terbaik yang didapat menggunakan SVM kernel *linear* pada masing-masing data *training* setiap media mendapatkan ketepatan klasifikasi yang sangat baik dengan nilai akurasi, *G-mean*, dan AUC yang hampir sama pada setiap media. Pada data *training* TV One dan Metro TV, ketepatan klasifikasi terbaik didapat menggunakan parameter C sebesar 10, sedangkan pada data *training* Kompas TV, ketepatan klasifikasi terbaik menggunakan parameter C sebesar 1000. Pada data *testing* TV One didapat akurasi terbaik sebesar 84% menggunakan parameter C sebesar 10. Pada data *testing* Metro TV didapat nilai *G-mean* dan AUC terbaik sebesar 75,19% dan 78,03% menggunakan parameter 100. Sedangkan pada data Kompas TV didapat akurasi terbaik sebesar 85% menggunakan parameter C sebesar 1.

4.3.2 Klasifikasi Menggunakan SVM Kernel RBF

Pembahasan klasifikasi data menggunakan SVM kernel *radial basis fuction* akan sama dengan pembahasan pada klasifikasi data menggunakan SVM kernel *linear*, perbedaannya adalah pada klasifikasi SVM kernel RBF menggunakan 2

parameter yaitu parameter C dan parameter gamma. Nilai parameter C dan parameter gamma menggunakan rentang nilai yang sama yaitu 10^{-2} hingga 10^{2}. Hasil perhitungan ketepatan klasifikasi menggunakan SVM kernel RBF pada setiap media dapat dilihat pada Lampiran 5 hingga Lampiran 8. Berikut merupakan hasil ketepatan klasifikasi terbaik dari setiap media dengan kriteria terbaik dari media TV One, Kompas TV, dan Metro TV.

Tabel 4.8 Ketepatan Klasifikasi SVM Kernel RBF

Data	Media	C	Gamma	Akurasi	G-Mean	AUC
Training	TV One	10	1	0.9790	0.9734	0.9736
	Metro TV	10	1	0.9910	0.9735	0.9738
	Kompas TV	10	1	0.9930	0.9922	0.9923
Testing	TV One	10	0.1	0.8500	0.8123	0.8245
	Metro TV	10	0.1	0.9700	0.7559	0.7857
	Kompas TV	1	1	0.8700	0.8763	0.8839

Tabel 4.8 menunjukkan ketepatan klasifikasi terbaik yang didapat dari klasifikasi data *training* dan *testing* menggunakan SVM kernel *radial basis function*. Ketepatan klasifikasi terbaik data *training* pada ketiga media menggunakan kombinasi parameter yang sama yaitu dengan parameter C sebesar 10 dan gamma sebesar 1. Nilai akurasi, *G-mean*, dan AUC yang didapat sangat tinggi yaitu masing-masing lebih besar dari 97%. Ketepatan klasifikasi data *testing* TV One menggunakan parameter C sebesar 10 dan gamma sebesar 0,1 mendapatkan akurasi 85%. Ketepatan klasifikasi terbaik pada data *testing* Metro TV menghasilkan nilai *G-mean* dan AUC sebesar 75,59% dan 78,57% menggunakan parameter C dan gamma masing-masing sebesar 10 dan 0,1. Ketepatan klasifikasi terbaik pada data *testing* Kompas TV mendapatkan akurasi 87% menggunakan nilai parameter C dan gamma yang sama yaitu sebesar 1.

Hasil ketepatan klasifikasi menggunakan SVM kernel *linear* dan SVM kernel RBF hampir sama. Ketepatan klasifikasi menggunakan SVM kernel RBF sedikit lebih tinggi dibandingkan menggunakan SVM kernel *linear* baik pada data *training* maupun data *testing*.

4.3.3 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. Hasil ketepatan klasifikasi terbaik ketiga data media pada metode SVM kernel RBF menggunakan nilai parameter yang sama yaitu menggunakan nilai gamma sebesar 1, kemudian nilai gamma tersebut disubtitusikan pada persamaan kernel RBF yang terdapat pada Tabel 2.4, sehingga fungsi kernel RBF pada ketiga data media yang terbentuk sama yaitu sebagai berikut.

$$K(x_{m_1}, x_{m_2}) = \exp\left(-\left(0,5 \times \left(x_{m_1} - x_{m_2}\right)^T \left(x_{m_1} - x_{m_2}\right)\right)\right)$$

Fungsi kernel tersebut kemudian digunakan untuk membentuk fungsi *hyperplane* dengan cara mensubtitusikan nilai *support vector* kategori positif pada x_{m1} dan nilai *support vector* kategori negatif pada $x_{m2.}$ Fungsi *hyperplane* dihitung dengan mensubtitusikan fungsi kernel pada persamaan (2.14). Sehingga didapat fungsi *hyperplane* pada setiap data media sebagai berikut.

Tabel 4.9 Persamaan *Hyperplane* pada Setiap Media

Media	**Persamaan *Hyperplane***
TV One	$f(x) = \sum_{m=1}^{1324}\left(-0,1617\tilde{}_m x + ... + 0,8564\tilde{}_m x\right) - 0,4031$
Metro TV	$f(x) = \sum_{m=1}^{1303}\left(0,3962\tilde{}_m x + ... + 0,0367\tilde{}_m x\right) + 0.6036$
Kompas TV	$f(x) = \sum_{m=1}^{976}\left(0,8376\tilde{}_m x + ... + 0,1590\tilde{}_m x\right) + 0,1622$

Persamaan *hyperplane* yang telah didapat pada setiap media digunakan untuk mengklasifikasi data. Pada persamaan *hyperplane* tersebut, α_m merupakan nilai koefisien dari *support vector* dan *x* adalah nilai input yang akan diklasifikasi. Jika didapat nilai $f(x) \leq 0,5$ maka data tersebut akan dikategorikan dalam *tweet* dengan sentimen negatif, namun jika nilai $f(x) \geq 0,5$ maka data tersebut akan dikategorikan dalam *tweet* dengan sentimen positif.

4.4 Perbandingan Antara NBC dan SVM

Langkah yang akan dilakukan setelah mendapatkan ketepatan klasifikasi dari masing-masing metode dan di setiap media adalah membandingkan hasil dari kedua metode tersebut. Perbandingan metode *Naïve Bayes Classifier* dan *Support Vector Machine* pada penelitian ini mempertimbangkan hasil terbaik dari ketepatan klasifikasi terbaik dari setiap media dan metode.

Tabel 4.10 Perbandingan Ketepatan Klasifikasi Data *Traininig*

Media	Metode	Akurasi	G-Mean	AUC
TV One	NBC	0.9580	0.9518	0.9522
	SVM	**0.9790**	**0.9734**	**0.9736**
Metro TV	NBC	0.9100	0.8130	0.8236
	SVM	**0.9910**	**0.9735**	**0.9738**
Kompas TV	NBC	0.9780	0.9768	0.9769
	SVM	**0.9930**	**0.9922**	**0.9923**

Perbandingan ketepatan klasifikasi pada data *training* yang terdapat pada Tabel 4.10 menunjukkan bahwa secara keseluruhan performa metode SVM lebih baik baik dibandingkan metode NBC. Perbandingan tersebut dilihat dari nilai akurasi, *G-mean* dan AUC dari ketiga media menunjukkan metode SVM menghasilkan ketepatan klasifikasi yang lebih besar daripada NBC. Namun kedua metode sama-sama menghasilkan tingkat ketepatan klasifikasi yang tinggi pada data *training*.

Tabel 4.11 Perbandingan Ketepatan Klasifikasi Data *Testing*

Media	Metode	Akurasi	G-Mean	AUC
TV One	NBC	0.7900	0.6984	0.7439
	SVM	**0.8500**	**0.8123**	**0.8245**
Metro TV	NBC	0.9700	0.7559	0.7857
	SVM	0.9700	0.7559	0.7857
Kompas TV	NBC	0.8700	0.8394	0.8523
	SVM	**0.8700**	**0.8763**	**0.8839**

Ketepatan klasifikasi data *testing* terbaik pada Tabel 4.11 menunjukkan perbandingan yang hampir sama dengan Tabel 4.10. Ketepatan klasifikasi data *testing* menggunakan SVM pada media TV One dan Kompas TV lebih besar daripada ketepatan klasifikasi data *testing* menggunakan NBC. Pada data Metro TV mendapatkan ketepatan klasifikasi yang sama antara kedua metode.

Dari segi waktu pelatihan model dan klasifikasi menggunakan *software* Python 2.7 dan Jupyter Notebook dari kedua metode tidak menunjukkan perbedaan waktu yang signifikan. Kedua metode melatih dan melakukan prediksi kategori data dengan cepat.

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 sentimennya sehingga dapat diketahui kata-kata yang sering muncul pada setiap sentimen. Visualisasi *word cloud* dilakukan menggunakan *software* RStudio. Ukuran *font* pada *word cloud* menunjukkan frekuensi kemunculan kata. Semakin besar ukuran *font* berarti semakin besar frekuensi kemunculan kata tersebut.

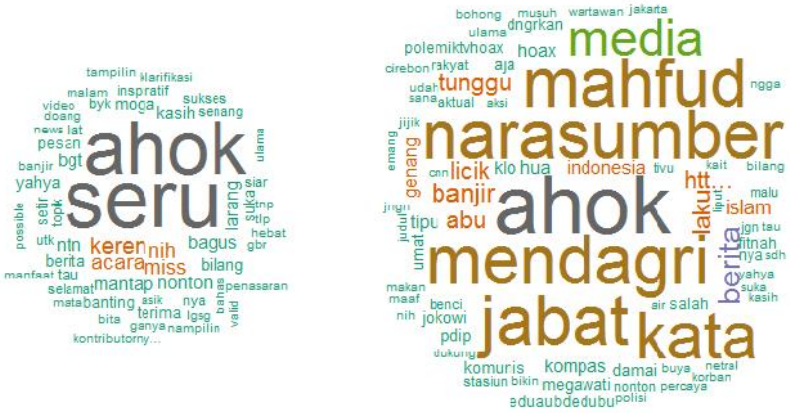
Visualisasi *word cloud* akan dilakukan dengan membandingkan antara data *tweet* yang memiliki sentimen positif dan data *tweet* yang memiliki sentimen negatif. Perbandingan tersebut dilakukan dengan tujuan mengetahui penyebab mayoritas publik

menilai masing-masing media berdasarkan kategori sentimen. Berikut merupakan *word cloud* data pada setiap media.

Gambar 4.4 *Word Cloud* Media TV One Sentimen Positif (kiri) dan Negatif (kanan)

Word cloud data media TV One pada Gambar 4.4 menunjukkan kata-kata yang sering muncul pada setiap kategori sentimen. Kata yang paling sering muncul pada sentimen positif media TV One yaitu kata “selamat”, “indonesia”, “beda”, “berita” “ulang”, “moga” dan lain-lain yang menunjukkan bahwa publik pengguna Twitter banyak yang sedang membahas hari ulang tahun dan turut memberikan selamat, doa, dan dorongan yang dapat dilihat dari kata-kata yang berukuran lebih kecil seperti kata “moga”, “semangat”, “mantap”, dan lainnya. Selain itu publik pengguna Twitter menilai positif media TV One dilihat dari kata “imbang”, “kritis”, “sambung”, “inspirasi”, dan “obyektif” sehingga kata-kata tersebut menjadi saran dan prioritas pihak penyedia media TV One untuk meningkatkan kualitas dari penyiaran berita yang berimbang, kritis, obyektif, menginspirasi, dan penyambung suara rakyat. Pada sentimen negatif, *word cloud* didominasi kata “aksi”, “aniaya”, dan “wartawan” yang mengindikasikan publik pengguna Twitter banyak yang sedang membahas pemberitaan

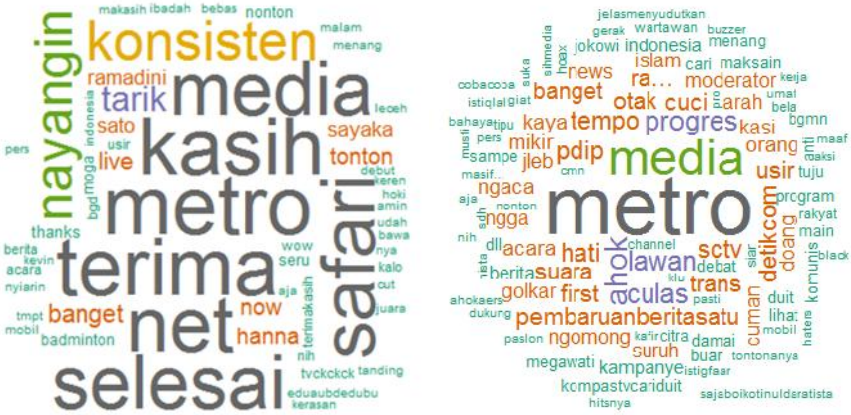
wartawan TV One yang dianiaya peserta aksi demo. Kata-kata lain pada sentimen negatif mempunyai frekuensi yang jauh lebih kecil yang ditandai dengan ukuran *font* yang berbeda jauh.



Gambar 4.5 *Word Cloud* Media Metro TV Sentimen Positif (kiri) dan Negatif (kanan)

Word cloud media Metro TV pada Gambar 4.5 terlihat berbeda ukuran antara kategori sentimen positif dan sentimen negatif. Hal ini disebabkan karena perbandingan publik pengguna Twitter yang menilai Metro TV positif dan negatif tidak seimbang. Hanya terdapat sedikit publik yang memiliki sentimen positif terhadap Metro TV seperti yang telah dijelaskan pada subbab 4.1. Kata yang sering muncul pada sentimen positif adalah kata “ahok” dan “seru” yang menunjukkan publik pengguna Twitter merespon positif adanya acara Mata Najwa yang mengundang pembicara Ahok atau Basuki Tjahya Purnama. Kata-kata lain pada kategori sentimen positif berukuran sangat kecil yang berarti kemunculan kata tersebut memiliki frekuensi yang rendah. Pada sentimen negatif, kata yang paling sering muncul adalah kata “ahok”. Hal ini menandakan mayoritas penilaian negatif publik pengguna Twitter terhadap Metro TV berkaitan dengan pembahasan tokoh Ahok. Kata-kata lain yang sering muncul adalah kata “mahfud”, “nara-sumber”, “mendagri”, “jabat”, dan “kata”. Hal ini berkaitan dengan

ketidakpuasan publik Twitter terhadap Metro TV yang selalu memotong perkataan Mahfud M.D. saat menjadi narasumber bersama Mendagri membahas tentang jabatan Ahok.



Gambar 4.6 *Word Cloud* Media Kompas TV Sentimen Positif (kiri) dan Negatif (kanan)

Kata-kata yang berukuran besar pada kategori positif Gambar 4.6 menunjukkan bahwa publik Twitter mengucapkan terimakasih kepada Kompas TV karena telah konsisten menayangkan pertandingan bulutangkis walaupun tim perwakilan Indonesia tidak lolos hinga final. Respon baik dari publik Twitter ini dapat dijadikan saran bagi penyedia media Kompas TV untuk terus menayangkan program olahraga terutama pertandingan bulutangkis. Selain itu terdapat kata "metro" dan "safari" pada kategori positif. Hal ini dikarenakan banyak yang mengucapkan terimakasih karena Kompas TV dan Metro TV telah menyiarkan safari media untuk atlit bulutangkis Indonesia usai pertandingan bulutangkis. Pada kategori sentimen negatif, terdapat kata yang paling sering muncul yaitu kata "metro" yang mengindikasikan bahwa publik Twitter menilai Kompas TV dan Metro TV memiliki karakter sama yang tidak disukai publik Twitter. Hal yang tidak disukai tersebut diantaranya tergambarkan pada kata-kata yang berukuran lebih kecil

seperti kata “ahok” yang mengindikasikan Kompas TV memihak Ahok, kata ”lawan” dan “culas” yang mewakili *tweet* lawan keculasan Kompas TV, dan lainnya. Melalui kata-kata yang sering muncul tersebut dapat dijadikan pertimbangan Kompas TV dalam perbaikan penyediaan media televisi.

(Halaman sengaja dikosongkan)

``

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan diperoleh kesimpulan sebagai berikut.

1. Hasil ketepatan klasifikasi menggunakan Naive Bayes Classifier pada media TV One dan Kompas TV diperoleh akurasi sebesar 95,8% dan 97,8%, sedangkan ketepatan klasifikasi media Metro TV diperoleh nilai G-*mean* dan AUC sebesar 81,3% dan 82,36%.
2. Hasil ketepatan klasifikasi menggunakan Support Vector Machine pada media TV One dan Kompas TV menghasilkan akurasi sebesar 97,9% dan 99,3%. Pada media Metro TV didapat nilai G-*mean* dan AUC sebesar 97,35% dan 97,38%. Performa SVM kernel RBF menghasilkan ketepatan klasifikasi yang yang lebih baik dibanding performa SVM kernel *linear*.
3. Secara keseluruhan perbandingan performa metode NBC dan SVM menunjukkan hasil bahwa performa SVM lebih baik dalam mengklasifikasi data.
4. Kata-kata yang sering muncul pada media TV One kategori sentimen positif adalah ucapan selamat atas ulang tahun TV One. Sedangkan pada kategori sentimen negatif didominasi kata ”aksi”, ”aniaya”, dan ”wartawan”. Pada media Metro TV kategori sentimen positif, kata yang sering muncul adalah ”ahok” dan ”seru” yang menunjukkan publik twitter mengapresiasi acara Metro TV yang mengundang Basuki Tjahya Purnama. Sedangkan pada kategori sentimen negatif didominasi kata yang menggambarkan pihak Metro TV sering memotong perkataan Mahfud M.D saat diundang pada acara Metro TV. Pada media Kompas TV kategori sentimen positif, kebanyakan kata yang sering muncul menggambarkan apresiasi Kompas TV telah menayangkan acara pertandingan bulutangkis. Sedangkan pada kategori sentimen negatif didominasi kata ”metro” yang menggambarkan publik twitter memberikan sentimen negatif ketika Kompas TV memiliki karakter yang berkaitan dengan Metro TV.

5.2 Saran

Saran yang dapat diberikan dari hasil penelitian ini adalah sebagai berikut.

1. Untuk penyedia media *mainstream* dapat melakukan analisis sentimen publik pengguna Twitter menggunakan metode NBC maupun SVM karena kedua metode menghasilkan ketepatan klasifikasi yang cukup baik. Selain itu penyedia media *mainstream* dapat mempertimbangkan hasil *wordcloud* sebagai perbaikan program maupun acara untuk meningkatkan minat penonton televisi.
2. Untuk penelitian selanjutnya, penelitian serupa dapat dikembangkan dengan menggunakan API *Stream* dan dapat dibuat program untuk otomatisasi klasifikasi. Sehingga hasil analisis sentimen dapat diakses secara *realtime*. Selain itu, daftar kata pada *stopwords* dapat dilengkapi dengan daftar kata singkatan dan daftar kata *slang* dalam bahasa Indonesia.

DAFTAR PUSTAKA

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

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

Bekkar, M., Djemaa, H. K., & Alitouch, T. A. (2013). Evaluation Measure for Models Assesment over Imbalanced Data Sets. *Journal of Information Engineering and Aplications, 3,* 27-38.

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

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

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

Castella, Quim & Sutton, Charles. (2014). Word Storm: Multiples of Word Clouds for Visual Comparison of Documents.

Chomsky, N. (2014, Februari). *What Makes Mainstream Media Mainstream*. Diakses dari Z Magazine: https://zcomm.org/zmagazine/what-makes-*mainstream*-media-*mainstream*/

Dragut, E., Fang, F., Sistla, P., Yu, C., & Meng, W. (2009). Stop Word And Related Problem in *Web* Interface Integration. *VLDB Endowment*.

Falahah & Nur, D. D. A. (2015). Pengembangan Aplikasi Sentiment Analysis Menggunakan Metode Naïve Bayes. *Seminar Nasional Sistem Informasi Indonesia*, 335-340

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

Gokgoz, E., & Subasi, A. (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., Varma P. Saradhi, & Govardhan A. (2012). Preprocessing The Informal Text for Efficient Sentiment Analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 1*, 58-61

Hotho, A., Nurnberger, A., & Paass, G. (2005). *A Brief Survey of Text Mining.* Kassel: University of Kassel.

Kemenkominfo. (2016). *Kominfo: Pengguna Internet di Indonesia 63 Juta Orang*. Diakses pada 20 Januari 2017, dari Kementrian Informasi dan Informatika Republik Indonesia: https://kominfo.go.id/ind ex.php/content/detail/3415/Kominfo+%3A+Pengguna+Inte rnet+di+Indonesia+63+Juta+Orang/0/berita_satker

Liu, B. (2010). *Handbook of Natural Lenguage Processing 2nd Edition.* Boca Raton: CRC Press.

Metro TV. (2017). *Metro TV*. Diakses pada 20 Januari 2017, dari Metro TV: http://www. metrotvnews.com/

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

Rish, I. (2006). An Empirical Study of The *Naive Bayes Classifier*. *International Joint Conference on Artificial Intellegence*, 41-46

Sa'diyah, H., & Fadhilah, U. N. (2017). *Bersama Melawan Berita Palsu*. Diakses pada 25 Januari 2017, dari Republika: http://www.republika.co.id/berita/koran/hukum-koran/17/0 1/04/oj8rc6 14-bersama-mel awan-berita-palsu

Siang, J. J. (2005). *Jaringan Syaraf Tiruan dan Pemrogamannya Menggunakan MATLAB.* Yogyakarta: ANDI.

Sun, Y., Kamel, M. S., & Wang, Y. (2006). Boosting for Learning Multiple Classes with Im-balanced Class Distribution. *Sixth*

International Conference on Data Mining (ICDM'06), 421-431.

Tala, F. Z. (2003). *A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia.* Amsterdam: Institute for Logic, Language, and Computation, Universiteit van Amsterdam.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Boston: Pearson Education.

TV One. (2017). *Profil*. Diakses pada 20 Januari 2017, dari TV One: http://www. tvonenews.tv/profil

Twitter. (2016). *Twitter Support*. Diakses pada 22 Januari 2017, dari Twitter: http:// support.twitter.com/

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

Widhianingsih, T. D. A.. (2016). *Aplikasi Text Mining untuk Automasi Klasifikasi Artikel dalam Majalah Online Wanita Menggunakan Naïve Bayes Classifier (NBC) dan Artificial Neural Network (ANN)*. Surabaya: Institut Teknologi Sepuluh Nopember.

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

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1. Ketepatan Klasifikasi Data *Training* Menggunakan SVM Kernel *Linear*

a. TV One

C	Akurasi	G-Mean	AUC
0.001	0.616	0	0.5
0.01	0.76	0.612372436	0.6875
0.1	0.838	0.760345316	0.7890625
1	0.968	0.959720434	0.960294913
10	**0.979**	**0.974963924**	**0.975108225**
100	**0.979**	**0.974963924**	**0.975108225**
1000	**0.979**	**0.974963924**	**0.975108225**

b. Metro TV

C	Akurasi	G-Mean	AUC
0.001	0.828	0	0.5
0.01	0.828	0	0.5
0.1	0.887	0.585682	0.6715116
1	0.978	0.936401	0.9383496
10	**0.988**	**0.971721**	**0.9720256**
100	**0.988**	**0.971721**	**0.9720256**
1000	**0.988**	**0.971721**	**0.9720256**

Lampiran 1. Ketepatan Klasifikasi Data *Training* Menggunakan SVM Kernel *Linear* (Lanjutan)

c. Kompas TV

C	Akurasi	G-Mean	AUC
0.001	0.547	0	0.5
0.01	0.813	0.766287	0.793598
0.1	0.859	0.829905	0.844371
1	0.989	0.988409	0.988428
10	0.989	0.988805	0.988807
100	0.986	0.986785	0.986824
1000	**0.986**	**0.986954**	**0.987013**

Lampiran 2. Ketepatan Klasifikasi Data *Traininig* TV One Menggunakan SVM Kernel RBF

C	Gamma	Akurasi	G-Mean	AUC
0.01	0.01	0.616	0	0.5
0.01	0.1	0.616	0	0.5
0.01	1	0.76	0.612372	0.6875
0.01	10	0.76	0.612372	0.6875
0.01	100	0.76	0.612372	0.6875
0.1	0.01	0.616	0	0.5
0.1	0.1	0.764	0.620819	0.692708
0.1	1	0.835	0.75519	0.785156
0.1	10	0.821	0.730653	0.766927
0.1	100	0.821	0.730653	0.766927
1	0.01	0.764	0.620819	0.692708
1	0.1	0.843	0.76886	0.795573
1	1	0.97	0.960718	0.961428
1	10	0.968	0.957427	0.958333
1	100	0.968	0.957427	0.958333
10	0.01	0.845	0.77224	0.798177
10	0.1	0.978	0.973106	0.973316
10	**1**	**0.979**	**0.973363**	**0.973637**
10	10	0.968	0.957427	0.958333
10	100	0.968	0.957427	0.958333
100	0.01	0.977	0.971776	0.972014
100	0.1	0.98	0.975761	0.97592
100	**1**	**0.979**	**0.973363**	**0.973637**
100	10	0.968	0.957427	0.958333
100	100	0.968	0.957427	0.958333

Lampiran 3. Ketepatan Klasifikasi Data Metro TV Menggunakan SVM Kernel RBF

C	Gamma	Akurasi	G-Mean	AUC
0.01	0.01	0.828	0	0.5
0.01	0.1	0.828	0	0.5
0.01	1	0.828	0	0.5
0.01	10	0.828	0	0.5
0.01	100	0.828	0	0.5
0.1	0.01	0.828	0	0.5
0.1	0.1	0.828	0	0.5
0.1	1	0.887	0.585682	0.671512
0.1	10	0.887	0.585682	0.671512
0.1	100	0.887	0.585682	0.671512
1	0.01	0.828	0	0.5
1	0.1	0.896	0.628768	0.697674
1	1	0.986	0.958439	0.959302
1	10	0.986	0.958439	0.959302
1	100	0.986	0.958439	0.959302
10	0.01	0.901	0.651474	0.712209
10	0.1	0.989	0.969909	0.970326
10	**1**	**0.991**	**0.973486**	**0.973837**
10	10	0.986	0.958439	0.959302
10	100	0.986	0.958439	0.959302
100	0.01	0.987	0.968735	0.969119
100	0.1	0.988	0.971721	0.972026
100	**1**	**0.991**	**0.973486**	**0.973837**
100	10	0.986	0.958439	0.959302
100	100	0.986	0.958439	0.959302

Lampiran 4. Ketepatan Klasifikasi Data *Training* KompasTV Menggunakan SVM Kernel RBF

C	Gamma	Akurasi	G-Mean	AUC
0.01	0.01	0.547	0	0.5
0.01	0.1	0.547	0	0.5
0.01	1	0.709	0.59801	0.678808
0.01	10	0.812	0.764846	0.792494
0.01	100	0.812	0.764846	0.792494
0.1	0.01	0.547	0	0.5
0.1	0.1	0.816	0.770597	0.796909
0.1	1	0.843	0.808345	0.826711
0.1	10	0.842	0.806978	0.825607
0.1	100	0.842	0.806978	0.825607
1	0.01	0.816	0.770597	0.796909
1	0.1	0.903	0.886494	0.892936
1	1	0.991	0.990016	0.990066
1	10	0.98	0.977676	0.977925
1	100	0.979	0.976546	0.976821
10	0.01	0.91	0.895167	0.900662
10	0.1	0.991	0.990628	0.990635
10	**1**	**0.993**	**0.992244**	**0.992274**
10	10	0.98	0.977676	0.977925
10	100	0.979	0.976546	0.976821
100	0.01	0.99	0.989717	0.989721
100	0.1	0.99	0.9901	0.990101
100	**1**	**0.993**	**0.992244**	**0.992274**
100	10	0.98	0.977676	0.977925
100	100	0.979	0.976546	0.976821

Lampiran 5. Ketepatan Klasifikasi Data *Testing* Menggunakan SVM Kernel *Linear*

a. TV One

C	Akurasi	G-Mean	AUC
0.001	0.59	0	0.5
0.01	0.59	0	0.5
0.1	0.71	0.541002	0.646341
1	0.83	0.78272	0.800124
10	**0.84**	**0.80511**	**0.81604**
100	0.84	0.80511	0.81604
1000	0.84	0.80511	0.81604

b. Metro TV

C	Akurasi	G-Mean	AUC
0.001	0.93	0	0.5
0.01	0.93	0	0.5
0.1	0.95	0.5345225	0.6428571
1	0.97	0.7559289	0.7857143
10	0.95	0.7477565	0.7749616
100	**0.96**	**0.751854**	**0.780338**
1000	**0.96**	**0.751854**	**0.780338**

Lampiran 5. Ketepatan Klasifikasi Data *Training* Menggunakan SVM Kernel *Linear* (Lanjutan)

c. Kompas TV

C	Akurasi	G-Mean	AUC
0.001	0.56	0	0.5
0.01	0.56	0	0.5
0.1	0.83	0.834523	0.848214
1	**0.85**	**0.854704**	**0.856331**
10	0.85	0.853279	0.853896
100	0.85	0.853279	0.853896
1000	0.85	0.853279	0.853896

Lampiran 6. Ketepatan Klasifikasi Data *Testing* TV One Menggunakan SVM Kernel RBF

C	Gamma	Akurasi	G-Mean	AUC
0.01	0.01	0.59	0	0.5
0.01	0.1	0.59	0	0.5
0.01	1	0.59	0	0.5
0.01	10	0.59	0	0.5
0.01	100	0.59	0	0.5
0.1	0.01	0.59	0	0.5
0.1	0.1	0.59	0	0.5
0.1	1	0.59	0	0.5
0.1	10	0.71	0.541002	0.646341
0.1	100	0.71	0.541002	0.646341
1	0.01	0.59	0	0.5
1	0.1	0.59	0	0.5
1	1	0.84	0.780869	0.804878
1	10	0.79	0.69843	0.743902
1	100	0.79	0.69843	0.743902
10	0.01	0.75	0.624695	0.695122
10	**0.1**	**0.85**	**0.812266**	**0.824514**
10	1	0.83	0.774223	0.796403
10	10	0.79	0.69843	0.743902
10	100	0.79	0.69843	0.743902
100	**0.01**	**0.85**	**0.812266**	**0.824514**
100	**0.1**	**0.85**	**0.812266**	**0.824514**
100	1	0.83	0.774223	0.796403
100	10	0.79	0.69843	0.743902
100	100	0.79	0.69843	0.743902

Lampiran 7. Ketepatan Klasifikasi Data *Testing* Metro TV Menggunakan SVM Kernel RBF

C	Gamma	Akurasi	G-Mean	AUC
0.01	0.01	0.93	0	0.5
0.01	0.1	0.93	0	0.5
0.01	1	0.93	0	0.5
0.01	10	0.93	0	0.5
0.01	100	0.93	0	0.5
0.1	0.01	0.93	0	0.5
0.1	0.1	0.93	0	0.5
0.1	1	0.95	0.534522	0.642857
0.1	10	0.95	0.534522	0.642857
0.1	100	0.95	0.534522	0.642857
1	0.01	0.93	0	0.5
1	0.1	0.95	0.534522	0.642857
1	1	0.95	0.534522	0.642857
1	10	0.95	0.534522	0.642857
1	100	0.95	0.534522	0.642857
10	0.01	0.95	0.534522	0.642857
10	**0.1**	**0.97**	**0.75593**	**0.78571**
10	1	0.96	0.654654	0.714286
10	10	0.95	0.534522	0.642857
10	100	0.95	0.534522	0.642857
100	**0.01**	**0.97**	**0.75593**	**0.78571**
100	0.1	0.96	0.751854	0.780338
100	1	0.96	0.654654	0.714286
100	10	0.95	0.534522	0.642857
100	100	0.95	0.534522	0.642857

Lampiran 8. Ketepatan Klasifikasi Data *Testing* Kompas TV Menggunakan SVM Kernel RBF

C	Gamma	Akurasi	G-Mean	AUC
0.01	0.01	0.56	0	0.5
0.01	0.1	0.56	0	0.5
0.01	1	0.56	0	0.5
0.01	10	0.56	0	0.5
0.01	100	0.56	0	0.5
0.1	0.01	0.56	0	0.5
0.1	0.1	0.56	0	0.5
0.1	1	0.83	0.834523	0.836039
0.1	10	0.83	0.834523	0.848214
0.1	100	0.83	0.834523	0.848214
1	0.01	0.56	0	0.5
1	0.1	0.83	0.834523	0.848214
1	**1**	**0.87**	**0.876275**	**0.883929**
1	**10**	**0.87**	**0.876275**	**0.883929**
1	**100**	**0.87**	**0.876275**	**0.883929**
10	0.01	0.83	0.834523	0.848214
10	0.1	0.85	0.853279	0.853896
10	1	0.86	0.86626	0.872565
10	**10**	**0.87**	**0.876275**	**0.883929**
10	**100**	**0.87**	**0.876275**	**0.883929**
100	0.01	0.85	0.853279	0.853896
100	0.1	0.85	0.853279	0.853896
100	1	0.86	0.86626	0.872565
100	**10**	**0.87**	**0.876275**	**0.883929**
100	**100**	**0.87**	**0.876275**	**0.883929**

Lampiran 9. *Syntax Crawling* Data Menggunakan RStudio

```
#ID Twitter API
consumer_key <- 'your consumer key'
consumer_secret <- 'your consumer secret'
access_token <- 'your access token'
access_secret <- 'your access secret'

#Login Twitter API
setup_twitter_oauth(consumer_key, consumer_secret, access_token,
access_secret)

#Searching tweet
tvone <- searchTwitter('tv+one', lang="id", n=15000,
resultType="recent")
write.csv(twListToDF(tvone), file="tvone.csv")

metrotv <- searchTwitter('metro+tv', lang="id", n=15000,
resultType="recent")
write.csv(twListToDF(metrotv), file="metrotv.csv")

kompastv <- searchTwitter('kompas+tv', lang="id", n=15000,
resultType="recent")
write.csv(twListToDF(kompastv), file="kompastv.csv")
```

Lampiran 10. *Syntax* Input dan Praproses Data Menggunakan Python 2.7

```
import pandas as pd
import string
import nltk
import re
import sys
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Import Data
df = pd.read_csv('D:/Data TA/tvonetraining.csv')
text_train = df['tweet'] #ambil kolom text
df_train = text_train
df_label = df['class'] #ambil kolom class
print df.head()
test = pd.read_csv('D:/Data TA/tvonetesting.csv')
text_test = test['tweet'] #ambil kolom text
df_test = text_test
df_test_label = test['class'] #ambil kolom class

# Menghapus Link
trainnolink = []
for line in df_train:
    result = re.sub(r"http\S+", "", line)
    trainnolink.append(result)
testnolink = []
for line in df_test:
    result = re.sub(r"http\S+", "", line)
    testnolink.append(result)
```

Lampiran 10. *Syntax* Input dan Praproses Data Menggunakan Python 2.7 (Lanjutan)

```
# Menghapus Simbol Retweet
trainnort = []
for line in trainnolink:
    result = re.sub(r"RT", "", line)
    trainnort.append(result)
testnort = []
for line in testnolink:
    result = re.sub(r"RT", "", line)
    testnort.append(result)

# Menghapus Username
trainnousername = []
for line in trainnort:
    result = re.sub(r"@\S+", "", line)
    trainnousername.append(result)
testnousername = []
for line in testnort:
    result = re.sub(r"@\S+", "", line)
    testnousername.append(result)

# Case Folding
train_lower = []
for line in trainnousername:
    a = line.lower()
    train_lower.append(a)
test_lower = []
for line in testnousername:
    a = line.lower()
    test_lower.append(a)
```

Lampiran 10. *Syntax* Input dan Praproses Data Menggunakan Python 2.7 (Lanjutan)

```
# Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
train_stemmed = map(lambda x: stemmer.stem(x), train_lower)
train_no_punc = map(lambda x: x.lower().translate(None,
string.punctuation), train_stemmed)
test_stemmed = map(lambda x: stemmer.stem(x), test_lower)
test_no_punc = map(lambda x: x.lower().translate(None,
string.punctuation), test_stemmed)

# Menghapus Stopwords
stopword = open("D:/Data TA/stopword tvone.txt","r").read()
trainfinal = []
for line in train_no_punc:
    word_token = nltk.word_tokenize(line) # get word token for
every line (split line into each separate words)
    word_token = [word for word in word_token if not word in
stopword and not word[0].isdigit()] # remove indonesian stop words
and number
    trainfinal.append(" ".join(word_token))
testfinal = []
for line in test_no_punc:
    word_token = nltk.word_tokenize(line) # get word token for
every line (split line into each separate words)
    word_token = [word for word in word_token if not word in
stopword and not word[0].isdigit()] # remove indonesian stop words
and number
    testfinal.append(" ".join(word_token))
```

Lampiran 11. *Syntax* Klasifikasi Data Menggunakan Python 2.7

```
import codecs
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn import cross_validation
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn import model_selection

def learn_model(training_data, training_label, classifier):
    count_vectorizer = CountVectorizer(binary=True)
    train = count_vectorizer.fit_transform(training_data)
    tfidf_train = TfidfTransformer(use_idf=True).fit_transform(train)
    data_train, data_test, target_train, target_test =
   cross_validation.train_test_split(tfidf_train, training_label,
   test_size=0.2, random_state=43)
    classify = classifier.fit(data_train, target_train)
    scores = cross_validation.cross_val_score(classify, data_test,
   target_test, cv=10)
    return scores.mean()
    # print("Accuracy with 10-fold validation: %0.2f (+/- %0.2f)" %
(scores.mean(), scores.std() * 2))
```

Lampiran 11. *Syntax* Klasifikasi Data Menggunakan Python 2.7 (Lanjutan)

```
def predict(training_data, test_data, test_label, classifier):
    # method for predicting new data
    count_vectorizer = CountVectorizer(binary=True)
    count_vectorizer.fit_transform(training_data)
    test_data = count_vectorizer.transform(test_data)
    test_data_clean = TfidfTransformer(use_idf=True)
   .fit_transform(test_data)
    prediction = classifier.predict(test_data_clean)
    classification_report(test_label, prediction)
    acc = accuracy_score(test_label, prediction)
    return acc
    # print "The accuracy score of new data is
{:.2%}".format(accuracy_score(test_label, prediction))

# Naïve Bayes Classifier
nb_classifier = BernoulliNB()
#10-fold cross validation
learn_model(trainfinal, df_label, nb_classifier)
#Predict new testing data
predict(trainfinal, testfinal, df_test_label, nb_classifier)
predicted = testfinal
df_test = predicted
count_vectorizer = CountVectorizer(binary=True)
count_vectorizer.fit_transform(trainfinal)
test_data = count_vectorizer.transform(df_test)
test_data_clean =
TfidfTransformer(use_idf=True).fit_transform(test_data)
predicted1 = nb_classifier.predict(test_data_clean)
prediction = pd.DataFrame(predicted1, columns =
['predictions']).to_csv('D:/tvone nb.csv')
```

Lampiran 11. *Syntax* Klasifikasi Data Menggunakan Python 2.7 (Lanjutan)

```
#Support Vector Machine
svm_classifier = SVC(kernel='rbf', C=1, gamma=1)
#10-fold cross validation
learn_model(trainfinal, df_label, svm_classifier)
#Predict new testing data
predict(trainfinal, testfinal, df_test_label, svm_classifier)
predicted = testfinal
df_test = predicted
print len(df_test)
count_vectorizer = CountVectorizer(binary=True)
count_vectorizer.fit_transform(trainfinal)
test_data = count_vectorizer.transform(df_test)
test_data_clean =
TfidfTransformer(use_idf=True).fit_transform(test_data)

predicted = svm_classifier.predict(test_data_clean)
prediction = pd.DataFrame(predicted,
columns=['predictions']).to_csv('D:/tvone svm rbf c1 g1.csv')
```

Lampiran 12. *Syntax Word Cloud* Menggunakan RStudio

```
#Import data tweet
metrotvtraining <- read.csv("d:/metrotv-training.csv")
metrotvtesting <- read.csv("d:/metrotv-testing.csv")
metrotv <- read.csv("d:/Data TA/metrotv.csv")
metrotvCorpus <- Corpus(VectorSource(metrotv[,2]))
class(metrotvtraining)
inspect(metrotvCorpus[1])

#Preprocessing
#Menghapus URL, rt, username
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
metrotv_clean <- tm_map(metrotvCorpus, removeURL)
removeRT <- function(y) gsub("RT", "", y)
metrotv_clean <- tm_map(metrotv_clean, removeRT)
removeUN <- function(z) gsub("@\\w+", "", z)
metrotv_clean <- tm_map(metrotv_clean, removeUN)
#Remove punctuation, to lower
metrotv_clean <- tm_map(metrotv_clean, removePunctuation)
metrotv_clean <- tm_map(metrotv_clean, tolower)

#Menghapus stopwords
file_stop <- file("stopword.txt",open = "r")
id_stopwords <- readLines(file_stop)
close(file_stop)
id_stopwords = c(id_stopwords, "amp")
metrotv_clean <- tm_map(metrotv_clean, removeWords,
id_stopwords)
#Menghapus nomor, strip white space
metrotv_clean <- tm_map(metrotv_clean, removeNumbers)
metrotv_clean <- tm_map(metrotv_clean, stripWhitespace)
metrotv_clean <- tm_map(metrotv_clean, PlainTextDocument)
inspect(metrotv_clean[1:5])

metrotv_clean <- tm_map(metrotv_clean, removeWords, c("tv",
"metro", "metrotv"))
```

Lampiran 12. *Syntax Word Cloud* Menggunakan RStudio (Lanjutan)

```
#Stemming
stem_text <- function(text,mc.cores=1) {
  #stem each word in a block of text
  stem_string <- function(str)
  {
    str <- tokenize(x=str)
    str <- sapply(str, katadasaR)
    str <- paste(str, collapse="")
    return(str)
  }
  #stem each text block in turn
  x <- mclapply(X=text, FUN=stem_string, mc.cores=mc.cores)
  #return stemed text blocks
  return(unlist(x))
}
metrotv_clean <- tm_map(metrotv_clean, stem_text)

#Tokenizing
tokenizing <- function(x) strsplit(x, " ")
metrotv_clean <- tm_map(metrotv_clean, tokenizing)
#Stemming
stemming <- function(w) sapply(w, katadasaR)
metrotv_clean <- tm_map(metrotv_clean, stemming)

#Wordcloud
wordcloud(metrotv_clean)
wordcloud(metrotv, random.order=F)
wordcloud(metrotv_clean, random.order=F, scale=c(3, 0.5))
wordcloud(metrotv_clean, random.order=F,min.freq=10,
colors=brewer.pal(8, "Dark2"))
```

Lampiran 13. Surat Pernyataan Data

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMIPA ITS:

Nama : Taufik Kurniawan
NRP : 1313100075

Menyatakan bahwa data yang digunakan dalam Tugas Akhir ini merupakan data yang diambil dari:

Sumber : Twitter API (*Application Program Interface*)
Keterangan : Data *tweet* dengan *keyword* "tv one", "metro tv", dan "kompas tv"

Surat ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Surabaya, 9 Juni 2017

(Taufik Kurniawan)
NRP. 1313100075

Mengetahui,

Pembimbing Tugas Akhir	Co Pembimbing Tugas Akhir
(Dr. Kartika Fithriasari, M.Si.)	(Dr. Irhamah, S.Si., M.Si.)
NIP. 19691212 199303 2 001	NIP. 19790406 200112 2 002

BIODATA PENULIS

Penulis lahir di Sukoharjo pada 27 Juli 1996 dengan nama Taufik Kurniawan sebagai anak ketiga dari tiga bersaudara. Penulis telah menempuh pendidikan formal dimulai dari TK Dharma Wanita 2 Sukoharjo, SD Negeri Mulur 3, SMP Negeri 1 Sukoharjo, SMA Negeri 1 Sukoharjo dan melanjutkan studinya di Program Sarjana Jurusan Statistika ITS melalui jalur SBMPTN. Selama masa perkuliahan, penulis pernah aktif dalam organisasi HIMASTA-ITS sebagai staff Departemen Keilmiahan dan JMMI-ITS sebagai staff Kaderisasi. Penulis tergabung dalam lembaga dakwah jurusan FORSIS-ITS pada dua periode kepengurusan. Selain itu, penulis juga aktif dalam kepanitiaan Bina Cinta Statistika selama tiga periode. Penulis juga berkesempatan menjadi mentor sebagai media dakwah di kampus ITS. Jika ingin berdiskusi lebih lanjut mengenai Tugas Akhir penulis, dapat menghubungi penulis melalui email berikut: kurniawantfk27@gmail.com.