TUGAS AKHIR – KS184822

KLASIFIKASI SENTIMEN PENUMPANG PESAWAT MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE* (STUDI KASUS MASKAPAI PENERBANGAN PT. X)

BIMA PUTRA GOKLAS
NRP 062116 4000 0124

Dosen Pembimbing
Irhamah, S.Si., M.Si., Ph.D.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

TUGAS AKHIR – KS184822

KLASIFIKASI SENTIMEN PENUMPANG PESAWAT MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE* (STUDI KASUS MASKAPAI PENERBANGAN PT. X)

BIMA PUTRA GOKLAS
NRP 062116 4000 0124

Dosen Pembimbing
Irhamah, S.Si., M.Si., Ph.D.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

FINAL PROJECT – KS184822

CLASSIFICATION OF PASSENGER AIRCRAFT SENTIMENTS USING NAÏVE BAYES CLASSSIFIER AND SUPPORT VECTOR MACHINE METHOD (STUDY CASE PT. X AIRLINE COMPANY)

BIMA PUTRA GOKLAS
SN 062116 4000 0124

Supervisor
Irhamah, S.Si., M.Si., Ph.D.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF SCIENCE AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

KLASIFIKASI SENTIMEN PENUMPANG PESAWAT MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE* (STUDI KASUS MASKAPAI PENERBANGAN PT. X)

TUGAS AKHIR

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

Oleh:
Bima Putra Goklas
NRP. 062116 4000 0124

Disetujui oleh Pembimbing Tugas Akhir:
Irhamah, S.Si., M.Si., Ph.D. ( )
NIP. 19780406 200112 2 002

Mengetahui,
Kepala Departemen

Dr. Dra. Kartika Fithriasari, M.Si
NIP. 19691212 199303 2 002

SURABAYA, JULI 2020

(Halaman ini sengaja dikosongkan)

KLASIFIKASI SENTIMEN PENUMPANG PESAWAT MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE* (STUDI KASUS MASKAPAI PENERBANGAN PT. X)

Nama Mahasiswa : Bima Putra Goklas
NRP : 062116 4000 0124
Departemen : Statistika-FMKSD-ITS
Dosen Pembimbing : Irhamah, S.Si., M.Si., Ph.D.

Abstrak

PT X sebagai perusahaan penyedia jasa transportasi penerbangan perlu menjaga kualitas layanannya. Untuk mengukur kinerja perusahaan, PT X kerap meminta ulasan penumpang untuk menanggapi layanan yang diberikan. Seiring perusahaan yang terus berkembang dan penumpang yang semakin banyak, membaca ulasan secara keseluruhan tentu membutuhkan waktu yang lama. Untuk itu perlu usaha untuk mengumpulkan ulasan tersebut dan mengolahnya menjadi informasi yang bermanfaat bagi perusahaan, yaitu dengan memanfaatkan teknik analisis sentimen. Penelitian ini bertujuan untuk memprediksi sentimen penumpang dari ulasan yang diberikan menggunakan metode klasifikasi. Naïve Bayes Classifier (NBC) dan Support Vector Machine (SVM) adalah dua metode yang digunakan pada penelitian ini. Sebelum dilakukan klasifikasi, data terlebih dahulu dilakukan praproses teks. Apabila data menunjukkan imbalanced pada masing-masing kategori, maka dilakukan teknik SMOTE untuk mengatasinya. Data yang digunakan pada penelitian ini adalah data sekunder yang merupakan data ulasan penumpang domestik tahun 2019. Ulasan akan diklasifikasikan menjadi dua kategori sentimen, yaitu sentimen positif dan negatif. Hasil penelitian menunjukkan aplikasi SMOTE pada data dapat meningkatkan nilai ketepatan klasifikasi. Akurasi dan AUC data training terbaik ditunjukkan oleh metode SVM kernel RBF, sedangkan akurasi dan AUC data testing terbaik ditunjukkan oleh metode SVM kernel linear.

Kata Kunci : Analisis Sentimen, Klasifikasi, Naïve Bayes Classifier, SMOTE, Support Vector Machine

(Halaman ini sengaja dikosongkan)

CLASSIFICATION OF PASSENGER AIRCRAFT SENTIMENTS USING NAÏVE BAYES CLASSSIFIER AND SUPPORT VECTOR MACHINE METHOD (STUDY CASE PT. X AIRLINE COMPANY)

Name : **Bima Putra Goklas**
Student Number : **062116 4000 0124**
Department : **Statistics**
Supervisors : **Irhamah, S.Si., M.Si., Ph.D.**

Abstract

PT X as a provider of aviation transportation services needs to maintain the quality of its services. To measure company performance, PT X often requests passenger reviews to respond to services provided. As the company continues to grow and passengers increasingly, reading the overall review certainly takes a long time. For this reason, an effort is needed to collect these reviews and process them into useful information for companies, namely by utilizing sentiment analysis techniques. This study aims to predict passenger sentiment from the reviews provided using the classification method. Naïve Bayes Classifier (NBC) and Support Vector Machine (SVM) are two methods used in this study. Before classification, the text data is preprocessed. If the data shows imbalanced in each category, then the SMOTE technique is used to overcome them. The data used in this study are secondary data which are data on domestic passenger reviews in 2019. Reviews will be classified into two categories of sentiments, namely positive and negative sentiments. The results showed the application of SMOTE to the data could increase the value of classification performance. The best accuracy and AUC for training data is shown by the SVM kernel RBF method, while the best accuracy and AUC for testing data is shown by the linear SVM kernel method.

Keywords: Sentiment Analysis, Classification, Naïve Bayes Classifier, SMOTE, Support Vector Machine

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala kasih dan penyertaanNya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini yang berjudul **"Klasifikasi Sentimen Penumpang Pesawat Menggunakan Metode *Naïve Bayes Classifier* dan *Support Vector Machine* (Studi Kasus Maskapai Penerbangan PT. X)"** dengan tepat waktu.

Penulis menyadari dalam penyusunan Tugas Akhir ini tidak akan selesai tanpa bantuan maupun dukungan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan terima kasih kepada:

1. Orang Tua (Papa & Mama tercinta) dan adik yang selalu memberikan doa dan dukungan selama penyusunan Tugas Akhir.
2. Ibu Irhamah, S.Si., M.Si., Ph.D. selaku dosen pembimbing yang telah memberikan bimbingan, saran, serta motivasi selama penyusunan Tugas Akhir berlangsung.
3. Prof. Drs. NUR Iriawan, MIKom., Ph.D. dan Dr. Dra. Kartika Fithriasari, M.Si. selaku dosen penguji yang telah memberikan masukan dan bantuan dalam menyelesaikan Tugas Akhir.
4. Bapak Dr. R. Mohamad Atok, S.Si., M.Si. dan Dr. Achmad Choiruddin, S.Si., M.Sc. selaku dosen wali yang telah banyak memberikan saran dan arahan dalam proses belajar selama ini di Departemen Statistika.
5. Dr. Dra. Kartika Fithriarsari M.Si selaku Kepala Departemen Statistika dan Vita Ratnasari, S.Si., M.Si. dan Dr. Santi Wulan Purnami, S.Si., M.Si. selaku Sekretaris Departemen Statistika FSAD ITS.
6. Bapak Swastika Harimurti dan Kang Decky Prasakti dari unit *Corporate Research Development* PT X yang telah banyak membantu dalam pengerjaan Tugas Akhir dan mendapatkan data penelitian.
7. Syifa K. Gunawan yang selalu memberikan support dan semangat dalam penyusunan Tugas Akhir.

8. Teman-teman seperjuangan TA dalam bimbingan Ibu Irhamah khususnya Inan, Rachel, dan Naufal yang menemani penulis simulasi presentasi Tugas Akhir dan sebagai teman diskusi serta teman-teman TR16GER lainnya yang selalu memberikan semangat kepada penulis dalam penyusunan Tugas Akhir.
9. Seluruh pihak yang turut membantu dalam penyelesaian laporan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari masih banyak kekurangan dalam pembuatan laporan Tugas Akhir ini. Penulis berharap semoga laporan Tugas Akhir ini dapat bermanfaat dan menambah wawasan bagi pembaca. Kritik dan saran sangat diperlukan untuk perbaikan di masa yang akan datang.

Surabaya, Juli 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Transportasi merupakan salah satu komponen utama dalam sistem hidup dan kehidupan, sistem pemerintahan, dan sistem kemasyarakatan. Kondisi sosial demografi suatu wilayah memiliki pengaruh terhadap kinerja transportasi di wilayah tersebut. Tingkat kepadatan penduduk akan memiliki pengaruh signifikan terhadap kemampuan transportasi dalam melayani kebutuhan masyarakat (Susantoro & Parikesit, 2004). Indonesia merupakan negara kepulauan yang memiki lebih dari 17.000 pulau dengan total wilayah 735.355 mil persegi. Dengan jumlah penduduk lebih dari 250 juta jiwa, Indonesia menempati peringkat keempat dari 10 negara berpopulasi terbesar di dunia. Tanpa sarana transportasi yang memadai, tentu akan sulit untuk menghubungkan seluruh wilayah yang ada di Indonesia.

Kebutuhan akan transportasi merupakan kebutuhan turunan (*derived demand*) yang diakibatkan oleh aktivitas ekonomi, sosial masyarakat, dan lainnya. Dalam aspek makro-ekonomi, transportasi merupakan salah satu tulang punggung perekonomian nasional, baik di level regional maupun lokal. Perkembangan peradaban manusia khususnya dalam bidang teknologi telah membawa peradaban manusia kedalam suatu sistem transportasi yang lebih maju dibandingkan dengan era sebelumnya (Adji, 2005). Perkembangan tersebut tentu membawa dampak positif bagi pemakai jasa transportasi berupa kemudahan dan kenyamanan dalam berpindah tempat dari suatu tempat ke tempat yang lain. Untuk memenuhi kebutuhan mobilisasi masyarakat, Indonesia membutuhkan sarana transportasi baik darat, laut, maupun udara (Suriaatmaja, 2005). Dari ketiga jasa angkutan yang ditawarkan tersebut, jasa transportasi udara memang yang paling terakhir berkembang dan kini menjadi andalan bagi beberapa pelaku usaha yang bergerak dalam jasa transportasi (Abdulkadir, 1991). Perkembangan dan pertumbuhan industri penerbangan tersebut tidak lepas dari peningkatan jumlah pengguna jasa transportasi udara yang juga mengalami perkembangan pesat. Pada era modern

seperti saat ini, penerbangan merupakan salah satu transportasi yang sudah banyak digunakan oleh masyarakat. Hal ini dikarenakan oleh kebutuhan masyarakat terhadap transportasi jarak jauh dengan waktu singkat sudah cukup tinggi. Selain itu, harga dari transportasi penerbangan juga sudah relatif terjangkau oleh masyarakat Indonesia. Ada beberapa alasan konsumen menggunakan jasa transportasi udara, diantaranya untuk kepentingan bisnis, kepentingan parwisata, dan berbagai kepentingan lainnya. Menurut data Badan Pusat Statistik (2018) menujukkan bahwa pada tahun 2018 jumlah penumpang yang menggunakan jasa transportasi untuk penerbangan domestik mencapai 101.260.614 penumpang. Jumlah tersebut mengalami peningkatan sebesar 6,14% dibandingkan jumlah penumpang tahun 2017 dan meningkat sebesar 37,04% selama 5 tahun terakhir.

Apabila berbicara mengenai tranportasi udara, tentu kita tidak bisa lepas dari perusahaan penerbangan yang menjadi basis penyedia jasa transportasi udara. Sedikit melihat kilas balik perkembangan dunia aviasi di Indonesia, semenjak Indonesia memasuki era reformasi hingga sekarang ini, kebijakan transportasi udara cenderung liberal. Perusahan penerbangan tumbuh dengan pesat, pada tahun 2004 terdapat 103 perusahaan penerbangan milik pemerintah bersama milik swasta. Dengan keluarnya Keputusan Menteri Perhubungan Nomor KM 11 2001 yang disempurnakan dengan Keputusan Menteri Perhubungan Nomor KM 81 Tahun 2004, yang mengatur tentang angkutan udara niaga (*commercial airlines*) dan bukan niaga (*general aviation*), jumlah perusahaan penerbangan meningkat lagi dari 103 perusahaan pada tahun 2004 menjadi 157 perusahaan penerbangan yang terdiri dari atas perusahaan penerbangan milik pemerintah, swasta, dan penerbangan umum. Akibat dinamisasi kebijakan transportasi udara yang cenderung liberal ini perusahaan penerbangan terpaksa bersaing secara keras dan ketat, seperti saling menurunkan tarif batas bawah, sehingga secara langsung mereka saling mematikan perusahaan penerbangan lain yang tak sanggup mengimbangi persaingan ditengah biaya operasional (*operational cost*) yang semakin tinggi. Untuk itu disempurnakanlah Keputusan Menteri Perhubungan Nomor KM

81 Tahun 2004 menjadi Keputusan Menteri Perhubungan Nomor KM 25 Tahun 2008 yang selanjutnya diterbitkan Undang-Undang No. 1 Tahun 2009 tentang Penerbangan (Martono & Sudiro, 2010). Undang-undang tersebut menggantingan Undang-Undang No. 16 Tahun 1992 tentang Penerbangan yang dirasa sudah tidak sesuai lagi dengan kondisi, perubahan lingkungan strategis, dan kebutuhan penyelanggaraan penerbangan.

Setelah dilakukan revisi undang-undang tersebut, per tahun 2018 terdapat 62 perusahaan penerbangan komersil di Indonesia yang tersertifikasi AOC (*Air Operator Certificate*) 121/135 (BPS, 2018). Dikutip dari Lampiran Peraturan Menteri Perhubungan Nomor PM 61 (2017), AOC 121 merupakan sertifikasi yang diberikan kepada maskapai yang mengoperasikan pesawat komersil dengan muatan lebih dari 30 kursi, sedangkan AOC 135 merupakan sertifikasi yang diberikan kepada maskapai yang mengoperasikan pesawat/helikopter komersil dengan muatan kurang dari 30 kursi. Salah satu perusahaan penerbangan yang tersertifikasi AOC 121 adalah PT X. PT X merupakan maskapai penerbangan yang beroperasi sejak tahun 1949. PT X tercatat di Bursa Efek Indonesia sebagai perusahaan publik pada Februari 2011. Sebagai perusahaan penerbangan komersil, PT X tentu tak akan lepas dari hal yang berkaitan dengan pelayanan terhadap konsumen dan perlu menaruh perhatian khusus terhadap kualitas layanannya. Salah satu alasan PT X perlu menaruh perhatian khusus terhadap kualitas layanan adalah dikarenakan kualitas layanan merupakan salah satu faktor yang vital dalam menciptakan *superior value* untuk pelanggan (Musnaini, 2011). Menurut Menon, Jaworski, dan Kohli (1997), terciptanya *superior value* bagi pelanggan merupakan suatu batu loncatan bagi perusahaan untuk memperoleh keunggulan kompetitif. Sedangkan Drage, Vickery, dan Markland (1995) dalam Fanny (2006) berpendapat bahwa keunggulan kompetitif yang dimiliki perusahaan pada akhirnya akan mempengaruhi kinerja perusahaan.

Seiring perkembangannya, PT X pernah jatuh dan kemudian bangkit kembali. Sepanjang tahun 2019 PT X sempat dilanda berbagai isu hingga cukup disorot oleh masyarakat dan berbagai media massa. Agar tak kehilangan kepercayaan pelanggannya, PT

X harus tetap mempertahankan kualitas layanannya sebagai penyedia layanan transportasi udara komersil, baik dari segi keselamatan, keamanan, maupun kenyamanannya mengingat PT X memiliki segmen yang berbeda dengan perusahaan penerbangan lainnya. Konsep layanan penuh atau biasa disebut *full service* menjadikan PT X memiliki pasar kalangan menengah hingga atas, sebanding dengan harga yang ditawarkan terhadap kualitas pelayanan yang diberikan sebelum penerbangan (*pre-flight*), selama penerbangan (*in-flight*), dan hingga setelah tiba (*post-flight*). Untuk mengetahui kinerja pelayanan penerbangan, PT X secara rutin melakukan survey layanan kepada penumpang. Survei dilakukan secara langsung kepada penumpang, dimana penumpang mengisi *suggestion form* yang diberikan saat penerbangan. Melalui *suggestion form* tersebut penumpang dapat secara bebas memberikan opini/tanggapan/ulasan terhadap pelayanan penerbangan yang diberikan oleh PT X. Tantangan yang dihadapi dari ulasan yang diberikan penumpang adalah data berbentuk teks sehingga harus dibaca satu per satu dan tidak dapat diolah menggunakan metode kuantitatif secara langsung. Membaca ulasan penumpang satu per satu tentu memakan waktu yang lama, terlebih apabila hanya sedikit ulasan yang dibaca, evaluasi yang dihasilkan tentu akan bias. Dari permasalahan tersebut tentu diperlukan usaha untuk mengumpulkan semua ulasan tersebut dan mengolahnya menjadi informasi yang dapat menjawab keingintahuan PT X terhadap ulasan penumpangnya secara cepat dan efisien. Usaha tersebut dapat dicapai dengan memanfaatkan teknik analisis sentimen.

Analisis sentimen atau *opinion mining* merupakan metode analisis berbasis komputasi mengenai pendapat, sentimen, dan emosi yang diekspresikan ke dalam bentuk teks (Liu, 2010). Analisis sentimen akan mengelompokan polaritas dari teks yang ada dalam kalimat atau dokumen, dengan demikian dapat diketahui kecenderungan suatu sentimen atau pendapat, apakah pendapat tersebut cenderung beropini positif atau negatif. Proses ini melibatkan penggunaan Pemrosesan Bahasa Alami (*Natural Language Processing*) atau NLP. Proses penggunaan NLP, penarikan informasi, dan teknik *machine learning* digunakan untuk

mengurai data teks tidak terstruktur ke dalam bentuk lebih terstruktur dan mengambil pola dan wawasan dari data tersebut yang berguna oleh pengguna (Sarkar, 2016). Sebelum melakukan analisis sentimen menggunakan metode klasifikasi teks, terlebih dahulu dilakukan *preprocessing* data dengan metode *text mining* untuk mengolah data teks agar siap untuk dianalisis. *Preprocessing* data teks meliputi *case folding* (merubah semua teks menjadi huruf kecil), *tokenizing* (memecah teks yang berasal dari kalimat menjadi kata per kata), penghapusan *stopwords* (*stopwords* merupakan kosakata yang tidak termasuk dalam kata unik atau ciri dari sebuah dokumen), dan *stemming* (proses mendapatkan kata dasar dengan menghilangkan imbuhan kata) (Hemalatha, Varma, & Govardhan, 2012). Dikarenakan data yang digunakan dalam penelitian ini merupakan data Bahasa Indonesia, maka proses *stemming* akan menggunakan algoritma *Confix-Stripping Stemmer* yang merupakan pengembangan dari algoritma *Nazief and Adriani's Stemmer*. Kedua algoritma diatas merupakan algoritma yang dikembangkan berdasarkan aturan Bahasa Indonesia untuk mendapatkan kata dasar, yaitu dengan menghilangkan imbuhan kata meliputi awalan (*prefix*), sisipan (*infix*), akhiran (*suffixes*), dan kombinasi antara awalan dan akhiran (*confixes*) (Yosephine & Prabowo, 2016).

Setelah dilakukan *preprocessing* pada data, selanjutnya dilakukan klasifikasi sentimen menggunakan metode klasifikasi teks. Terdapat banyak metode klasifikasi yang dapat digunakan untuk analisis sentimen, metode yang sering digunakan adalah *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). Hal serupa juga diungkapkan oleh Rita McCue dalam penelitiannya, yaitu metode *Naïve Bayes Classifier* memiliki beberapa kelebihan antara lain, sederhana, cepat dan berakurasi tinggi (McCue, 2009). Sedangkan metode SVM digunakan karena metode ini sangat cepat dan efektif pada klasifikasi data teks (Feldman & Sanger, 2007). Penggunaan metode NBC dan SVM pada analisis sentimen sudah pernah dilakukan oleh penelitian sebelumnya, diantaranya

penelitian yang dilakukan oleh Rita (2009) mengenai Klasifikasi Spam menggunakan *Support Vector Machine* dan *Naïve Bayes*, menunjukkan bahwa metode SVM dan NBC masing-masing menghasilkan akurasi klasifikasi sebesar 96% dan 97,8% dalam mengklasifikasikan *email* spam. Penelitian dengan metode serupa juga pernah dilakukan oleh Taufik (2017) tentang Analisis Sentimen Pengguna Twitter terhadap Media *Mainstream* menggunakan NBC dan SVM. Secara keseluruhan perbandingan performa metode NBC dan SVM menunjukkan hasil bahwa performa SVM lebih baik dalam mengklasifikasikan data. Penelitian lain juga dilakukan oleh Widhianingsih (2016) berjudul Aplikasi *Text Mining* untuk Automasi Klasifikasi Artikel dalam Majalah Online Wanita Menggunakan *Naïve Bayes Classifier* (NBC) dan *Artificial Neural Network* (ANN). Penelitian yang dilakukan oleh Widhianingsih mengklasifikasikan artikel majalah wanita. Akurasi yang dihasilkan oleh model NBC sebesar 80,71%, model ANN sebesar 75%, dan model Regresi Logistik Multinomial sebesar 57,86%. Selanjutnya terdapat penelitian yang dilakukan oleh Moh. Hasan Basri (2016) yang berjudul Identifikasi Topik Informasi Publik Media Sosial di Kota Surabaya Berdasarkan Klasterisasi Teks Pada *Twitter* dengan Menggunakan Algoritma *K-Means*. Hasil identifikasi topik untuk data keseluruhan didapatkan pelabelan 1, 2, dan 3 berturut-turut memiliki hasil klasifikasi optimal menggunakan algoritma SVM Kernel '*Linear*' dengan akurasi 95,92%, 95,51%, dan 96,79%.

Berdasarkan penelitian yang sudah dilakukan sebelumnya, maka pada penelitian ini akan dilakukan klasifikasi menggunakan metode *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM). Struktur data yang digunakan terdiri dari variabel independen yaitu data ulasan penumpang pesawat PT X yang sudah dilakukan *preprocessing* sebelumnya dan variabel dependen yaitu kategori sentimen ulasan (positif dan negatif). Penelitian ini bertujuan untuk melakukan klasifikasi sentimen ulasan penumpang terhadap kualitas layanan penerbangan yang diberikan oleh PT X. Dengan demikian diharapkan metode klasifikasi yang digunakan pada penelitian ini menghasilkan ketepatan klasifikasi yang tinggi yang diukur menggunakan akurasi, *sensitivity*, dan *specificity*, serta

dapat digunakan oleh PT X untuk mengetahui sentimen penumpang dan dapat dijadikan acuan dalam menilai kinerja pelayanannya.

1.2 Rumusan Masalah

Mengetahui tanggapan penumpang terhadap layanan penerbangan merupakan hal yang penting dan dapat digunakan untuk mengukur kinerja perusahaan. PT X secara rutin meminta ulasan (komentar) kepada penumpang terhadap layanan mereka. Kolom komentar dapat dimanfaatkan penumpang untuk memberikan ulasan mengenai kualitas dan layanan penerbangan yang disediakan oleh PT X. Membaca ulasan tersebut secara keseluruhan tentu membutuhkan waktu yang lama, sedangkan apabila hanya sedikit ulasan yang dibaca akan menimbulkan bias terhadap evaluasi yang dihasilkan. Usaha untuk mengumpulkan semua ulasan tersebut dan mengolahnya menjadi informasi yang dapat bermanfaat bagi perusahaan sangat dibutuhkan dan usaha tersebut dapat dicapai dengan memanfaatkan teknik analisis sentimen. Analisis sentimen akan mengelompokan polaritas dari teks yang ada dalam kalimat atau dokumen. Dengan demikian, kecondongan sebuah teks untuk bersifat positif atau negatif dapat diketahui dengan mudah. 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 kesalahan klasifikasi yang paling kecil.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan, tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Mengetahui karakteristik data ulasan penumpang pesawat terhadap layanan penerbangan PT X.

2. Mendapatkan hasil ketepatan klasifikasi sentimen penumpang pesawat terhadap layanan penerbangan PT X menggunakan metode *Naïve Bayes Classifier*.
3. Mendapatkan hasil ketepatan klasifikasi sentimen penumpang pesawat terhadap layanan penerbangan PT X menggunakan metode *Support Vector Machine*.
4. Membandingkan hasil ketepatan klasifikasi metode *Naïve Bayes Classifier* dengan metode *Support Vector Machine*.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat membantu PT X dalam memahami tanggapan penumpang pesawat mengenai layanan penerbangan yang diberikan PT X secara cepat dan efisien dikarenakan PT X tidak perlu membaca seluruh ulasan satu per satu. Setelah mengetahui sentimen penumpang, diharapkan agar menjadi pertimbangan PT X dalam memperbaiki kualitas layanannya terhadap aspek yang dinilai negatif oleh penumpang serta mempertahankan dan meningkatkan kualitas layanannya terhadap aspek yang dinilai positif oleh penumpang. Manfaat untuk peneliti adalah peneliti mampu menerapkan metode NBC dan SVM pada masalah klasifikasi sentimen dan mengetahui kelebihan serta kekurangan masing-masing metode. Selain itu, penelitian ini juga dapat digunakan sebagai informasi atau wawasan umum dalam penggunaan teknik analisis sentimen dari data teks yang berisi ulasan/komentar.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah penelitian tidak memperhatikan latar belakang atau demografi dari penumpang pesawat. Data yang digunakan merupakan data ulasan penumpang berbahasa Indonesia pada penerbangan domestik selama periode bulan Juli 2019 sampai Desember 2019. Selain itu klasifikasi sentimen data awal ditentukan secara subyektif oleh peneliti.

BAB II
TINJAUAN PUSTAKA

Tinjauan pustaka berisi landasan teori yang dipakai pada penelitian ini. Teori yang digunakan pada penelitian ini berasal dari buku, jurnal ilmiah, dan beberapa penelitian sebelumnya.

2.1 Statistika Deskriptif dan *Word Cloud*

Statistika deskriptif berkenaan dengan deskripsi suatu data seperti halnya menghitung rata-rata dan varians dari data mentah; mendeskripsikan menggunakan tabel atau gambar (grafik) sehingga data lebih bermakna dan mudah dipahami. Sebelum dianalisis lebih lanjut, data penelitian biasanya dilakukan statistika deskriptif untuk mengetahui informasi awal dari data. Statistika deskriptif menunjukkan bagaimana data dapat digambarkan (dideskripsikan) atau disimpulkan baik secara numerik (misal menghitung rata-rata dan deviasi standar) atau secara grafis (dalam bentuk tabel atau grafik) untuk mendapatkan gambaran sekilas mengenai data tersebut sehingga lebih mudah dibaca dan bermakna. Menurut (Walpole R. E., 2007) statistika desktiptif memiliki fungsi untuk memberikan informasi mengenai seputar data tanpa mengambil keputusan atau menarik kesimpulan (inferensia) dari data tersebut.

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 2.1 Contoh Visualisasi Data Teks Menggunakan *Word Cloud*

2.2 Analisis Sentimen pada *Text Mining*

Analisis teks atau bisa juga disebut sebagai penggalian teks (*text mining*) merupakan metodologi dan sebuah proses yang diiringi dengan perolehan kualitas dan informasi yang dapat ditindaklanjuti serta wawasan dari data tekstual. Proses ini melibatkan penggunaan Pemrosesan Bahasa Alami (*Natural Language Processing*) atau NLP. Proses penggunaan NLP, penarikan informasi, dan teknik *machine learning* digunakan untuk mengurai data teks tidak terstruktur ke dalam bentuk lebih terstruktur dan mengambil pola dan wawasan dari data tersebut yang berguna oleh pengguna (Sarkar, 2016).

Text mining merupakan penambangan data berupa teks yang diperoleh dari data tidak terstruktur berupa kalimat-kalimat didalam dokumen, lalu dari dokumen tersebut dicari kata-kata yang dapat mewakili isi dokumen untuk dapat menganalisis inti dari dokumen tersebut. Penjelasan lebih sederhananya, *text mining* adalah proses penyaringan wawasan yang dapat ditindaklanjuti atau dianalisa dari teks. Secara spesifik, *text mining* dapat digunakan untuk mengidentifikasikan kabar dari media sosial yang dapat ditindaklanjuti untuk organisasi layanan pelanggan. *Text mining* mewakili kemampuan untuk mengambil bahasa tidak terstruktur dalam jumlah yang besar dan cepat dalam mengekstrak wawasan yang berguna untuk pengambilan keputusan. Hal-hal

tersebut dilakukan tanpa memaksa seseorang untuk membaca seluruh badan teks (Kwartler, 2017).

Tujuan *text mining* adalah untuk mendeskripsikan dan mengeksplor data tekstual, untuk mengungkap ciri struktural dan memproses prediksi (Lebart, 1998). Perangkat lunak *text mining* dapat digunakan untuk membuat arsip informasi berukuran besar tentang orang atau peristiwa tertentu. Secara umum, *text mining* terbagi menjadi dua tipe, yang pertama disebut "*bag of words*" dan yang satunya lagi disebut "*syntatic parsing*", dengan kelebihan dan kekurangan masing-masing.

Analisis sentimen, juga disebut *opinion mining*, adalah bidang studi yang menganalisa pendapat, sentimen, penilaian, sikap, dan emosi orang terhadap entitas dan diungkapkan dalam teks tertulis. Entitas dari analisis sentimen dapat berupa produk, layanan, organisasi, individu, acara, isu, atau topik (Liu, 2015). Tugas dasar dalam analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen atau kalimat. Sentimen yang dimaksud bisa berupa negatif, positif, netral, dan lain-lain.

Sentimen negatif, positif, atau netral pada umumnya berupa sikap dalam bentuk verbal atau pendapat yang diungkapkan dalam teks terhadap subjek tertentu. Besarnya pengaruh dan manfaat dari analisis sentimen menyebabkan penelitian ataupun aplikasi mengenai analisis sentimen berkembang pesat, bahkan menurut (Liu, 2012), di Amerika kurang lebih 20-30 perusahaan yang memfokuskan pada layanan analisis sentimen.

2.3 *Text Preprocessing*

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 pembentukan 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.
    a. Standar, yang mencakup imbuhan ‘di-‘, ‘ke-‘, dan ‘se-‘.
    b. 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.

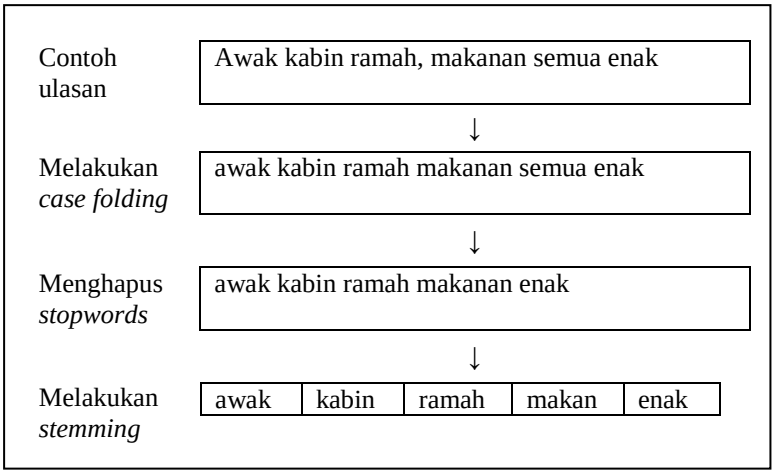
Aturan pembentukan kata dalam Bahasa Indonesia berkaitan dengan praproses teks karena hasil akhir praproses teks diharapkan mendapatkan kata dasar yang sesuai dengan Kamus Besar Bahasa Indonesia. Tahapan dalam praproses teks adalah sebagai berikut.

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, Sistla, 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*.

Penjelasan mengenai hasil dari setiap tahap praproses teks akan dijabarkan pada simulasi praproses teks pada sebuah data ulasan. Ulasan yang akan digunakan sebagai contoh adalah ulasan "Awak kabin ramah, makanan semua enak"



Gambar 2.2 Simulasi Praproses Teks

Untuk ulasan berikutnya, seperti ulasan "Kualitas makanan menurun" akan dilakukan praproses teks dengan langkah-langkah yang sama sehingga menghasilkan hasil praproses terakhir sebagai berikut.

kualitas	makan	turun

Gambar 2.3 Contoh Hasil Praproses Teks

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

Tabel 2.1 Contoh Struktur Data setelah Praproses Teks

Ulasan	**Variabel Prediktor**						
ke	awak	kabin	ramah	makan	enak	kualitas	turun
1	1	1	1	1	1	0	0
2	0	0	0	1	0	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 ulasan 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 ulasan 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 mengembalikan 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. Langkah-langkah *Nazief and Adriani's Stemmer* hingga mendapatkan kata dasar yang diinginkan 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:
      c. Terjadi kombinasi awalan dan akhiran terlarang seperti pada Tabel 2.2
      d. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
      e. 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.
      f. Standar (‘di-‘, ‘ke-‘, ‘se-‘) merupakan awalan yang dapat dihilangkan langsing dari kata
      g. 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 ‘teram-puni’.

2.6 *Naïve Bayes Classifier*

Metode *Naïve Bayes Classifier* (NBC) merupakan metode yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Feldman & Sanger, 2007). Metode *Naïve Bayes Classifier* merupakan salah satu metode yang dapat mengklasifikasikan teks. Kelebihan NBC adalah algoritmanya sederhana tetapi memiliki akurasi yang tinggi. Terdapat dua tahap dalam mengklasifikasikan ulasan. Tahap pertama adalah pelatihan (*training*) terhadap ulasan yang telah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi ulasan yang belum diketahui kategorinya (*testing*) (Falahah & Nur, 2015).

Pada tahap klasifikasi ditentukan nilai kategori dari suatu dokumen berdasarkan *term* yang muncul dalam dokumen yang diklasifikasi. Lebih jelasnya, jika diasumsikan dimiliki dokumen $D = \{d_i | \, i = 1, 2, \dots, |D|\} = \{d_1, d_2, \dots, d_{|D|}\}$ dan kategori $V = \{v_j | \, j = 1, 2, \dots, |V|\} = \{v_1, v_2, \dots, v_{|V|}\}$. Klasifikasi NBC dilakukan dengan cara mencari probabilitas $P(V = v_j | D = d_i)$, yaitu probabilitas kategori v_j jika diketahui dokumen d_i. Dokumen d_i dipandang sebagai *tuple* dari kata-kata dalam dokumen, yaitu $< a_1, a_2, \dots, a_i >$, dimana a_1 adalah kata pertama, a_2 adalah kata kedua dan seterusnya. Selanjutnya klasifikasi dokumen adalah mencari nilai maksimum dari:

$$V_{MAP} = \arg\max_{vj=V} P(v_j | a_1, a_2, \dots, a_i) \tag{2.1}$$

Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan (V_{MAP}).

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

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{2.2}$$

keterangan:

A : Hipotesis data B merupakan suatu *class* spesifik

B : Data dengan *class* yang belum diketahui

$P(A|B)$: Probabilitas hipotesis A berdasar kondisi B

$P(A)$: Probabilitas hipotesis A
$P(B|A)$: Probabilitas B berdasar kondisi pada hipotesis A
$P(B)$: Probabilitas dari B

Dengan menerapkan teorema Bayes, persamaan (2.1) dapat ditulis:

$$V_{MAP} = \arg\max_{vj=V} \frac{P(a_1, a_2, \dots, a_i|v_j)P(v_j)}{P(a_1, a_2, \dots, a_i)} \quad (2.3)$$

Karena nilai $P(a_1, a_2, \dots, a_i)$ untuk semua v_j besarnya sama maka nilainya dapat diabaikan, sehingga persamaan (2.3) menjadi.

$$V_{MAP} = \arg\max_{vj=V} P(a_1, a_2, \dots, a_i|v_j)P(v_j) \quad (2.4)$$

Dengan mengasumsikan bahwa setiap kata dalam $< a_1, a_2, \dots, a_i >$ adalah independen, maka $P(a_1, a_2, \dots, a_i|v_j)P(v_j)$ dalam persamaan (2.4) dapat ditulis sebagai.

$$P(a_1, a_2, \dots, a_i|v_j) = \prod_i P(a_i|v_j) \quad (2.5)$$

Sehingga persamaan (2.4) dapat ditulis:

$$V_{MAP} = \arg\max_{vj=V} P(v_j) \prod_i P(a_i|v_j) \quad (2.6)$$

Nilai $P(v_j)$ dihitung pada saat *training*, didapat dengan rumus sebagai berikut.

$$P(v_j) = \frac{|doc_{ij}|}{|training|} \quad (2.7)$$

dimana $|doc_{ij}|$ merupakan jumlah ulasan ke-i yang memiliki kategori j dalam *training*. Sedangkan $|training|$ merupakan jumlah ulasan yang digunakan untuk data *training*. Untuk setiap probabilitas kata a_i untuk setiap kategori $P(a_i|v_j)$, dihitung pada saat *training*.

$$P(a_i|v_j) = \frac{n_i + 1}{n + |kosakata|} \quad (2.8)$$

dimana n_i adalah jumlah kemunculan kata a_i dalam ulasan yang berkategori v_j, sedangkan n adalah jumlah seluruh kata dalam ulasan dengan kategori v_j dan $|kosakata|$ adalah banyaknya kata dalam data *training*. Diilustrasikan perhitungan dalam mengklasifikasi data ulasan menggunakan metode *Naïve Bayes Classifier*. Hal pertama yang dilakukan adalah menghitung

probabilitas setiap kelas sentimen menggunakan persamaan (2.7). Setelah probabilitas setiap kelas sentimen diketahui, data ulasan yang sudah di *preprocessing* kemudian dilakukan perhitungan probabilitas kemunculan setiap kata pada masing-masing kategori menggunakan persamaan (2.8). Setelah didapatkan probabilitas kemunculan setiap kata pada setiap kategori ulasan, selanjutnya adalah mencari probabilitas tertinggi dari ulasan yang diujikan menggunakan persamaan (2.6).

2.7 *Term Frequency Inverse Document Frequency* (TF-IDF)

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 ulasan dimana suatu kata tersebut muncul (idf). Metode TF-IDF dilakukan dengan menghitung bobot dengan cara integrasi antara *term frequency* (tf) dan i*nverse document frequency* (idf). Berikut merupakan rumus untuk menemukan pembobot dengan TF-IDF.

$$w_{ij} = tf_{ij} \times idf$$
$$idf = log\left(\frac{N}{df_{ji}}\right) \quad (2.9)$$

dimana:

w_{ij} : bobot dari kata i pada artikel ke j
N : jumlah seluruh ulasan
tf_{ij} : jumlah kemunculan kata i pada ulasan j
df_{ji} : jumlah ulasan j yang mengandung kata i.

TF-IDF dilakukan agar data dapat dianalisis dengan menggunakan *Support Vector Machine*.

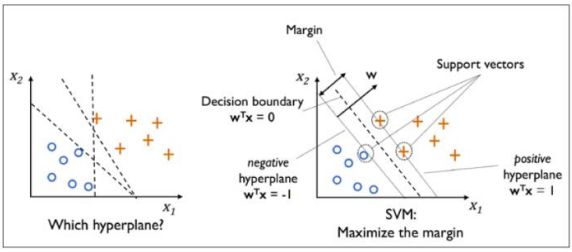
2.8 *Support Vector Machine*

Support Vector Machine (SVM) adalah metode yang mempelajari area yang memisahkan antar kategori dalam sebuah observasi. Dalam terminologi 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 teori 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 *Support Vector Machine* pada *Linearly Separable Data*

SVM pada *linearly separable data* adalah penerapan metode SVM pada data yang dapat dipisahkan secara linier. Misalkan $x_i = \{x_i, x_i + 1, \dots, x_n\}$ adalah dataset dan $y_i = \{1, -1\}$ adalah label kategori untuk dataset. Apabila x_i merupakan anggota dari kelas 1, maka x_i mempunyai label y_i= 1, begitu pula sebaliknya Penggambaran *linearly separable data* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Alternatif Bidang Pemisah (kiri) dan Bidang Pemisah Terbaik dengan Margin (m) Terbesar (kanan)

Pada Gambar 2.4, 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.

$$x_i' w + b \qquad (2.10)$$

Dimana *w* adalah vektor bobot yang tegak lurus terhadap *hyperplane* dan *b* merupakan posisi bidang relatif terhadap pusat koordinat. Nilai *margin* (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) yaitu.

$$\frac{2}{\|w\|} \tag{2.11}$$

Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan (2.10). 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 (2.10) 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 sebagai berikut.

$$y_i(x_i'w + b) - 1 \geq 0 \tag{2.12}$$

maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi *constraint*, yaitu.

$$min\left(\frac{1}{2}\|w\|^2\right) \tag{2.13}$$

Lagrangian untuk masalah mendasar pada kasus ini adalah sebagai berikut.

$$\min_{w,b} L_d(w,b) = \min_{w,b}\left[\frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i\left[y_i(x_i'w + b) - 1\right]\right] \tag{2.14}$$

Meminimalkan *L* terhadap *w* dan *b* dapat diberikan sebagai berikut.

$$\begin{aligned} &\frac{\partial L_d(w,b)}{\partial w} = 0, \\ &w = \sum_{i=1}^{n}\alpha_i y_i x_i, \end{aligned} \tag{2.15}$$

$$\frac{\partial L_d(w,b)}{\partial b} = 0,$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.16}$$

Lagrangian untuk permasalahan ganda dengan mensubstitusi persamaan (2.15) dan (2.16) adalah sebagai berikut.

$$\max_{\alpha} L_d(\alpha) = \max_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \alpha_{i_1} \alpha_{i_2} y_{i_1} y_{i_2} \left(x'_{i_1} x_{i_2} \right) \right] \tag{2.17}$$

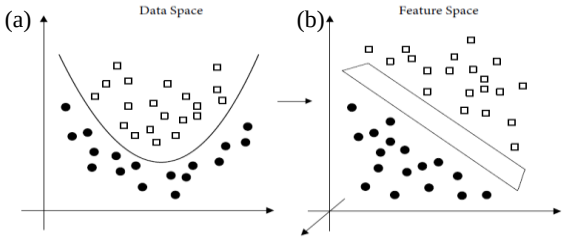
Dimana n merupakan jumlah seluruh kata dalam ulasan. Persamaan L_d digunakan untuk mencari nilai-nilai α_i (*support vector*) dengan membuat L_d optimum. L_d optimum didapat dengan cara mencari turunan parsial L_d terhadap α_i. Setelah mendapatkan nilai α_i, langkah selanjutnya adalah mencari nilai ω dan b dengan persamaan sebagai berikut.

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \text{ dan } b = 1 - w'x \tag{2.18}$$

Dengan α_i merupakan Lagrange multiplier yang berkorespondensi dengan x_i. Diilustrasikan perhitungan dalam mengklasifikasi data ulasan menggunakan metode *Support Vector Machine*. Data ulasan yang sudah di *preprocessing* kemudian diberi pembobotan dengan *term frequency-inverse document frequency*. Setelah didapatkan nilai pembobot masing-masing kata pada setiap kategori ulasan, kemudian mensubtitusi nilai bobot sebagai variabel x dan nilai sentimen sebagai variabel y pada persamaan (2.17) sehingga akan didapat persamaan L_d. Persamaan tersebut digunakan untuk mendapatkan α_i. Kemudian dilanjutkan pada persamaan (2.18) untuk mendapatkan nilai w dan nilai b. Nilai-nilai tersebut selanjutnya akan digunakan untuk membangun persamaan *hyperplane*.

2.8.2 *Support Vector Machine* pada *Non Linearly Separable Data* menggunakan Kernel

Pada kehidupan nyata kerap ditemukan data yang bersifat nonlinier dibandingkan linier. Klasifikasi data yang tidak dapat dipisahkan secara linier memerlukan modifikasi pada formula SVM agar dapat menemukan solusinya. Data nonlinier perlu dipetakan dengan menggunakan fungsi pemetaan $\psi: \mathcal{R}^p \rightarrow H$ ke dalam ruang yang berdimensi tinggi H dimana aturan klasifikasi bersifat linier yang ilustrasinya dapat dilihat pada Gambar 2.5.



Gambar 2.5 Pemetaan Ruang Dua Dimensi Data Menjadi Tiga Dimensi

Gambar 2.5 (a) menunjukkan bahwa data tidak dapat dipisahkan secara linier apabila menggunakan pemisah dua dimensi sehingga dipetakan dengan fungsi ψ menjadi ruang tiga dimensi (b) dan dapat dipisahkan secara linier. Fungsi transformasi pada SVM menggunakan *kernel trick* yang digunakan untuk menghitung *scalar product* melalui fungsi *kernel* (Schölkopf & Smola, 2002). Proyeksi fungsi $\psi: \mathcal{R}^p \rightarrow H$ memastikan bahwa *inner product* $\psi(x_{i_1})'\psi(x_{i_2})$ ditunjukkan oleh fungsi *kernel*

$$K(x_{i_1}, x_{i_2}) = \psi(x_{i_1})'\psi(x_{i_2}) \tag{2.19}$$

Pencarian *hyperplane* yang optimal akan memperhatikan data-data yang tidak berada pada kelasnya (*misclassification error*) yang dilambangkan dengan variabel *slack* ξ. Penambahan variabel ini menunjukkan pelanggaran terhadap ketelitian pemisah yang memungkinkan suatu titik berada di dalam *error margin* $0 \leq \xi \leq 1$ atau disebut misklasifikasi, $\xi > 1$ sehingga klasifikasi x_i adalah sebagai berikut (Härdle, dkk., 2014).

$$x_i'w + b \geq 1 - \xi_i, \text{ untuk } y_i = 1, \tag{2.20}$$

$$x_i' w + b \geq -(1 - \xi_i), \text{untuk } y_i = -1, \quad (2.21)$$

Sehingga dari persamaan (2.20) dan (2.21) dihasilkan persamaan berikut.

$$y_i = (x_i' w + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (2.22)$$

Penalti untuk kesalahan klasifikasi terkait dengan jarak titik kesalahan klasifikasi x_i dari *hyperplane* yang membatasi kelasnya. Jika $\xi_i \geq 0$, maka kesalahan dalam memisahkan dua set terjadi. Fungsi obyektif sesuai dengan maksimalkan *penalized margin* kemudian diformulasikan sebagai berikut.

$$\min_{w,\xi} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \right) \quad (2.23)$$

Dengan nilai $\xi_i \geq 0$ dan parameter pinalti $C > 0$ dimana C adalah parameter yang menentukan besar kecilnya bobot akibat *misclassification* yang nilainya ditentukan. Persamaan *Lagrange Multiplier* pada data yang tidak dapat dipisahkan secara linier adalah sebagai berikut.

$$\max_{\alpha} L_d(\alpha) = \max_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \alpha_{i_1} \alpha_{i_2} y_{i_1} y_{i_2} K(x_{i_1} x_{i_2}) \right] \quad (2.24)$$

Constraint yang digunakan untuk memaksimalkan α_i pada persamaan (2.24) adalah sebagai berikut.

$$\sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1,2, \dots, n \quad (2.25)$$

Salah satu *kernel* yang banyak digunakan adalah *linear* dan *Radial Basis Function* (RBF), yang dirumuskan sebagai berikut.

Tabel 2.3 Fungsi Kernel pada SVM

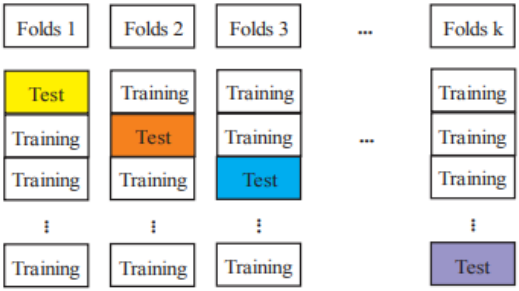
Fungsi Kernel	Rumus $K(x_{i_1}, x_{i_2})$	Parameter
Linear	${x_{i_1}}^T . x_{i_2}$	C
RBF	$\exp\left(-\gamma \left\| x_{i_1} - x_{i_2} \right\|^2\right)$	γ dan C

Berdasarkan adanya tambahan fungsi *kernel* maka fungsi hasil *training* yang dihasilkan adalah sebagai berikut.

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{\alpha}_i y_i \, K\left(x_{i_1}, x_{i_2}\right) + \hat{b} \qquad (2.26)$$

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 2.6 Ilustrasi Pembagian Data

Berdasarkan Gambar 2.6, diilustrasikan pembagian data-*training* dan data *testing* menggunakan 5-*fold cross validation*. 5-*fold cross validation* merupakan salah satu metode *cross validation* yang direkomendasikan untuk pembagian data *training* dan *testing* terbaik karena cenderung memberikan estimasi akurasi yang kurang bias dibandingkan dengan *cross validation* biasa, *leave-one-out cross validation*, dan *bootstrap*. Pada 5-*fold cross validation*, data dibagi menjadi 5 *fold* yang berukuran kira-kira sama, sehingga dimiliki 5 *subset* data yang digunakan untuk mengevaluasi data yang akan digunakan menjadi data *testing*. Pada

masing-masing dari 5 *subset* data tersebut, *cross validation* akan menggunakan 4 *fold* untuk melatih data dan 1 *fold* untuk menguji data.

2.10 *Synthetic Minority Oversampling Technique* (SMOTE)

Synthetic Minority Oversampling Technique atau SMOTE adalah salah satu teknik *over-sampling* yang sering digunakan untuk menangani masalah data *imbalanced* (data yang tidak seimbang) dengan cara membuat data sintetis pada kelas data minor sehingga data menjadi hampir seimbang (Bunkhumpornpat, Sinapiromsaran, & Lursinsap, 2009). Penggunaan teknik SMOTE diharapkan berimbas pada akurasi klasifikasi yang lebih baik. Cara menentukan data sintetis dirumuskan dalam persamaan berikut.

$$x_{syn} = x_i + \delta(x_{knn} - x_i) \qquad (2.27)$$

keterangan:

x_{syn} : nilai bobot sintetik

x_i : nilai bobot data ke-i di kelas minoritas

x_{knn} : nilai bobot dari data di kelas minoritas yang memiliki jarak terdekat dengan x_i

δ : bilangan acak antara 0 dan 1

Prosedur pembangkitan data sintetis untuk:

1. Data Numerik
   a. Hitung perbedaan antar nilai bobot dari data di kelas minoritas yang memiliki jarak terdekat dengan x_i.
   b. Kalikan perbedaan dengan angka yang diacak diantara 0 dan 1.
   c. Tambahkan perbedaan tersebut ke dalam nilai bobot data ke-i di kelas minoritas sehingga diperoleh nilai bobot sintetiknya.
2. Data Kategorik
   a. Pilih mayoritas antara nilai bobot dari data di kelas minoritas yang memiliki jarak terdekat dengan x_i untuk nilai nominal. Jika terjadi nilai sama maka pilih secara acak.
   b. Jadikan nilai tersebut sebagai data contoh kelas buatan baru.

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

Tabel 2.4 *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 menghitung akurasi, *specificity* dan *sensitivity*.

$$akurasi = \frac{TN + TP}{TN + TP + FN + FP} \tag{2.28}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{2.29}$$

$$specificity = \frac{TN}{TN + FP} \tag{2.30}$$

Sedangkan untuk data *imbalanced*, pengukuran ketepatan klasifikasi yang digunakan adalah *Area Under Curve* (AUC). AUC merupakan indikator performa kurva ROC (*Receiver Operating*

Characteristic) yang dapat meringkas kinerja sebuah *classifier* menjadi satu nilai (Bekkar, Djemaa, & Alitouch, 2013).

$$AUC = \frac{1}{2}(sensitivity + specificity) \quad (2.31)$$

2.12 *Airline Company*

PT X merupakan maskapai penerbangan nasional Indonesia yang beroperasi sejak tahun 1949 dan berstatus sebagai BUMN (Badan Usaha Milik Negara). Pangsa pasar yang dituju oleh PT X adalah kalangan menengah hingga atas, sesuai dengan konsep yang diterapkan PT X yaitu layanan penuh atau *full service*. Sebagai bukti terhadap kualitas yang ditawarkan, PT X sudah tersertifikasi sebagai maskapai bintang lima oleh *Skytrax*, lembaga pemeringkat penerbangan independen yang berbasis di London. Tak cukup sampai disitu, PT X juga bergabung dengan aliansi maskapai penerbangan *SkyTeam*, salah satu aliansi maskapai penerbangan terkemuka di dunia. Dilansir dari situs resmi PT X (2020), maskapai penerbangan plat merah ini juga meraih beberapa pencapaian diantaranya "*The World's Best Cabin Crew*" selama empat tahun berturut-turut, dari tahun 2014 hingga 2017; "*The World's Most Loved Airline 2016*" dan "*The World's Best Economy Class 2013*" dari *Skytrax*.

Akan tetapi dalam perjalanannya, kisah PT X tak selalu mulus. Tahun 2019 merupakan tahun yang cukup berat bagi PT X, dimana perusahaan ini sempat dilanda berbagai isu yang berkaitan dengan manajemen perusahaan sehingga cukup disorot oleh masyarakat dan media massa. Sebagai maskapai penerbangan kebanggaan Indonesia, isu yang melanda PT X tentu tidak boleh mempengaruhi kualitas yang diberikan kepada konsumennya. Untuk terus memantau tanggapan penumpang mengenai kualitas pelayanan yang diberikan baik sebelum penerbangan (*pre-flight*), selama penerbangan (*in-flight*), dan hingga setelah tiba (*post-flight*), PT X secara rutin melakukan survey layanan kepada penumpang. Survei dilakukan secara langsung kepada penumpang, dimana penumpang mengisi *suggestion form* yang diberikan saat

penerbangan. Melalui *suggestion form* tersebut penumpang dapat secara bebas memberikan opini/tanggapan/ulasan terhadap pelayanan penerbangan yang diberikan oleh PT X.

(Halaman ini sengaja dikosongkan)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari PT X. PT X melakukan survei kepada penumpang, dimana penumpang mengisi *suggestion form* yang diberikan saat penerbangan. Melalui *suggestion form* tersebut penumpang memberikan ulasan terhadap pelayanan penerbangan yang diberikan. Data merupakan ulasan (komentar) penerbangan domestik selama periode Juli 2019 sampai Desember 2019. Terdapat 1492 ulasan yang digunakan pada penelitian ini.

3.3 Struktur Data

Struktur data yang digunakan dalam penelitian ini setelah dilakukan praproses pada data teks ulasan terdiri dari variabel prediktor yaitu kata dasar setiap ulasan dan variabel respon yaitu klasifikasi sentimen ulasan (positif dan negatif). Berikut merupakan contoh struktur data penelitian.

Tabel 3.1 Struktur Data Penelitian

No	Ulasan	Sentimen
1	pramugari ramah	Positif
2	senang terbang bersama garuda	Positif
3	terbang nyaman	Positif
4	film kurang update	Negatif
5	makan kurang enak	Negatif
6	delay lama tunggu	Negatif
…	…	…

3.4 Langkah Analisis

Langkah analisis yang digunakan dalam penelitian ini antara lain adalah sebagai berikut.

1. Menyiapkan data ulasan, daftar *stopwords*, dan kata dasar.
2. Melakukan praproses data untuk menghindari data yang belum siap olah, seperti: data yang kurang sempurna, data yang mengandung gangguan (*noise*), dan data yang tidak

konsisten (Hemalatha, 2012). Tahapan praproses adalah sebagai berikut.

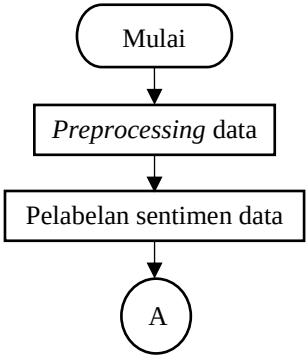
a. Menghapus data ulasan yang mengandung dua sentimen sekaligus (positif dan negatif).
b. Menghapus simbol dan tanda baca, seperti: `~!@#$%^&*()_+-={}[]\|:;”’<>,.
c. Melakukan *case folding*, yaitu merubah semua teks menjadi huruf kecil (non kapital).
d. Menghapus kata pada ulasan yang terdapat di dalam daftar *stopwords*.
e. Melakukan *tokenizing* untuk memecah ulasan menjadi kata per kata.
f. Melakukan *stemming* untuk menghilangkan imbuhan dari kata dan mendapatkan kata dasar.
g. Mengubah data ulasan kedalam bentuk frekuensi kemunculan kata seperti pada contoh struktur data pada Tabel 2.1

3. Melakukan pelabelan sentimen data secara manual.
4. Melakukan eksplorasi data.
5. Membagi data menjadi data *training* dan data *testing* menggunakan *K-Fold Cross Validation.*
6. Melakukan teknik SMOTE jika jumlah data antar kategori sentimen tidak seimbang (*imbalanced*).
7. Klasifikasi data menggunakan *Naïve Bayes Classifier* untuk data ulasan.
    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 (2.2).

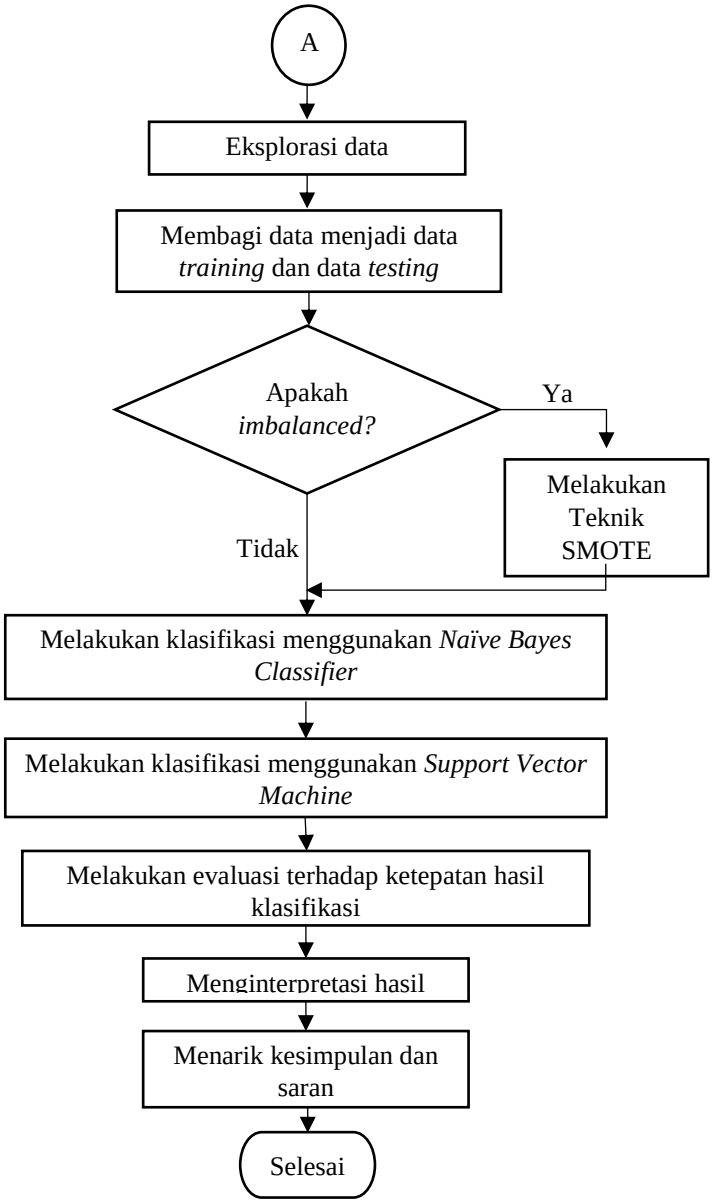
e. Mencari nilai V_{MAP} paling maksimum dan memasukkan ulasan tersebut pada kategori dengan V_{MAP} maksimum.

8. Klasifikasi data menggunakan *Support Vector Machine* untuk data ulasan.
   a. Mengubah teks menjadi vektor dan pembobotan kata dengan TF-IDF menggunakan persamaan (2.5).
   b. Melakukan tuning parameter dengan memasukkan nilai $10^{-2} \leq \gamma \leq 10^{2}$ dan $10^{-2} \leq C \leq 10^{2}$ dengan penambahan nilai dikalikan 10 pada masing-masing parameter.
   c. Menentukan nilai-nilai γ dan C yang akan digunakan berdasarkan langkah 8b.
   d. Membangun model SVM menggunakan fungsi kernel *linear* dan *Radial Basis Function*.
9. Evaluasi hasil klasifikasi
   Menghitung ketepatan klasifikasi dan membandingkan performansi metode NBC dan SVM berdasarkan tingkat akurasi dengan persamaan (2.24) jika data *balance* atau berdasarkan nilai AUC dengan persamaan (2.27) jika data *imbalance*.
10. Interpretasi, menarik kesimpulan dan saran.

Berdasarkan langkah analisis yang telah dijelaskan dapat digambarkan diagram alir penelitian ini pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian



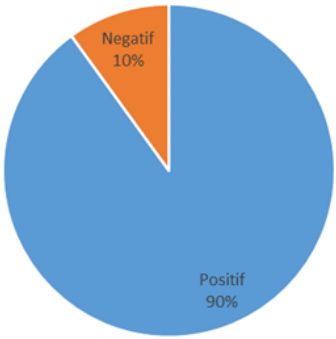
Gambar 3.2 Diagram Alir Penelitian (lanjutan)

BAB IV
ANALISIS DAN PEMBAHASAN

Bab ini membahas mengenai hasil analisis dari klasifikasi sentimen menggunakan metode *Naïve Bayes Classifier* dan *Support Vector Machine*. Pada pembahasan di bab ini menjelaskan proses pengolahan data teks dari tahap praproses sampai hasil klasifikasi sentimen ulasan.

4.1 Praproses dan Karakteristik Data

Karakteristik atau gambaran umum data sebagai informasi awal dari sebuah data dapat didapatkan dengan melakukan ekplorasi. Eksplorasi dilakukan sebelum menentukan atau menerapkan metode analisis yang tepat. Terdapat 1492 ulasan penumpang pesawat yang digunakan pada penelitian ini. Ulasan tersebut terpisah menjadi 2 kategori sentimen, yaitu positif dan negatif. Berikut disajikan *pie chart* guna mengetahui perbandingan frekuensi antar kategori sentimen.



Gambar 4.1 *Pie Chart* Perbandingan Kategori Sentimen

Berdasarkan Gambar 4.1, dapat dilihat bahwa dari 1492 ulasan penumpang, 1343 ulasan berkategori positif dan 149 ulasan sisanya berkategori negatif. Dari jumlah tersebut, 90% ulasan didominasi oleh ulasan positif. Hasil eksplorasi menggunakan *pie chart* dapat memberikan informasi bahwa terdapat ketimpangan (*imbalanced*) jumlah data antar kategori sentimen. Sebelum melakukan eksplorasi lebih dalam, terlebih dahulu dilakukan

praproses pada data. Data ulasan penumpang pesawat yang telah terkumpul dilakukan praproses teks meliputi *case folding*, *stopwords removal*, *stemming*, dan *tokenizing*. Praproses teks dilakukan dengan langkah langkah seperti pada Gambar 3.1. Berikut merupakan struktur data ulasan penumpang pesawat sebelum dilakukan praproses teks.

Tabel 4.1 Struktur Data Sebelum Praproses

Kategori Sentimen	**Ulasan**
Positif	Sangat menikmati pengalaman dengan layanan yang baik terima kasih untuk semua crew atas pelayanannya
Positif	Merasa nyaman menyenangkan pelayanan ramah sopan dan penuh perhatian
Negatif	Business lounge di CGK sangat tidak nyaman karena makanan tidak enak fasilitas dibandara CGK loungenya tidak nyaman dan disejajarkan dengan skyteam
Positif	pramugrai dapat berkoordinasi dengan baik, pelayanan ramah sopan baik cepat tanggap dan jelas dalam memberikan instruksi
Negatif	hiburan didalam pesawat khususnya film sangat tidak up to date
Positif	Awak kabin sangat ramah sangat membantu dan saya merasa senang terbang bersama garuda indonesia
…	…
Positif	saya sangat puas dengan pelayanan pramugari ramah baik ,berharap setiap penerbangan ke china ada yang bisa berbahasa mandarin

Data ulasan yang belum dilakukan praproses masih tersusun dalam satu kolom seperti pada Tabel 4.1. Data pada tabel menunjukkan ulasan yang masih membuat huruf besar, kata berimbuhan, kata-kata yang bukan merupakan kata penting dalam ulasan (*stopwords*), dan simbol-simbol lainnya yang tidak menggambarkan isi ulasan seperti tanda baca, sehingga perlu

dilakukan praproses guna mendapatkan ulasan yang tidak memuat hal-hal tersebut. Praproses teks juga bertujuan untuk meningkatkan performa klasifikasi dan mengurangi misklasifikasi pada data. Berikut merupakan struktur data ulasan penumpang pesawat yang telah dilakukan praproses.

Tabel 4.2 Struktur Data Setelah Praproses

Ulasan	baik	…	hibur	…	layan	…	nyaman
nikmat layan…	0	…	0	…	1	…	0
…terima kasih…	0	…	0	…	0	…	0
nyaman senang…	0	…	0	…	0	…	1
layan ramah…	0	…	0	…	1	…	0
sopan hati…	0	…	0	…	0	…	0
fasilitas bandara…	0	…	0	…	0	…	0
…tidak nyaman…	0	…	0	…	0	…	1
hibur pesawat…	0	…	1	…	0	…	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
sangat puas…	0	…	0	…	0	…	0

Data pada Tabel 4.2 yang sudah berbentuk *document term matrix* dapat dilakukan perhitungan jumlah kata yang selanjutnya akan menjadi jumlah variabel dari data ulasan. Terdapat 6374 kata dari data ulasan penumpang pesawat yang digunakan pada penelitian ini. Setelah terbentuk struktur data yang diinginkan, dilakukan perhitungan frekuensi kemunculan kata pada masing-masing kategori sentimen. Berikut merupakan frekuensi kemunculan kata tertinggi dari masing-masing kategori sentimen.

Tabel 4.3 Frekuensi Kemunculan Kata Tertinggi Setiap Kategori Sentimen

Positif		Negatif	
Kata	**Jumlah**	**Kata**	**Jumlah**
pelayanan	827	update	44
ramah	471	film	38

Tabel 4.3 Frekuensi Kemunculan Kata Tertinggi Setiap Kategori Sentimen (Lanjutan)

Positif		Negatif	
Kata	**Jumlah**	**Kata**	**Jumlah**
baik	425	makanan	37
puas	327	hiburan	13
penerbangan	314	menu	12
nyaman	249	kualitas	11

Daftar kata dengan frekuensi kemunculan tertinggi pada data ulasan penumpang pesawat pada Tabel 4.3 menunjukkan bahwa kata-kata tersebut merupakan kata-kata yang sering dituliskan dalam masing-masing kategori ulasan dan mempunyai pengaruh signifikan dalam pembangunan model klasifikasi. Selain menggunakan tabel diatas, untuk mengetahui kata-kata yang peling sering muncul pada data dapat menggunakan visualisasi data menggunakan *word cloud*. Ukuran *font* pada *word cloud* menunjukkan frekuensi kemunculan kata, semakin besar ukuran *font* berarti semakin besar frekuensi kemunculan kata tersebut.

Visualisasi menggunakan *word cloud* akan dilakukan dengan membandingkan antara data ulasan yang berkategori sentimen positif dan data ulasan yang berkategori sentimen negatif. Perbandingan tersebut dilakukan dengan tujuan untuk mengetahui penyebab mayoritas penumpang menilai pelayanan maskapai berdasarkan kategori sentimen. Berikut merupakan *word cloud* data masing-masing kategori sentimen.

(a) (b)

Gambar 4.2 *Word Cloud* Sentimen Positif (a) dan Negatif (b)

Kata-kata yang memiliki ukuran *font* besar pada kategori sentimen positif Gambar 4.2 menunjukkan bahwa penumpang yang memberikan ulasan positif paling banyak membahas mengenai pelayanan maskapai penerbangan PT. X. Respon positif mengenai pelayanan penerbangan dapat dijadikan saran bagi PT. X agar terus mempertahankan bahkan meningkatkan pelayanannya. Respon positif tersebut diikuti kata 'ramah', 'baik', 'puas', dan 'nyaman' yang berarti penumpang merasakan kepuasan dalam pelayanan yang ramah, baik, dan nyaman.

Pada kategori sentimen negatif, penumpang paling banyak menyebutkan kata 'update', diikuti kata 'film', 'makanan', 'hiburan', 'menu', dan 'kualitas'. Penyebutan kata-kata tersebut berarti penumpang yang memberikan ulasan negatif menyoroti film, makanan, atau hiburan yang kurang *update* hingga menyoroti kualitas makanan yang diberikan oleh PT. X selama penerbangan. Hal ini dapat dijadikan saran dan evaluasi bagi PT. X untuk senantiasa melakukan *update* pada sektor hiburan serta memperbaiki kualitas makanan yang diberikan kepada penumpang.

4.2 Klasifikasi Menggunakan *Naïve Bayes Classifier* (NBC)

Metode klasifikasi pertama yang digunakan adalah *Naïve Bayes Classifier* (NBC). Model dilatih terlebih dahulu menggunakan data *training* dari ulasan penumpang pesawat yang telah dilakukan praproses menggunakan *software* Python 3.7. Data *training* dan *testing* dibagi menggunakan metode 5-*fold cross validation*. Model yang telah dilatih menggunakan data *training* kemudian digunakan untuk mengklasifikasikan data *testing* ke dalam 2 kategori sentimen, yaitu positif dan negatif. Dalam mengklasifikasikan ulasan, metode *Naïve Bayes Classifier* menghasilkan probabilitas yang digunakan untuk menentukan apakah suatu ulasan masuk ke dalam kategori sentimen positif atau negatif. Probabilitas tersebut diperoleh menggunakan persamaan (2.3) dan persamaan (2.4). Berikut beberapa nilai probabilitas yang dihasilkan dari model.

Tabel 4.4 Probabilitas Klasifikasi NBC

Probabilitas Positif	Probabilitas Negatif	Keputusan
0,9999191	8,09E-05	Positif
0,0013797	0,9986203	Negatif
0,9999191	8,09E-05	Positif
1	3,21E-09	Positif
1	3,73E-09	Positif
0,9992907	0,0007093	Positif
…	...	...
0.99058025	0,0094175	Positif

Nilai probabilitas ulasan pada tabel 4.4 tersebut menunjukkan peluang suatu ulasan untuk masuk ke dalam setiap kategori sentimen sebesar nilai yang ada pada kolom probabilitas masing-masing kategori sentimen. Suatu ulasan akan dikategorikan ke dalam salah satu kategori sentimen yang memiliki probabilitas paling besar. Sehingga apabila probabilitas suatu ulasan masuk ke kategori sentimen positif lebih besar dari probabilitas masuk ke kategori sentimen negatif, maka ulasan tersebut akan dikategorikan positif dan sebaliknya.

4.2.1 Hasil Klasifikasi Metode *Naïve Bayes Classifier*

Hasil klasifikasi menggunakan metode *Naïve Bayes Classifier* perlu diukur ketepatannya. Pengukuran ketepatan klasifikasi dilakukan dengan membentuk *confusion matrix* berdasarkan hasil prediksi. Berikut merupakan contoh *confusion matrix* hasil klasifikasi pada data *training*.

Tabel 4.5 *Confusion Matrix* Data *Training* Metode NBC

Kelas Aktual	Kelas Prediksi	
	Negatif	Positif
Negatif	103	17
Positif	2	1072

Metode *Naïve Bayes Classifier* dapat memprediksi data *training* secara tepat sebanyak 1.175 sentimen positif maupun

negatif. Hanya terdapat 19 data yang belum bisa diprediksi kategori sentimennya secara tepat. Berdasarkan *confusion matrix* yang terlah terbentuk pada Tabel 4.5, selanjutnya dapat dihitung nilai ketepatan klasifikasi metode *Naïve Bayes Classifier* dengan hasil sebagai berikut.

Tabel 4.6 Nilai Ketepatan Klasifikasi Metode NBC

Data	Training	Testing
Akurasi	0,984	0,973
AUC	0,928	0,893

Berdasarkan Tabel 4.6 dapat diketahui metode *Naïve Bayes Classifier* memiliki nilai akurasi sebesar 0,984 pada data *training* dan 0,973 pada data *testing*. Pada nilai AUC, terdapat penurunan nilai ketepatan klasifikasi menjadi sebesar 0,928 pada data *training* dan 0,893 pada data *testing*. Nilai AUC yang jauh lebih rendah dibandingkan dengan nilai akurasi mengindikasikan terdapat ketimpangan (*imbalanced*) antar kategori sentimen pada data *training*. Hal tersebut sesuai dengan perbandingan data ulasan positif yang berbanding jauh dengan data ulasan negatif, yaitu sebanyak 1.343 ulasan positif dan 149 ulasan negatif. Untuk mengatasi *imbalanced* pada data tersebut, maka akan dilakukan teknik SMOTE (*Syntethic Minority Oversampling Technique*). SMOTE akan melakukan teknik *oversampling* dengan cara membuat data sintetis pada kelas data minor sehingga data menjadi hampir seimbang. Berikut merupakan *confusion matrix* hasil klasifikasi pada data *training* setelah dilakukan SMOTE.

Tabel 4.7 *Confusion Matrix* Data *Training* Metode NBC dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	1067	7
Positif	6	1068

Setelah dilakukan SMOTE, metode *Naïve Bayes Classifier* dapat memprediksi data *training* secara tepat sebanyak 2.129 sentimen positif maupun negatif. Hanya terdapat 13 data yang belum bisa diprediksi kategori sentimennya secara tepat. Data yang

belum terklasifikasi secara tepat dikarenakan jumlah kata yang terdapat dalam data tersebut terlalu sedikit. Dimisalkan ulasan "fasilitas baik", ulasan tersebut seharusnya dikategorikan sebagai ulasan positif akan tetapi dideteksi sebagai ulasan negatif. Hal ini dikarenakan kata "fasilitas" lebih sering disebutkan di sentimen negatif sehingga probabilitas kata "baik" tertutupi oleh probabilitas kata "fasilitas" dan menghasilkan probabilitas final yang lebih condong ke sentimen negatif. Berdasarkan *confusion matrix* yang telah terbentuk pada Tabel 4.7, selanjutnya dapat dihitung nilai ketepatan klasifikasi metode *Naïve Bayes Classifier*. Nilai ketepatan klasifikasi sebelum dan sesudah dilakukan SMOTE akan dibandingkan dengan hasil sebagai berikut.

Tabel 4.8 Perbandingan Nilai Ketepatan Klasifikasi Metode NBC

Data	Tanpa SMOTE		Dengan SMOTE	
	Training	**Testing**	**Training**	**Testing**
Akurasi	0,984	0,973	0,993	0,983
AUC	0,928	0,893	0,994	0,945

Tabel 4.8 menunjukkan perbandingan nilai ketepatan klasifikasi sebelum dan sesudah dilakukan SMOTE. Setelah dilakukan SMOTE, nilai akurasi klasifikasi meningkat menjadi 0,993 pada data *training* dan 0,983 pada data *testing*. Penanganan *imbalanced* pada data juga terbukti dalam meningkatkan nilai AUC yaitu menjadi sebesar 0,994 pada data *training* dan 0,945 pada data *testing*. Artinya, berdasarkan nilai AUC, 99,3% data *training* dapat tepat terklasifikasi menggunakan metode *Naïve Bayes Classifier*, begitupun dengan 94,5% data *testing* yang juga dapat terklasifikasi secara tepat. Secara keseluruhan, hasil klasifikasi terbaik menggunakan metode *Naïve Bayes Classifier* ditunjukkan oleh data yang telah dilakukan SMOTE, baik pada data *training* maupun data *testing*.

4.3 Klasifikasi Menggunakan *Support Vector Machine*

Metode klasifikasi kedua yang digunakan adalah *Support Vector Machine* (SVM). Pada penelitian ini digunakan 2 macam kernel pada algoritma *Support Vector Machine*, yaitu kernel *linear* dan kernel *radial basis function* (RBF). Data yang akan

diklasifikasikan menggunakan *Support Vector Machine* terlebih dahulu dilakukan pembobotan *term frequency-inverse document frequency* (TF-IDF) terlebih dahulu pada setiap kata.

4.3.1 Klasifikasi Menggunakan SVM Kernel *Linear*

Model dilatih terlebih dahulu menggunakan data *training*. Pelatihan model menggunakan data *training* dengan SVM kernel *linear* perlu mempertimbangkan parameter *C*. Pemilihan parameter *C* akan ditentukan melalui *tuning* parameter dengan mencoba nilai parameter dari 10^{-3} hingga 10^{3}. Model yang telah dilatih menggunakan data *training* kemudian digunakan untuk mengklasifikasikan data *testing* ke dalam 2 kategori sentimen, yaitu positif dan negatif. Berikut merupakan hasil ketepatan klasifikasi pada data menggunakan SVM kernel *linear*.

Tabel 4.9 Nilai Ketepatan Klasifikasi SVM Kernel *Linear*

C	Akurasi		AUC	
	Training	**Testing**	**Training**	**Testing**
0,001	0,899	0,903	0,5	0,5
0,01	0,899	0,903	0,5	0,5
0,1	0,924	0,926	0,621	0,621
1	0,995	0,977	0,986	0,941
10	0,998	0,977	0,992	0,941
100	0,998	0,973	0,992	0,939
1000	0,998	0,973	0,992	0.939

Berdasarkan hasil kinerja evaluasi, model terbaik dengan akurasi dan AUC terbaik baik pada data *training* maupun *testing* terdapat pada parameter *C* sebesar 10. Pada parameter tersebut, model memiliki akurasi sebesar 0,998 pada data *training* dan 0,997 pada data *testing*. Artinya, SVM kernel *linear* mampu mengklasifikasikan data 99,8 % akurat pada data *training* dan 99,7% akurat pada data *testing*. AUC dari data *training* adalah sebesar 0,992 dan pada data *testing* sebesar 0,941. Pada metode SVM kernel *linear* juga akan dilakukan penangan *imbalanced* pada data menggunakan teknik SMOTE. Dengan penanganan

imbalanced diharapkan dapat meningkatkan performa ketepatan klasifikasi. Hasil dari kinerja klasifikasi setelah dilakukan SMOTE dapat dilihat sebagai berikut.

Tabel 4.10 Nilai Ketepatan Klasifikasi SVM Kernel *Linear* dengan SMOTE

C	Akurasi		AUC	
	Training	Testing	Training	Testing
0,001	0,962	0,939	0,962	0,936
0,01	0,948	0,916	0,948	0,938
0,1	0,979	0,966	0,979	0,966
1	0,995	0,970	0,995	0,953
10	0,998	0,983	0,998	0,975
100	0,998	0,970	0,998	0,968
1000	0,998	0,966	0,998	0,966

Penanganan *imbalanced* pada data dengan menggunakan teknik SMOTE terbukti dapat meningkatkan kinerja klasifikasi SVM kernel *linear*. Akurasi dan AUC terbaik baik pada data *training* maupun *testing* setelah dilakukan SMOTE masih terdapat pada parameter *C* sebesar 10. Akurasi pada data *training* tetap pada 0,998 dan pada *testing* meningkat menjadi 0,983. AUC pada data *training* meningkat menjadi 0,998 dan pada data *testing* meningkat menjadi 0,975. Perbandingan kinerja klasifikasi sebelum dan sesudah dilakukan SMOTE dapat dilihat pada tabel berikut.

Tabel 4.11 Perbandingan Nilai Ketepatan Klasifikasi Metode SVM Kernel *Linear*

Data	Tanpa SMOTE		Dengan SMOTE	
	Training	Testing	Training	Testing
Akurasi	0,998	0,977	0,998	0,983
AUC	0,992	0,941	0,998	0.975

Berdasarkan Tabel 4.11 dapat diketahui bahwa penanganan *imbalanced* pada data dapat meningkatkan kinerja klasifikasi metode SVM kernel *linear*, baik pada akurasi maupun AUC. Akurasi pada data *testing* setelah dilakukan SMOTE sebesar 0,983. Artinya, metode SVM kernel *linear* dapat mengklasifikasikan data

testing 98,3% akurat, atau tingkat kesalahannya hanya sebesar 1,7%.

4.3.2 Klasifikasi Menggunakan SVM Kernel RBF

Pembahasan klasifikasi menggunakan SVM kernel *Radial Basis Function* (RBF) akan sama dengan SVM kernel *linear*. Apabila pelatihan model dengan SVM kernel *linear* perlu mempertimbangkan parameter *C*, maka pada SVM kernel RBF akan ditambah parameter gamma. Pemilihan parameter *C* dan gamma akan ditentukan melalui *tuning* parameter dengan rentang nilai parameter yang sama yaitu dari 10^{-2} hingga 10^{2}. Model yang telah dilatih menggunakan data *training* kemudian digunakan untuk mengklasifikasikan data *testing* ke dalam 2 kategori sentimen, yaitu positif dan negatif. Berikut merupakan hasil ketepatan klasifikasi pada data menggunakan SVM kernel RBF.

Tabel 4.12 Nilai Ketepatan Klasifikasi SVM Kernel RBF

C	Gamma	Akurasi		AUC	
		Training	Testing	Training	Testing
100	0,01	0,996	0,977	0,987	0,941
100	0,1	0,998	0,980	0,992	0,943
100	1	0,998	0,977	0,992	0,910
100	10	0,998	0,909	0,992	0,534
100	100	0,998	0,909	0,992	0.534

Berdasarkan hasil kinerja evaluasi, model terbaik dengan akurasi dan AUC terbaik baik pada data *training* maupun *testing* terdapat pada parameter *C* sebesar 100 dan gamma sebesar 0,1. Pada parameter tersebut, model memiliki akurasi sebesar 0,998 pada data *training* dan 0,98 pada data *testing*. Artinya, metode SVM kernel RBF mampu mengklasifikasikan data 99,8% akurat pada data *training* dan 98% akurat pada data *testing*. *AUC* dari data *training* adalah sebesar 0,992 dan pada data *testing* sebesar 0,943. Pada metode SVM kernel RBF juga akan dilakukan penangan *imbalanced* pada data menggunakan teknik SMOTE. Dengan penanganan *imbalanced* diharapkan dapat meningkatkan performa

ketepatan klasifikasi. Hasil dari kinerja klasifikasi setelah dilakukan SMOTE dapat dilihat sebagai berikut.

Tabel 4.13 Nilai Ketepatan Klasifikasi SVM Kernel RBF dengan SMOTE

C	Gamma	Akurasi		AUC	
		Training	Testing	Training	Testing
100	0,01	0,997	0,973	0,997	0,954
100	0,1	0,999	0,980	0,999	0,958
100	1	0,999	0,977	0,999	0,910
100	10	0,999	0,909	0,999	0,534
100	100	0,999	0,909	0,999	0.534

Penanganan *imbalanced* pada data dengan menggunakan teknik SMOTE terbukti dapat meningkatkan kinerja klasifikasi SVM kernel RBF. Akurasi dan AUC terbaik baik pada data *training* maupun *testing* setelah dilakukan SMOTE masih terdapat pada parameter *C* sebesar 100 dan gamma sebesar 0,1. Akurasi pada data *training* meningkat menjadi 0,999 dan pada *testing* tetap pada 0,980. AUC pada data *training* meningkat menjadi 0,999 dan pada data *testing* meningkat menjadi 0,958. Perbandingan kinerja klasifikasi sebelum dan sesudah dilakukan SMOTE dapat dilihat pada tabel berikut.

Tabel 4.14 Perbandingan Nilai Ketepatan Klasifikasi Metode SVM Kernel RBF

Data	Tanpa SMOTE		Dengan SMOTE	
	Training	Testing	Training	Testing
Akurasi	0,998	0,980	0,999	0,980
AUC	0,992	0,943	0,999	0.958

Berdasarkan Tabel 4.14 dapat diketahui bahwa penanganan *imbalanced* pada data dapat meningkatkan kinerja klasifikasi metode SVM kernel RBF, baik pada akurasi maupun AUC. Akurasi pada data *testing* setelah dilakukan SMOTE sebesar 0,98. Artinya, metode SVM kernel RBF dapat mengklasifikasikan data *testing* 98% akurat, atau tingkat kesalahannya hanya sebesar 2%.

4.3.3 Model *Support Vector Machine*

Pembahasan hasil klasifikasi menggunakan SVM kernel *Radial Basis Function* (RBF) dan SVM kernel *linear* menunjukkan bahwa SVM kernel *linear* mempunyai hasil ketepatan klasifikasi yang lebih baik. Hasil ketepatan klasifikasi terbaik pada metode SVM kernel *linear* menggunakan nilai *C* sebesar 10. Nilai *support vector* kategori positif pada x_{i_1} dan nilai *support vector* kategori negatif pada x_{i_2} kemudian digunakan untuk membangun fungsi kernel *linear*. Setelah fungsi kernel didapatkan, selanjutnya fungsi *hyperplane* dihitung dengan mensubstitusikan fungsi kernel *linear*. Sehingga didapat fungsi *hyperplane* pada setiap data media sebagai berikut.

$$f(x) = \sum_{i=1}^{585} (2{,}2378\alpha_i x + \cdots - 0{,}7144\alpha_i x) + 0{,}6636 \qquad (4.1)$$

Persamaan *hyperplane* yang telah didapat digunakan untuk mengklasifikasikan data. Pada persamaan *hyperplane* (4.1), α_i merupakan nilai koefisien dari *support vector* dan x merupakan nilai input yang akan diklasifikasi.

4.4 Perbandingan Hasil *Naïve Bayes Classifier* dan *Support Vector Machine*

Ketepatan klasifikasi menggunakan metode *Naïve Bayes Clssifier* dan *Support Vector Machine* sudah didapatkan, langkah selanjutnya adalah membandingkan hasil dari kedua metode tersebut. Perbandingan metode NBC dan SVM pada data *training* dapat dilihat pada tabel berikut.

Tabel 4.15 Perbandingan Nilai Ketepatan Klasifikasi Data *Training*

Data	Tanpa SMOTE		Dengan SMOTE	
	Akurasi	**AUC**	**Akurasi**	**AUC**
NBC	0,984	0,928	0,994	0,994
SVM *linear*	0,998	0,992	0,998	0,998
SVM RBF	0,998	0,992	0,999	0.999

Berdasarkan hasil yang ditunjukkan pada Tabel 4.15 secara keseluruhan data yang telah dilakukan SMOTE menunjukkan hasil yang lebih baik dalam mengklasifikasikan data *training*. Pada data *training*, SVM kernel RBF menunjukkan hasil terbaik dalam mengklasifikasikan data ulasan dengan nilai akurasi dan AUC sebesar 99,9%. Perbandingan nilai ketepatan klasifikasi pada data *testing* ditunjukkan sebagai berikut.

Tabel 4.16 Perbandingan Nilai Ketepatan Klasifikasi Data *Testing*

Data	Tanpa SMOTE		Dengan SMOTE	
	Akurasi	**AUC**	**Akurasi**	**AUC**
NBC	0,973	0,893	0,980	0,943
SVM *linear*	0,977	0,941	0,983	0,975
SVM RBF	0,980	0,943	0,980	0.958

Berdasarkan hasil yang ditunjukkan pada Tabel 4.16 secara keseluruhan data yang telah dilakukan SMOTE menunjukkan hasil yang lebih baik dalam mengklasifikasikan data *testing*. Pada data *testing*, akurasi terbaik ditunjukkan oleh metode SVM kernel *linear* dengan skor sebesar 98,3%. AUC data *testing* terbaik ditunjukkan oleh metode SVM kernel *linear* dengan skor sebesar 97,5%.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut :

1. Dari 1.492 ulasan yang digunakan, 90% atau 1.342 ulasan didominasi oleh ulasan dengan sentimen positif, sedangkan 10% atau 149 ulasan sisanya adalah ulasan dengan sentimen negatif. Pada ulasan positif, kata-kata yang mendominasi adalah ‘pelayanan’, ‘ramah’, ‘baik’, ‘puas’, ‘penerbangan’ dan ‘nyaman’, yang menunjukkan para penumpang puas dengan pelayanan yang ramah dan baik selama penerbangan yang nyaman. Pada ulasan negatif, kata-kata yang mendominasi adalah ‘update’, ‘film’, ‘makanan’, ‘hiburan’, ‘menu’, dan ‘kualitas’, yang menunjukkan para penumpang menginginkan *update* pada sektor hiburan (film) dan meningkatkan kualitas dari menu makanan yang disajikan selama penerbangan.
2. Klasifikasi menggunakan metode *Naïve Bayes Classifier* menunjukkan hasil yang lebih baik pada data yang telah dilakukan SMOTE dengan akurasi data *training* sebesar 99,3% dan data *testing* sebesar 98,3%. AUC pada data *training* sebesar 99,4% dan pada data *testing* sebesar 94,5%.
3. Klasifikasi menggunakan metode *Support Vector Machine* menunjukkan hasil yang lebih baik pada data yang telah dilakukan SMOTE, baik pada kernel *linear* maupun RBF. Pada data *training*, kernel RBF menunjukkan hasil yang lebih baik dengan akurasi dan AUC sebesar 99,9%. Pada data *training*, kernel *linear* menunjukkan hasil yang lebih baik dengan akurasi sebesar 98,3% dan AUC sebesar 97,5%.
4. Secara keseluruhan data yang telah dilakukan SMOTE menunjukkan hasil yang lebih baik dalam mengklasifikasikan data. Pada data *training* performa terbaik ditunjukkan oleh metode SVM kernel RBF dan pada data *testing* performa terbaik ditunjukkan oleh metode SVM *linear*.

5.2 Saran

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

1. PT X dapat melakukan analisis sentimen penumpang pesawat menggunakan metode NBC maupun SVM karena kedua metode tersebut menghasilkan ketepatan klasifikasi yang cukup baik. Selain itu PT X dapat mempertimbangkan hasil visualisasi ulasan menggunakan *word cloud* sebagai bahan pertimbangan untuk mepertahankan serta memberbaiki kualitas layanan penerbangan yang disediakan oleh PT X.
2. Untuk penelitian selanjutnya, penelitian serupa dapat dikembangkan untuk bisa mengklasifikasikan ulasan yang memiliki 2 sentimen sekaligus. Sehingga penumpang yang memberikan ulasan positif serta negatif dalam 1 ulasan dapat ikut terklasifikasi. Penelitian selanjutnya juga dapat dikembangkan untuk mengatasi kelemahan metode NBC pada ulasan yang memiliki jumlah kata sangat sedikit sehingga terjadi misklasifikasi.

DAFTAR PUSTAKA

Abdulkadir, M. (1991). *Hukum Pengangkutan Darat, Laut,dan Udara.* Bandung: Citra Aditya Bakti.

Adji, S. U. (2005). *Hukum Pengangkutan di Indonesia.* Jakarta: PT Rineka Cipta.

Basri, M. (2016). *Identifikasi Topik Informasi Publik Media Sosial di Kota Surabaya Berdasarkan Klasterisasi Teks Pada Twitter dengan Menggunakan Algoritma K-Means.* Surabaya: ITS.

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.

BPS. (2018). *Statistik Transportasi Udara 2018.* BPS RI.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling Technique. *Advances in Knowledge Discovery and Data Mining.* Bangkok.

Castella, Q., & Sutton, C. (2014). Word Storm: Multiples of Word Clouds for Visual Comparison of Documents. *WWW '14: Proceedings of the 23rd international conference on World wide web*, (hal. 665-676).

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. (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: University Press.

Garuda Indonesia. (2020, Januari). *Tentang Garuda Indonesia*. Diambil kembali dari Garuda Indonesia: https://www.garuda-indonesia.com/id/id/corporate-partners/company-profile/about/index.page?

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. (2012). Preprocessing the Informal Text for Efficient Sentiment Analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETICS)*, 58-61.

Hemalatha, I., Varma, G. P., & Govardhan, A. (2012). Preprocessing the Informal Text for Efficient Sentiment Analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 58-61.

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

Kwartler, T. (2017). *Text Mining in Practire With R.* New Jersey: John Wiley & Sons Ltd.

Lebart, L. (1998). Text Mining in Different Languages.

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

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Morgan & Claypool Publishers.

Liu, B. (2015). *Sentiment Analysis Mining Opinions, Sentiments, and Emotions.* United States of America: Cambridge University Press.

Martono, H. K., & Sudiro, A. (2010). *Hukum Angkutan Udara Berdasarkan UU RI No. 1 Tahun 2009.* Jakarta: Rajawali Pers.

McCue, R. (2009). *A Comparison of the Accuracy of Support Vector Machine and Naıve Bayes Algorithms In Spam Classification.* Santa Cruz: University of California at Santa Cruz.

Menon, A., Jaworski, B. J., & Kohli, A. K. (1997). Product Quality: Impact of Interdepartmental Interactions. *Journal of the Academy of Marketing Science 25 (3)*, 187-200.

Musnaini. (2011). Analisis Kualitas Layanan Konsumen Terhadao Keunggulan Bersaing Jasa Transportasi Darat Pada PT. Kereta Api Indonesia (PERSERO) Kelas Argo. *Jurnal Manajemen Teori dan Terapan*, 1.

Republik Indonesia. (2017). Peraturan Menteri Perhubungan. *PM 61 Tahun 2017*.

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

Sarkar, D. (2016). *Text Analytics With Python.* Bangalore, Karnataka: Apress.

Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge: MIT Press.

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

Suriaatmaja, T. T. (2005). *Pengangkutan Kargo Udara Tanggung Jawab Pengangkut Dalam Dimensi Hukum Udara dan Internasional.* Bandung: Pustaka Bani Quraisy.

Susantoro, B., & Parikesit, D. (2004). *1-2-3 Langkah: Langkah Kecil yang Kita Lakukan Menuju Transportasi yang Berkelanjutan.* Jakarta: Majalah Transportasi Indonesia, Vol. 1.

Walpole, R. E. (2007). *Pengantar Statistika*. Jakarta: PT Gramedia Pustaka.

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

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

Yosephine, M., & Prabowo, Y. D. (2016). Pengembangan Aplikasi Pemeriksaan Kata Dasar dan Imbuhan pada Bahasa Indonesia. *Kalbis Scientia Jurnal Sains dan Teknologi*, 118-130.

LAMPIRAN

Lampiran 1. Data Penelitian

No	Ulasan	Sentimen
1	Sangat menikmati pengalaman dengan layanan yang baik terima kasih untuk semua crew atas pelayanannya	1
2	Merasa nyaman menyenangkan pelayanan ramah sopan dan penuh perhatian	1
3	merasa nyaman aman, dan pelayanan memuaskan	1
4	Pelayanan bagus dari awak kabin	1
5	hiburan didalam pesawat khususnya film sangat tidak up to date	0
6	Pelayanan bagus on schedule	1
7	merasa senang terbang bersama garuda indonesia, pramugari ramah baik hati	1
8	Pelayanan ramah baik sopan toilet bersih	1
9	pelayanan semua baik ramah dan merasa terbang nyaman aman	1
10	Cabin crew selalu menyenangkan dalam melayani dan selalu semangat	1
12	Pelayanan sangat baik pramugarinya ramah	1
13	pelayanan baik, merasa nyaman terbang dengan garuda indonesia	1
15	Penerbangan nyaman	1
16	Kualitas makanan menurun	0
17	Business lounge di CGK sangat tidak nyaman karena makanan tidak enak fasilitas dibandara CGK loungenya tidaknyaman dan disejajarkan dengan skyteam	0
…	…	…
1492	Pelayanan baik memuaskan dan makanan enak	1

Lampiran 2. Hasil *Confusion Matrix*

***Confusion Matrix* Metode NBC**

Data *training* tanpa SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	103	17
Positif	2	1072

Data *training* dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	1067	7
Positif	6	1068

Data *testing* tanpa SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	23	6
Positif	2	267

Data *testing* dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	26	3
Positif	3	266

Confusion Matrix* Metode SVM Kernel *Linear

Data *training* tanpa SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	118	2
Positif	0	1074

Data *training* dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	1074	0
Positif	5	1069

Data *testing* tanpa SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	26	3
Positif	4	265

Data *testing* dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	27	2
Positif	4	265

Lampiran 3. Hasil *Confusion Matrix* (Lanjutan)

***Confusion Matrix* Metode SVM Kernel RBF**

Data *training* tanpa SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	118	2
Positif	0	1074

Data *training* dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	1074	0
Positif	3	1071

Data *testing* tanpa SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	26	3
Positif	3	266

Data *testing* dengan SMOTE

Kelas Aktual	Kelas Prediksi	
	Negatif	**Positif**
Negatif	27	2
Positif	4	265

Lampiran 4. Ketepatan Klasifikasi Metode SVM Kernel RBF

Sebelum dilakukan SMOTE

C	Gamma	Akurasi		AUC	
		Training	Testing	Training	Testing
0.01	0.01	0.899	0.903	0.500	0.500
0.01	0.1	0.899	0.903	0.500	0.500
0.01	1	0.899	0.903	0.500	0.500
0.01	10	0.899	0.903	0.500	0.500
0.01	100	0.899	0.903	0.500	0.500
0.1	0.01	0.899	0.903	0.500	0.500
0.1	0.1	0.899	0.903	0.500	0.500
0.1	1	0.905	0.903	0.525	0.500
0.1	10	0.899	0.903	0.500	0.500
0.1	100	0.899	0.903	0.500	0.500
1	0.01	0.899	0.903	0.500	0.500

Lampiran 5. Ketepatan Klasifikasi Metode SVM Kernel RBF (Lanjutan)

Sebelum dilakukan SMOTE

C	Gamma	Akurasi		AUC	
		Training	Testing	Training	Testing
1	0.1	0.936	0.933	0.683	0.671
1	1	0.997	0.977	0.987	0.895
1	10	0.998	0.909	0.992	0.534
1	100	0.998	0.906	0.992	0.517
10	0.01	0.943	0.940	0.717	0.705
10	0.1	0.997	0.977	0.987	0.941
10	1	0.998	0.977	0.992	0.910
10	10	0.998	0.909	0.992	0.534
10	100	0.998	0.909	0.992	0.534
100	0.01	0.996	0.977	0.987	0.941
100	0.1	0.998	0.980	0.992	0.943
100	1	0.998	0.977	0.992	0.910
100	10	0.998	0.909	0.992	0.534
100	100	0.998	0.909	0.992	0.534

Setelah dilakukan SMOTE

C	Gamma	Akurasi		AUC	
		Training	Testing	Training	Testing
0.01	0.01	0.965	0.963	0.965	0.918
0.01	0.1	0.965	0.966	0.965	0.920
0.01	1	0.894	0.943	0.894	0.722
0.01	10	0.553	0.903	0.553	0.500
0.01	100	0.538	0.903	0.538	0.500
0.1	0.01	0.965	0.963	0.965	0.918
0.1	0.1	0.962	0.940	0.962	0.951

Lampiran 6. Ketepatan Klasifikasi Metode SVM Kernel RBF (Lanjutan)

Setelah dilakukan SMOTE

C	Gamma	Akurasi		AUC	
		Training	Testing	Training	Testing
0.1	1	0.994	0.980	0.994	0.927
0.1	10	0.783	0.906	0.783	0.517
0.1	100	0.686	0.906	0.686	0.517
1	0.01	0.960	0.940	0.960	0.951
1	0.1	0.936	0.933	0.683	0.671
1	1	0.999	0.977	0.999	0.910
1	10	0.999	0.909	0.999	0.534
1	100	0.999	0.909	0.999	0.534
10	0.01	0.983	0.980	0.983	0.973
10	0.1	0.998	0.977	0.998	0.956
10	1	0.999	0.977	0.999	0.910
10	10	0.999	0.909	0.999	0.534
10	100	0.999	0.909	0.999	0.534
100	0.01	0.997	0.973	0.997	0.954
100	0.1	0.999	0.980	0.999	0.958
100	1	0.999	0.977	0.999	0.910
100	10	0.999	0.909	0.999	0.534
100	100	0.999	0.909	0.999	0.534

Lampiran 7. *Syntax Input* dan *Preprocessing Data*

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

# Import data
data = pd.read_csv('comment coba.csv')
```

```
x = data['comment'] #get comment
y = data['class'] #get class

# K-fold Cross Validation
from sklearn.model_selection import KFold
kf=KFold(n_splits=5, shuffle=False)
i=1
for train_index, test_index in kf.split(x):
    print("Fold ",i)
    print("TRAIN :",train_index,"TEST :",test_index)
    x_train=x[train_index]
    x_test=x[test_index]
    y_train=y[train_index]
    y_test=y[test_index]
    i+=1

# Case Folding
train_lower = []
for line in x_train:
    a = line.lower()
    train_lower.append(a)
test_lower = []
for line in x_test:
    a = line.lower()
    test_lower.append(a)

# Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
train_stemmed = map(lambda x: stemmer.stem(x), train_lower)
translator = str.maketrans('', '', string.punctuation)
train_no_punc = map(lambda x: x.lower().translate(translator),train_stemmed)
test_stemmed = map(lambda x: stemmer.stem(x), test_lower)
test_no_punc = map(lambda x: x.lower().translate(translator),test_stemmed)

# Stopwords Removal
stopword = open("id.stopwords.txt","r").read()
trainfinal = []
for line in train_no_punc:
    word_token = nltk.word_tokenize(line)
    word_token = [word for word in word_token if not word in stopword and not
word[0].isdigit()]
    trainfinal.append(" ".join(word_token))
testfinal = []
for line in test_no_punc:
    word_token = nltk.word_tokenize(line)
    word_token = [word for word in word_token if not word in stopword and not
word[0].isdigit()]
    testfinal.append(" ".join(word_token))

# Export Pre-Processing Result
```

```
df_xtrain = pd.DataFrame(trainfinal)
df_xtrain.columns=['comment']
df_ytrain = pd.DataFrame(y_train)
df_xytrain = pd.concat([df_xtrain,df_ytrain],axis=1)
df_xytrain.to_csv('outputtrain.csv', header=True, index=False, sep=',')
df_xtest = pd.DataFrame(testfinal)
df_xtest.columns=['comment']
df_ytest = pd.DataFrame(y_test)
df_ytestnew = df_ytest.reset_index(drop=True)
df_xytest = pd.concat([df_xtest,df_ytestnew],axis=1)
df_xytest.to_csv('outputtest.csv', header=True, index=False, sep=',')
```

Lampiran 8. *Syntax* Klasifikasi Data

```
import codecs
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import KFold
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_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
from imblearn.over_sampling import SMOTE

# Import data
datanew = pd.read_csv('outputtrain.csv')
trainfinal = datanew['comment'] #get comment
dftrain_label = datanew['class'] #get class
datanew2 = pd.read_csv('outputtest.csv')
testfinal = datanew2['comment'] #get comment
dftest_label = datanew2['class'] #get class

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)
    sm=SMOTE()
    tfidf_train, training_label = sm.fit_sample(tfidf_train, training_label)
data_train,data_test,target_train,target_test=train_test_split(tfidf_train,training_label,te
st_size=0.2,random_state=43)
    classify=classifier.fit(data_train,target_train)
    scores=model_selection.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))

def predict(training_data,test_data,test_label,classifier):
    # method frot 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

#NAIVE BAYES CLASSIFIER
nb_classifier = BernoulliNB()
nb_classifier2 = BernoulliNB() #MODEL WITH SMOTE

#Training model
count_vectorizer=CountVectorizer(binary=True)
train=count_vectorizer.fit_transform(trainfinal)
tfidf_train=TfidfTransformer(use_idf=True).fit_transform(train)
classify=nb_classifier.fit(tfidf_train,dftrain_label)
ypred_train=nb_classifier.predict(tfidf_train)

#predict new testing data
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)
prediction1                                                                    =
pd.DataFrame(predicted1,columns=['predictions']).to_csv("naivebayes.csv")

#Training model WITH SMOTE
sm=SMOTE()
tfidf_train2, dftrain_label2 = sm.fit_sample(tfidf_train, dftrain_label)
classify2=nb_classifier2.fit(tfidf_train2,dftrain_label2)
ypred_train2=nb_classifier2.predict(tfidf_train2)

#predict new testing data WITH SMOTE
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)
predicted2 = nb_classifier2.predict(test_data_clean)
prediction2  =  pd.DataFrame(predicted2,columns=['predictions']).to_csv("naivebayes
WITH SMOTE.csv")

#HASIL NBC
print("HASIL NBC")
print(" ")
```

```
print("AKURASI TRAINING: ", accuracy_score(dftrain_label,ypred_train))
print("AKURASI TESTING: ", accuracy_score(dftest_label,predicted1))
print("AUC TRAINING: ", roc_auc_score(dftrain_label,ypred_train))
print("AUC TESTING: ", roc_auc_score(dftest_label,predicted1))
print(" ")
print("CONF MATRIX TRAINING NBC")
print(confusion_matrix(dftrain_label,ypred_train))
print(" ")
print("CONF MATRIX TESTING NBC")
print(confusion_matrix(dftest_label,predicted1))

print(" ")
print("============================================================
=")
print(" ")

#HASIL NBC SMOTED
print("HASIL NBC SMOTED")
print(" ")
print("AKURASI TRAINING SMOTED: ",
accuracy_score(dftrain_label2,ypred_train2))
print("AKURASI TESTING SMOTED: ", accuracy_score(dftest_label,predicted2))
print("AUC TRAINING SMOTED: ", roc_auc_score(dftrain_label2,ypred_train2))
print("AUC TESTING SMOTED: ", roc_auc_score(dftest_label,predicted2))
print(" ")
print("CONF MATRIX SMOTED TRAINING NBC")
print(confusion_matrix(dftrain_label2,ypred_train2))
print(" ")
print("CONF MATRIX SMOTED TESTING NBC")
print(confusion_matrix(dftest_label,predicted2))

#SUPPORT VECTOR MACHINE
svm_classifier = SVC(kernel='rbf',C=100,gamma=100)
svm_classifier2 = SVC(kernel='rbf',C=100,gamma=100)

#Training model
count_vectorizerr=CountVectorizer(binary=True)
train=count_vectorizerr.fit_transform(trainfinal)
tfidf_train=TfidfTransformer(use_idf=True).fit_transform(train)
classifyy=svm_classifier.fit(tfidf_train,dftrain_label)
ypred_trainn=svm_classifier.predict(tfidf_train)

#predict new testing data
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)
predicted3 = svm_classifier.predict(test_data_clean)
prediction3 = pd.DataFrame(predicted3,columns=['predictions']).to_csv("svm.csv")
```

```
#Training model WITH SMOTE
classifyy2=svm_classifier2.fit(tfidf_train2,dftrain_label2)
ypred_trainn2=svm_classifier2.predict(tfidf_train2)

#predict new testing data
predictedd = testfinal
df_testt = predictedd
count_vectorizer = CountVectorizer(binary=True)
count_vectorizer.fit_transform(trainfinal)
test_dataa = count_vectorizer.transform(df_testt)
test_data_cleann = TfidfTransformer(use_idf=True).fit_transform(test_dataa)
predicted4 = svm_classifier2.predict(test_data_cleann)
prediction4 = pd.DataFrame(predicted4,columns=['predictions']).to_csv("svm WITH
SMOTE.csv")

#HASIL SVM
print("HASIL SVM")
print(" ")
print("AKURASI TRAINING: ", accuracy_score(dftrain_label,ypred_trainn))
print("AKURASI TESTING: ", accuracy_score(dftest_label,predicted3))
print("AUC TRAINING: ", roc_auc_score(dftrain_label,ypred_trainn))
print("AUC TESTING: ", roc_auc_score(dftest_label,predicted3))
print(" ")
print("CONF MATRIX TRAINING SVM")
print(confusion_matrix(dftrain_label,ypred_trainn))
print(" ")
print("CONF MATRIX TESTING SVM")
print(confusion_matrix(dftest_label,predicted3))

print(" ")
print("==========================================================
=")
print(" ")

#HASIL SVM SMOTED
print("HASIL SVM SMOTED")
print(" ")
print("AKURASI TRAINING SMOTED: ",
accuracy_score(dftrain_label2,ypred_trainn2))
print("AKURASI TESTING SMOTED: ", accuracy_score(dftest_label,predicted4))
print("AUC TRAINING SMOTED: ", roc_auc_score(dftrain_label2,ypred_trainn2))
print("AUC TESTING SMOTED: ", roc_auc_score(dftest_label,predicted4))
print(" ")
print("CONF MATRIX SMOTED TRAINING SVM")
print(confusion_matrix(dftrain_label2,ypred_trainn2))
print(" ")
print("CONF MATRIX SMOTED TESTING SVM")
print(confusion_matrix(dftest_label,predicted4))
```

Lampiran 9. *Syntax World Cloud* Menggunakan RStudio

```
library("tm")
library("RColorBrewer")
library("wordcloud")
library("dplyr")
library("stringr")

docs<-readLines("comment coba pos.csv")
docs <- Corpus(VectorSource(docs))
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/")
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "\\|")

docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, toSpace, "[[:punct:]]")
docs <- tm_map(docs, toSpace, "[[:digit:]]")
docs <- tm_map(docs, stripWhitespace)

#Load file Stopword, dll
myStopwords = readLines("id.stopwords.txt")
#load slangword
slang <- read.csv("Slangword.csv", header=T)
old_slang <- as.character(slang$old)
new_slang <- as.character(slang$new)
#load stemming
stemm <- read.csv("Stemming.csv", header=T)
old_stemm <- as.character(stemm$old)
new_stemm <- as.character(stemm$new)
#load lemmatization
lemma <- read.csv("Lemmatization.csv", header=T)
old_lemma <- as.character(stemm$old)
new_lemma <- as.character(stemm$new)

stemmword <- function(x) Reduce(function(x,r)
gsub(stemm$old[r],stemm$new[r],x,fixed=T),
                seq_len(nrow(stemm)),x)
docs <- tm_map(docs,stemmword)
slangword <- function(x) Reduce(function(x,r) gsub(slang$old[r],slang$new[r],x,fixed=T),
                seq_len(nrow(slang)),x)
docs <- tm_map(docs,slangword)
lemmatization <- function(x) Reduce(function(x,r)
gsub(lemma$old[r],lemma$new[r],x,fixed=T),
                  seq_len(nrow(lemma)),x)
docs <- tm_map(docs,lemmatization)
docs <- tm_map(docs, removeWords, myStopwords)

dataframe=data.frame(text=unlist(sapply(docs, `[`)), stringsAsFactors=F)
write.csv(dataframe,file = 'DataClean.csv')

docs.for.find.word=read.csv('DataClean.csv', header = T)
```

```
View(docs.for.find.word)
docs.for.find.word=docs.for.find.word$text
find.word<- Corpus(VectorSource(docs.for.find.word))

myCorpus=find.word
tdm <- TermDocumentMatrix(myCorpus,
                          control = list(wordLengths = c(1, Inf)))
term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 1)
df <- data.frame(term = names(term.freq), freq = term.freq)
df <- df[with(df, order(-freq)), ]
n=dim(df)[1]
df <- data.frame(no=1:n, df)
View(df)

dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)

dtm <- TermDocumentMatrix(docs)
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words=51, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```

BIODATA PENULIS

Bima Putra Goklas biasa dipanggil dengan nama Bima yang merupakan anak pertama dari tiga bersaudara dan dilahirkan di Jakarta pada tanggal 20 April 1998. Pendidikan yang telah ditempuh oleh penulis adalah SDK Kalam Kudus Bandung (2004-2010), SMPK Kalam Kudus Bandung (2010-2013), dan SMA Negeri 8 Bandung (2013-2016). Kemudian dilanjutkan dengan menempuh pendidikan di Institut Teknologi Sepuluh Nopember Departemen Statistika. Untuk memperluas ilmu yang didapat, penulis mendapatkan kesempatan menjadi mahasiswa terpilih untuk mengikuti Asia University Taiwan *Summer Exchange Program* 2020 dengan *course Artificial Intelligence.* Selain dalam bidang akademik, penulis juga aktif organisasi di Badan Eksekutif Mahasiswa (BEM) ITS sebagai Direktur Jenderal Komunikasi Internasional dan Kerjasama ASEAN periode 2019/2020 dan Himpunan Mahasiswa Bandung-ITS sebagai Ketua Himpunan periode 2018/2019. Selain itu, penulis juga aktif dalam mengikuti kepanitiaan yang diadakan oleh tingkat jurusan, ITS, maupun nasional dan internasional seperti menjadi Ketua Pelaksana dalam kompetisi analisis data *Data Analysis Competition* 2018 *Southeast Asia* dan Ketua Pelaksaa ITS *International Fair* 2018. Selama menjalani perkuliahan penulis juga berkesempatan dalam menjalani program *internship* di PT. Garuda Indonesia (Persero) Tbk. Penulis juga pernah mengikuti kegiatan survei sebagai pengaplikasian ilmu statistika. Jika ingin memberikan saran, kritik, dan diskusi lebih lanjut, dapat menghubungi penulis melalui email: bima.goklas@gmail.com.