TUGAS AKHIR – KS184822

TEXT ANALYTICS PADA MASKAPAI PENERBANGAN INDONESIA DI *TWITTER* MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*

CAHYA BUANA PUTRI
NRP 062115 4000 0112

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari, M.Si

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

TUGAS AKHIR – KS184822

TEXT ANALYTICS PADA MASKAPAI PENERBANGAN INDONESIA DI *TWITTER* MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*

CAHYA BUANA PUTRI
NRP 062115 4000 0112

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari, M.Si

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

FINAL PROJECT – KS184822

TEXT ANALYTICS OF INDONESIAN AIRLINES ON TWITTER USING SUPPORT VECTOR MACHINE METHOD

CAHYA BUANA PUTRI
SN 062115 4000 0112

Supervisor
Dr. Dra. Kartika Fithriasari, M.Si

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TEXT ANALYTICS PADA MASKAPAI PENERBANGAN INDONESIA DI *TWITTER* MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*

TUGAS AKHIR

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

Oleh :
Cahya Buana Putri
NRP. 062115 4000 0112

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

Mengetahui,
Kepala Departemen Statistika

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI INSTITUT TEKNOLOGI SEPULUH NOPEMBER FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA DEPARTEMEN STATISTIKA

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2019

(Halaman ini sengaja dikosongkan)

TEXT ANALYTICS PADA MASKAPAI PENERBANGAN INDONESIA DI *TWITTER* MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*

Nama Mahasiswa	**:**	**Cahya Buana Putri**
NRP	**:**	**062115 4000 0112**
Departemen	**:**	**Statistika**
Dosen Pembimbing	**:**	**Dr. Dra. Kartika Fithriasari, M.Si**

Abstrak

Seiring dengan perkembangan teknologi dan media online yang cepat, komentar dan komplain pelanggan dapat dilihat dari berbagai media sosial, termasuk Twitter yang sering digunakan sebagai media berkomunikasi dengan pelanggan oleh pelaku bisnis termasuk di bidang transportasi udara. Maskapai penerbangan Indonesia yang digunakan dalam penelitian ini adalah AirAsia dan LionAir. Penelitian ini menerapkan pendekatan text mining, dimana setiap tweet bersentimen akan dilakukan klasifikasi menggunakan Support Vector Machine dengan Synthetic Minority Over-sampling Technique (SMOTE) untuk menangani kasus imbalanced data dan n-gram sebagai tahap tokenizing pada preprocessing. Hasil klasifikasi dievaluasi menggunakan nilai AUC (Area Under Curve). Model klasifikasi terpilih adalah hasil klasifikasi yang memberikan nilai AUC paling tinggi. Tweet yang diajukan kepada LionAir didominasi oleh tweet bersentimen negatif. Proporsi tweet positif dan negatif untuk AirAsia cenderung seimbang. Berdasarkan analisis kompetitif, pandangan publik di Twitter kepada AirAsia cenderung lebih baik dibanding LionAir. Hal tersebut disebabkan AirAsia lebih sering mengadakan promo dan kuis berhadiah. Penelitian ini juga menunjukkan bahwa kernel RBF lebih baik dibanding linier untuk kedua kasus. Metode SMOTE juga mampu meningkatkan nilai AUC dalam kasus imbalanced data. Melalui hasil klasifikasi pada penelitian ini, AirAsia dan LionAir diharapkan mampu mendapatkan informasi mengenai pendapat publik di Twitter secara otomatis sehingga dapat melakukan tindakan perbaikan maupun peningkatan kualitas pelayanan dengan cepat.

Kata kunci: AirAsia, LionAir, N-gram, SMOTE, Support Vector Machine

(Halaman ini sengaja dikosongkan)

TEXT ANALYTICS OF INDONESIAN AIRLINES ON TWITTER USING SUPPORT VECTOR MACHINE METHOD

Name	**:**	**Cahya Buana Putri**
Student Number	**:**	**062115 4000 0112**
Department	**:**	**Statistics**
Supervisor	**:**	**Dr. Dra. Kartika Fithriasari, M.Si**

Abstract

Along with the rapid development of technology and online media, public comments and complaints can be monitored from social media, including Twitter which is often used as a communication media in the business field. AirAsia and LionAir are two airlines operating in Indonesia whose one of their communication media is Twitter. A text mining approach was applied in this study, where the sentiment tweets will be classified using Support Vector Machine with SMOTE (Synthetic Minority Over-sampling Technique) to handle imbalanced data. N-gram concept is applied as well in the pre processing step. The classifier are being evaluated by AUC (Area Under Curve). Therefore, the selected classification model is the one gives the highest AUC value. LionAir's tweets were dominated by negative ones. The proportion of positive and negative tweets for AirAsia tends to be balanced. Based on competitive analysis, public views to AirAsia on Twitter tend to be better than LionAir due to promos and quizzes with prices held by AirAsia. This study shows that the performance of RBF Kernel classification is better than Linear's for both cases. It is also shown that SMOTE method is able to increase the AUC value for imbalanced data case. Thus, both AirAsia and LionAir are expected to be able to get information of public opinion on Twitter automatically so that corrective actions and service quality improvements can be carried out quickly.

Keywords: AirAsia, LionAir, N-gram, SMOTE, Support Vector Machine

(This page intentionally left blank)

KATA PENGANTAR

Puji syukur penulis panjatkan atas rahmat dan hidayah yang diberikan Allah SWT sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul "*Text Analytics* pada Maskapai Penerbangan Indonesia Menggunakan Metode *Support Vector Machine*" dengan lancar.

Penulis menyadari bahwa penyusunan Tugas Akhir ini dapat terselesaikan karena bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua, adik, kakak, dan keluarga besar atas segala do'a, dukungan, dan kasih sayang yang diberikan kepada penulis demi kesuksesan dan kebahagiaan penulis.
2. Ibu Dr. Dra. Kartika Fithriasari, M.Si selaku dosen pembimbing yang telah memberikan bimbingan dengan meluangkan waktu, pengarahan, saran, dan dukungan selama penyusunan Tugas Akhir ini.
3. Bapak Prof. Drs. Nur Iriawan, M.Ikom, Ph.D, Ibu Adatul Mukarromah, S.Si., M.Si, dan Bapak Novri Suhermi, S.Si., M.Sc selaku dosen penguji yang telah memberi banyak saran dalam penyelesaian Tugas Akhir.
4. Bapak Dr. Suhartono selaku Ketua Departemen Statistika dan Ibu Dr. Santi Wulan Purnami, S.Si., M.Si selaku Ketua Program Studi Sarjana yang telah memberikan fasilitas, sarana, dan prasarana demi menunjang kelancaran penyelesaian Tugas Akhir.
5. Ibu Erma Oktania Permatasari, S.Si., M.Si selaku dosen wali selama masa studi yang telah banyak membantu serta mendampingi, memberikan motivasi, saran hingga arahan dalam proses belajar di Departemen Statistika.
6. Seluruh civitas akademika Departemen Statistika ITS yang telah membantu dalam kelancaran penyelesaian Tugas Akhir ini serta memberikan ilmu dan pengetahuan.
7. Teman-teman Statistika ITS ∑26 angkatan 2015, yang selalu memberikan dukungan kepada penulis selama ini.

8. Semua teman, relasi dan berbagai pihak yang tidak bisa penulis sebutkan namanya satu persatu yang telah membantu dalam penulisan laporan ini.

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

Surabaya, Mei 2019

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Halaman

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

Halaman

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi khususnya di bidang informasi kini dapat memberikan manfaat lebih serta peluang untuk mengembangkan sistem yang dapat memahami opini masyarakat secara otomatis melalui internet. Salah satu metode yang digunakan adalah Analisis Sentimen yang didukung oleh *Text Mining*. Analisis sentimen atau *opinion mining* adalah studi komputasional dari opini-opini orang, sentimen dan emosi melalui entitas dan atribut yang dimiliki serta diekspresikan dalam bentuk teks (Liu, 2012). Analisis sentimen akan mengelompokkan polaritas dari teks yang ada dalam kalimat atau dokumen untuk mengetahui pendapat yang dikemukakan apakah bersifat positif, negatif, atau netral (Pang & Lee, 2008).

Selain itu, teknologi merupakan salah satu pemanfaatan yang tepat dalam membuka akses bagi pelanggan. Seiring dengan perkembangan teknologi dan media *online* yang cepat, komentar dan komplain pelanggan dapat dilihat dari berbagai media sosial. Dari total penduduk Indonesia, lebih dari setengahnya telah menggunakan internet dalam kegiatan sehari-harinya dan terdapat 150 juta jiwa yang telah aktif menggunakan media sosial. Salah satu media sosial yang digemari adalah *Twitter*. Di awal tahun 2019, *Twitter* merupakan *social media platforms* berbasis *social network* peringkat keempat paling aktif di Indonesia setelah *YouTube*, *Facebook* dan *Instagram*, dimana sekitar 52% pengguna internet memiliki *Twitter* (WeAreSocial, 2019). *Twitter* menyediakan fitur-fitur yang cukup mudah digunakan sehingga memudahkan penggunanya untuk berkicau. *Twitter* juga sering digunakan oleh berbagai perusahaan atau pelaku bisnis sebagai sarana pemuasan pelanggan dalam menanggapi saran, kritik, maupun komplain oleh berbagai perusahaan atau pelaku bisnis, termasuk di bidang transportasi.

Transportasi merupakan sarana penting dalam menunjang keberhasilan pembangunan terutama dalam mendukung kegiatan

perekomian masyarakat. Dewasa ini perkembangan transportasi menimbulkan beragam tipe transportasi publik baik di darat, air, maupun udara. Seiring dengan meningkatnya jumlah transportasi publik maka meningkat pula kompetisi antara pelaku bisnis di bidang transportasi. Ukuran yang seharusnya ditekankan dalam bisnis pelayanan transportasi adalah pelayanan itu sendiri. Pelayanan untuk memuaskan pelanggan merupakan aspek vital demi bertahan dan memenangkan dalam kompetisi berbisnis (Tjiptono, 2005). Beberapa perusahaan transportasi udara yang beroperasi di Indonesia antara lain adalah AirAsia dan LionAir.

AirAsia merupakan maskapai penerbangan swasta berbiaya rendah yang berpusat di Kuala Lumpur, Malaysia. AirAsia melebarkan jaringan rute hingga ke luar Malaysia termasuk Indonesia. Indonesia AirAsia berbasis di Jakarta dengan mengoperasikan penerbangan domestik terjadwal. AirAsia berhasil mencatatkan beberapa pengakuan internasional diantaranya adalah pencapaian sebagai "*World's Best Low-Cost Airlines*" atau maskapai berbiaya rendah terbaik dunia dengan rekor hingga ke-sepuluh kali pada tahun 2018 (Skytrax, 2018). Selain itu di tahun yang sama, AirAsia juga berhasil selama enam kali menyabet gelar "*Asia's Best Low-Cost Airlines*" (Ratnasari & Sankhyaadi, 2018). AirAsia menjadikan dirinya sebagai pemain regional yang akan berkompetisi dengan LionAir dari Indonesia. LionAir merupakan maskapai penerbangan swasta nasional asal Indonesia yang beroperasi sejak tahun 2000 yang juga mengusung penerbangan berbiaya rendah. Tahun 2017, LionAir Group yang terdiri atas Lion, Batik, dan Wings Air untuk pertama kalinya menyalip supremasi Garuda Group (Garuda Indonesia dan Citilink) dengan menguasai 51% pangsa pasar domestik, sementara Garuda Group hanya menggenggam 33% (Subastian & Nurjanah, 2018). Sebagai kompetitor kuat AirAsia di tingkat regional di Indonesia, berdasarkan situs pemeringkat maskapai airlineratings.com yang dilakukan oleh Organisasi Penerbangan Sipil Internasional (ICAO), LionAir berhasil menduduki peringkat ketiga dalam keamanan di Indonesia pada tahun 2018 (Fauzia, 2018).

Berdasarkan hal tersebut maka dalam penelitian ini akan diterapkan pendekatan *text mining* agar mampu membantu AirAsia dan LionAir dalam mendapatkan informasi mengenai pendapat publik di *Twitter* secara otomatis dan cepat sehingga dapat melakukan tindakan perbaikan maupun peningkatan kualitas pelayanan dengan cepat pula. Salah satu metode dalam *text mining* yang sering digunakan adalah klasifikasi. Klasifikasi pada *text mining* merupakan metode *supervised* yang mengekstraksi model yang menggambarkan kelas pada data (Han, Kamber, & Pei, 2012). Klasifikasi data ulasan berupa teks akan memudahkan pihak terkait dalam mendapatkan informasi mengenai pendapat pelanggannya. Dalam ilmu statistika, metode klasifikasi yang sering digunakan pada data teks antara lain adalah *Naive Bayes Classifier*, *Support Vector Machine*, dan *k-Nearest Neighbor*. Namun dalam penelitian ini akan digunakan metode *Support Vector Machine*. *Support Vector Machine* (SVM) adalah metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan pemisah antar kategori (*hyperplane*) terbaik (Susilowati, Sabariah, & Gozali, 2015). Dalam *text mining* sering ditemui kondisi dimana distribusi kelas data tidak seimbang, jumlah kelas data (*instance*) yang satu lebih sedikit atau lebih banyak dibanding jumlah kelas data lainnya (Ali, Shamsuddin, & Ralescu, 2013). Kondisi tersebut kemudian menjadi masalah utama pada bidang *machine learning* khususnya pada klasifikasi data berupa teks. Sehingga dalam penelitian akan digunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE) untuk mengatasi masalah jumlah kelas data yang tidak seimbang. SMOTE merupakan metode dengan menerapkan teknik sintesis sampel baru dari kelas minoritas (kelas data dengan jumlah yang lebih sedikit) untuk menyeimbangkan dataset dengan cara *resampling* pada kelas minoritas tersebut (Siringoringo, 2018).

Penelitian oleh Chandani, V., Wahono, Romi S., dan Purwanto (2015) dengan judul Komparasi Algoritma Klasifikasi *Machine Learning* dan *Feature Selection* pada Analisis Sentimen *Review Film* menerapkan metode *Naive Bayes Classifier, Artificial*

Neural Network, dan *Support Vector Machine* (SVM). Penelitian tersebut menyimpulkan bahwa metode SVM menghasilkan akurasi tertinggi untuk klasifikasi sentimen *film review* di situs Internet Movie Database (IMDb) dengan menerapkan metode *information gain* sebagai pemilihan fitur. Penelitian serupa oleh Buntoro (2017) menerapkan *Support Vector Machine* sebagai metode klasifikasi sentimen Calon Gubernur DKI Jakarta 2017 di *Twitter*. Penelitian tersebut menggunakan metode *Lexicon Based* berbahasa Indonesia untuk menentukan kelas sentimen (positif, negatif, atau netral) dan tokenisasi, *cleansing*, serta *filtering* sebagai tahap *pre processing* dan menghasilkan akurasi, presisi, dan *recall* ketepatan klasifikasi sebesar 90%. Penelitian lain yang serupa oleh Rahman (2018) menyimpulkan bahwa klasifikasi emosi teks berbahasa Indonesia pada pengguna *Twitter* tentang Presiden Joko Widodo terbaik dihasilkan oleh metode *Support Vector Machine* dengan kernel *Radial Basis Function* (RBF) dengan skor akurasi 95,2% dibanding metode kNN (*k-Nearest Neighbor*).

Semua algoritma di *machine learning* baik *supervised* maupun *unsupervised*, biasanya didahului dengan tahap *pre processing*. Tahapan tersebut berguna untuk mengurangi *noise*, menghomogenisasi kata serta mengurangi volume kata (Reyhana, dkk, 2018). Dalam penelitian ini diterapkan metode *n*-gram sebagai tahap *tokenizing* pada *pre processing*. *Tokenizing* dengan menggunakan *n*-gram akan memisahkan deretan kata di dalam satu kalimat, paragraf, atau halaman menjadi token unigram, bigram, dan trigram. Penggunaan *n*-gram ini digunakan berdasarkan hasil penelitian Indhiarta (2017) yang menunjukkan bahwa penggunaan bigram dapat meningkatkan akurasi ketepatan hasil klasifikasi sentimen mengenai pemilihan kepala daerah Jakarta. Penelitian ini menggunakan indikator berupa AUC (*Area Under Curve*) untuk mengukur kebaikan *classifier* (Iriawan, dkk., 2018).

Berdasarkan uraian di atas maka perlu meninjau suara pelanggan baik berupa saran, kritik, maupun komplain untuk mengetahui bagaimana kepuasan pelanggan terhadap pelayanan yang telah diberikan oleh maskapai AirAsia dan LionAir. Oleh karena

itu, dalam penelitian ini dilakukan *text analytics* oleh pengguna *Twitter* mengenai maskapai penerbangan AirAsia dan LionAir. Proses pemberian label sentimen akan dilakukan menurut subjektifitas peneliti. Selanjutnya akan dilakukan klasifikasi dengan menggunakan metode *Support Vector Machine* dengan *Synthetic Minority Over-sampling Technique* untuk menangani kasus *imbalanced data* dan *n*-gram sebagai tahap *tokenizing* pada *pre processing*. Sementara karakteristik sentimen akan dideskripsikan secara visual dengan *Wordcloud*. Dengan demikian, manajemen maskapai AirAsia dan LionAir diharapkan dapat mengetahui reputasinya serta meningkatkan kualitas pelayanannya.

1.2 Rumusan Masalah

Twitter sebagai media pelayanan dan sarana komunikasi dipilih oleh maskapai penerbangan AirAsia dan LionAir untuk menerima serta menanggapi pertanyaan, saran, maupun komplain dari pelanggannya. Berdasarkan penjelasan tersebut maka permasalahan dalam penelitian ini adalah bagaimana karakteristik sentimen pengguna *Twitter* mengenai kedua maskapai penerbangan tersebut, berapa tingkat ketepatan hasil klasifikasi sentimen pengguna *Twitter* mengenai AirAsia dan LionAir menggunakan *Support Vector Machine* dengan menerapkan SMOTE pada kasus *imbalanced data* serta *n*-gram pada tahap tokenisasi, bagaimana hasil komparasi ketepatan klasifikasi yang telah diperoleh, serta bagaimana hasil analisis kompetitif berdasarkan *tweet* untuk Air-Asia dan LionAir.

1.3 Tujuan Penelitian

Tujuan yang akan dicapai dari penelitian pada tugas akhir ini, antara lain adalah:

1. Mendeskripsikan karakteristik sentimen pengguna *Twitter* mengenai AirAsia dan LionAir menggunakan visualisasi grafik dan *Wordcloud*.
2. Mendapatkan hasil ketepatan klasifikasi sentimen pengguna *Twitter* mengenai AirAsia dan LionAir menggunakan metode *Support Vector Machine* dengan menerapkan SMOTE dan tiga jenis *n*-gram.

3. Membandingkan hasil ketepatan klasifikasi dengan penerapan SMOTE dan tiga jenis *n*-gram pada metode *Support Vector Machine* yang telah diperoleh.
4. Menganalisis kompetisi AirAsia dan LionAir berdasarkan *tweet* yang dikicaukan pengguna *Twitter*.

1.4 Manfaat Penelitian

Berdasarkan permasalahan dan tujuan yang telah dijelaskan, manfaat yang diharapkan dari penelitian ini adalah dapat memberi gambaran umum opini pengguna *Twitter* pada maskapai penerbangan AirAsia dan LionAir sehingga kedua pihak terkait tersebut dapat mengetahui reputasinya, melihat perbandingan secara kompetitif, serta meningkatkan kualitas pelayanan untuk memuaskan pelanggannya. Selain itu, kedua maskapai penerbangan tersebut dapat mempercepat proses klasifikasi tanggapan masyarakat (sentimen) karena telah mendapat model dari data latih. Bagi masyarakat umum, hasil penelitian pada tugas akhir ini diharapkan dapat memberikan informasi mengenai kedua maskapai penerbangan.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini adalah data yang digunakan adalah *tweet* atau kicauan yang ditujukan kepada akun resmi *Twitter* milik AirAsia (@AirAsia_indo) dan LionAir (@lionairgroup).

BAB II
TINJAUAN PUSTAKA

Bab ini membahas mengenai *text mining*, *twitter*, analisis sentimen, *wordcloud*, *text pre processing*, *term frequency-inverse document frequency* (TF-IDF), *synthetic minority over-sampling technique* (SMOTE), *k-fold cross validation*, *support vector machine* (SVM), dan evaluasi model klasifikasi.

2.1 *Text Mining*

Text mining adalah proses penemuan akan informasi atau tren baru yang sebelumnya tidak terungkap dengan memproses dan menganalisa data dalam jumlah besar. *Text mining* merupakan penggalian data yang berupa teks yang didapatkan dari dokumen atau kumpulan kalimat yang memiliki tujuan mencari inti dari konten dan selanjutnya dianalisa untuk didapatkan sebuah informasi. *Text mining* merupakan area yang menarik dari penelitian ilmu komputer karena dapat mengatasi krisis informasi yang berlebihan dengan menggabungkan teknik *Data Mining*, *Machine Learning*, *Natural Language Processing*, *Information Retrieval*, Statistik dan Matematik (Sulistyo, 2008). Dalam menganalisa sebagian atau keseluruhan *unstructured text*, *text mining* mencoba mengasosiasikan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu. Hasil yang diharapkan adalah informasi baru atau insight yang belum terungkap.

2.2 *Twitter*

Twitter adalah layanan jejaring sosial dan mikroblog daring yang memungkinkan penggunanya mengirim dan membaca pesan berbasis teks hingga 280 karakter yang dikenal dengan sebutan kicauan (*tweet*). *Twitter* didirikan oleh Jack Dorsey pada bulan Maret 2006, sementara situs jejaring sosialnya diluncurkan pada bulan Juli. *Twitter* mengalami pertumbuhan yang pesat dan sangat cepat menjadi media sosial populer hingga ke seluruh dunia. Indonesia merupakan salah satu negara dengan pengguna *Twitter* terbanyak di dunia. Pada Januari 2019, *Twitter* merupakan *social media platform* yang paling aktif keempat di Indonesia, setelah

Youtube, *Facebook*, dan *Instagram*, dimana sekitar 52% pengguna internet di Indonesia memiliki *Twitter* (WeAreSocial, 2019).

Twitter menyediakan API (*Application Programming Interface*) untuk dapat mengakses data seperti teks kicauan (*tweet*) oleh pengguna *Twitter*, identitas teks yang dikicaukan berupa nomer (status ID), tanggal *tweet* dikicaukan, dan lain sebagainya. Beragam tipe data *Twitter* tersebut dapat memberi informasi baru yang sebelumnya belum terungkap, salah satunya adalah dengan menganalisis sentimen pada *tweet* atau kicauan yang berupa teks. Metode untuk mengumpulkan data *Twitter* biasa dikenal sebagai *Crawling Twitter*.

2.3 Analisis Sentimen

Sentimen adalah pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu. Analisis sentimen atau biasa disebut *opinion mining* merupakan salah satu cabang penelitian *text mining*. Analisis sentimen merupakan riset komputasional dari opini, sentimen, atau emosi yang diekspresikan secara tekstual. Jika diberikan suatu document set berupa teks yang berisi opini mengenai suatu objek, maka *opinion mining* bertujuan untuk mengesktrak atribut dan komponen dari objek yang telah dikomentasi pada setiap dokumen dan untuk menentukan apakah komentar tersebut bermakna positif atau negatif (Kurniasari, 2018). Proses pelabelan data pada penelitian ini dilakukan secara manual, yaitu menurut pandangan peneliti. Sentimen yang digunakan berupa sentimen positif dan negatif.

Konsep pelabelan sentimen pada data secara manual yang dilakukan dalam penelitian ini dapat dianalogikan dengan konsep analisis regresi. Analisis regresi adalah salah satu analisis statistik yang paling populer yang mempelajari hubungan antara satu atau lebih variabel. Variabel yang memengaruhi sering dikenal sebagai variabel prediktor atau independen, sementara variabel yang dipengaruhi disebut sebagai variabel respon atau dependen. Dalam proses pelabelan, sentimen menjadi variabel yang dipengaruhi atau respon dimana terdapat dua jenis sentimen yaitu positif atau negatif. Sementara itu, kata-kata dalam *tweet* menjadi variabel

yang memengaruhi respon yang dihasilkan apakah *tweet* tersebut bersentimen positif atau negatif. Kata-kata yang menjadi variabel prediktor tersebut memiliki hubungan (korelasi) dengan kelas sentimen, sehingga dapat dikatakan bahwa kata-kata dalam sebuah *tweet* menjadi pertimbangan jenis sentimen yang dihasilkan.

2.4 *Wordcloud*

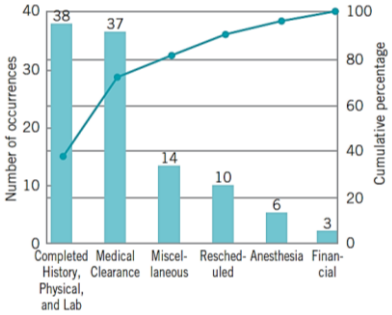
Karakteristik sentimen dapat divisualisasikan menggunakan *Wordcloud*. *Wordcloud* merupakan sebuah sistem yang memunculkan visualisasi kata-kata dengan memberikan penekanan pada frekuensi kemunculan kata terkait dalam suatu wacana tertulis. Frekuensi kemunculan kata ditunjukkan melalui ukuran *font* pada kata tersebut, semakin sering muncul maka ukuran *font* akan semakin besar. Pemakaian *wordcloud* dalam analisis wacana dapat memudahkan karena mampu memberikan gambaran garis besar isi teks secara cepat (Qeis, 2015). Gambar 2.1 merupakan salah satu contoh visualisasi dokumen teks dengan *wordcloud*.

Gambar 2.1 Contoh *Wordcloud*
(Sumber: Kurniasari, 2018)

2.5 *Pareto Chart*

Diagram Pareto atau yang biasa disebut *Pareto chart* merupakan salah satu alat yang sering digunakan dalam bidang industri, khususnya dalam mengevaluasi kecacatan suatu produk. *Pareto chart* biasa digunakan untuk mengidentifikasi jenis cacat apa yang paling sering terjadi sehingga akan menjadi prioritas untuk perbaikan. Dalam penelitian ini, *pareto chart* juga dapat dimanfaatkan untuk mengetahui garis besar isi sebuah teks. *Pareto chart* merupa-

kan diagram yang menggambarkan distribusi frekuensi sederhana dengan data atribut disusun berdasarkan kategori (Montgomery, 2009). Berikut pada Gambar 2.2 adalah contoh Diagram Pareto.



Gambar 2.2 Contoh *Pareto Chart*
(Sumber: Montgomery, 2009)

2.6 *Text Pre Processing*

Tahap *pre processing* adalah tahap dimana dilakukan persiapan berupa pembersihan data sehingga siap dianalisis lebih lanjut. Berikut merupakan langkah-langkah yang dilakukan.

1. *Cleaning*
   Menghapus simbol yang tidak diperlukan pada data *Twitter* antara lain simbol *retweet* (RT), simbol *hashtag* (#), *mention* dan *user-name* (*@username*), *link URL*, serta simbol lainnya seperti tanda baca dan angka.
2. *Case Folding*
   Case folding merupakan proses penyamaan *case* dalam sebuah dokumen untuk memudahkan pencarian dengan mengubah semua karakter menjadi huruf kecil atau huruf besar.
3. Penghilangan *Stopwords*
   Stopword didefinisikan sebagai *term* yang tidak berhubungan (*irrelevant*) dengan subyek utama dari database meskipun kata tersebut sering kali hadir dalam dokumen. Berikut adalah contoh *stopword* dalam bahasa Indonesia: yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, dan lain-lain. *Stopwords* perlu dihilangkan karena dapat mengurangi ukuran penyimpanan dari koleksi yang diindeks.

4. *Stemming*
   Kata-kata yang muncul dalam dokumen sering mempunyai varian morfologik yang tinggi. Setiap kata yang bukan *stop-words* direduksi dalam bentuk *stemmed word* yang tepat. Sehingga *stemming* merupakan tahapan pemotongan kata-kata menjadi kata dasarnya dengan menghilangkan awalan, akhiran, sisipan, dan konfiks. Pada penelitian ini algoritma untuk stemming digunakan algoritma *confix striping stemmer* yang merupakan pengembangan dari algoritma *Nazief and Andriani's Stemmer*.
5. *Tokenizing*
   Tahap ini memisahkan deretan kata di dalam satu kalimat, paragraf, atau halaman menjadi token. Tiga jenis *n*-gram akan diterapkan pada tahap ini.

2.6.1 *N*-gram

Bahasa tidak terbentuk dari kata-kata individu melainkan terdiri dari urutan kata individu dan frase dua, tiga, atau lebih yang kemudian lebih dikenal sebagai *n*-gram (Pujadayanti, Fauzi, & Sari, 2018). *N*-gram merupakan metode untuk memeriksa *n* kata berkelanjutan dari urutan teks tertentu. Dalam analisis sentimen, *n*-gram dapat membantu menganalisis sentimen teks atau dokumen berdasarkan ukuran *n* yang digunakan (Tripathy, Agrawal, & Rath, 2016). *Tokenizing* dengan menggunakan *n*-gram akan memisahkan deretan kata di dalam satu kalimat, paragraf, atau halaman menjadi token unigram, bigram, dan trigram. Tabel 2.1 adalah ilustrasi penerapan *n*-gram pada sebuah kalimat "aku suka banget produk cleanser ini."

Tabel 2.1 Ilustrasi Penggunaan *n*-gram

Unigram (n = 1)	'aku', 'suka', 'banget', 'produk', 'cleanser', 'ini'
Bigram (n = 2)	'aku suka', 'suka banget', 'banget produk', 'produk cleanser', 'cleanser ini'
Trigram (n = 3)	'aku suka banget', 'suka banget produk', 'banget produk cleanser', 'produk cleanser ini'

2.6.2 *Confix Stripping Stemmer*

Library Sastrawi di *Python* dapat digunakan untuk *stemming* pada *pre processing* dokumen berupa teks berbahasa Indonesia karena merupakan *stemmer* yang berdasarkan algoritma yang dikembangkan oleh Nazief dan Adriani pada tahun 1996. Metode *stemming* untuk teks Bahasa Indonesia yang dikembangkan oleh Nazief dan Adriani disebut *Confix Stripping* (CS) (Arifin, Mahendra, & Ciptaningtyas, 2009). Berikut merupakan algortima *Confix Stripping Stemmer* (Adriani, dkk, 2007).

1. Di awal pemrosesan, pada setiap langkah dilakukan pemeriksaan kata yang merujuk kamus kata dasar. Jika kata tersebut ditemukan maka kata tersebut dianggap sebagai kata dasar sehingga seluruh proses dihentikan.
2. Menghilangkan akhiran *inflectional particles* {"-kah", "-lah", "-tah", "-pun"} kemudian menghilangkan pula akhiran *inflectional possessive pronoun* seperti {"-ku", "-mu", atau "-nya"}. Misal terdapat kata "bajumulah" maka pertama kata tersebut dipotong menjadi "bajumu" kemudian menjadi "baju", dimana kata tersebut sudah merupakan kata dasar sehingga proses dihentikan.
3. Menghilangkan *derivational suffixes* seperti {"-i", "-kan", dan "-an"}. Contohnya kata "membelikan" akan dipotong menjadi "membeli" dengan menghilangkan akhiran "-kan". Karena kata tersebut belum menjadi kata dasar maka proses berlanjut selanjutnya untuk menghilangkan awalan (*prefix*).
4. Menghilangkan *derivational prefixes* {"be-", "di-", "ke-", "me-", "pe-", "se-", dan "te-"}.

a. Proses dihentikan jika:
   - Awalan yang teridentifikasi merupakan pasangan dari akhiran yang telah dihilangkan pada langkah 3.
   - Awalan yang teridentifikasi saat ini sama dengan awalan yang telah dihilangkan sebelumnya.
   - Tiga awalan telah dihilangkan.

b. Mengidentifikasi tipe awalan kemudian dihilangkan. Awalan dibedakan menjadi dua tipe seperti berikut.

- Sederhana. Terdiri atas{"di-", "ke-", "se-"} yang dapat dihilangkan langsung.
- Kompleks. Terdiri atas {"be-", "te-", "me-", atau "pe-"} yang dapat bermorfologi sesuai kata dasar yang mengikutinya (mengubah bentuk asli dari kata dasar). Misal awalan "me-" dapat menjadi "mem-", "men-", "meny-", atau "meng-" bergantung pada huruf yang berada di awal kata dasar. Pada langkah sebelumnya, kata "membelikan" telah dipotong menjadi "membeli". Pada langkah ini akan dipotong awalan "mem-" untuk mendapatkan "beli" dimana kata tersebut sudah merupakan kata dasar sehingga proses dihentikan.

c. Apabila kata tidak dapat ditemukan pada kamus kata dasar maka mengulangi langkah 4. Jika ditemukan maka proses dihentikan.

5. Apabila kata dasar masih belum ditemukan hingga langkah keempat maka perlu dilakukan proses *recoding* yaitu dengan menambah atau mengganti huruf awal kata yang terpenggal.

6. Apabila semua langkah tidak berhasil maka algoritma akan kembali pada kata yang belum dilakukan proses *stemming* dimana kata tersebut akan dianggap menjadi kata dasar.

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

Pembobotan kata (*term weighting*) merupakan proses pembobotan kata dimana pada dasarnya dilakukan dengan menghitung frekuensi kemunculan *term* pada dokumen. *Term Frequency* (TF) merupakan petunjuk sejauh mana *term* tersebut mewakili isi dokumen. Sehingga semakin banyak kemunculan suatu *term* (kata) dalam dokumen maka akan semakin besar nilai kesesuaian yang diberikan. Sementara *Inverse Document Frequency* (IDF) adalah pembobotan dengan memperhitungkan faktor kebalikan dari frekuensi dokumen yang mengandung suatu *term* (kata). Oleh karena itu, pada metode TF-IDF menurut Manning, Raghavan, & Schütze (2009) perhitungan bobot *term t* pada dokumen ke-*d* dapat digambarkan melalui persamaan (2.1) dengan mengalikan nilai *Term Frequency* ($k_{t,d}$) dan *Inverse Document Frequency* (l_t).

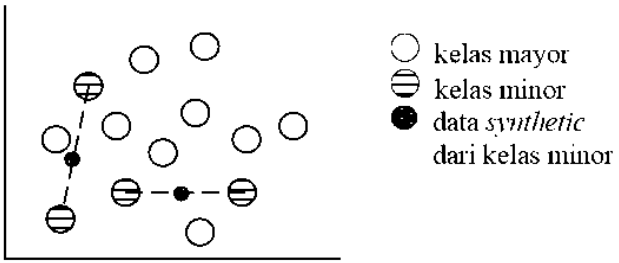
$$w_{t,d} = k_{t,d} \times l_t \quad \text{dengan} \quad l_t = \log\left(\frac{N}{m_t}\right), \qquad (2.1)$$

keterangan,

$w_{t,d}$ = bobot dari kata (*term*) ke-*t* pada *tweet* ke-*d*
$k_{t,d}$ = frekuensi kemunculan kata (*term*) ke-*t* pada *tweet* ke-*d*
N = jumlah seluruh *tweet*
$m_{t,d}$ = jumlah *tweet* yang mengandung kata (*term*) ke-*t*

2.8 *Synthetic Minority Over-sampling Technique* (SMOTE)

Dataset dikatakan tidak seimbang jika kategori klasifikasi tidak direpresentasikan secara merata. Dalam kehidupan nyata, seringkali sebagian besar kumpulan data berupa hal yang 'normal' dan hanya sebagian kecil berupa hal yang 'abnormal'. Kinerja algoritma *machine learning* dievaluasi menggunakan akurasi. Namun hal tersebut tidak tepat diterapkan ketika data tidak seimbang. *Synthetic Minority Over-sampling Technique* (SMOTE) bekerja dengan membuat replikasi dari data minoritas (kelas data dengan jumlah yang lebih sedikit) untuk menyeimbangkan dataset dengan cara *resampling* pada kelas minoritas. Replikasi tersebut dikenal dengan *synthetic data*. Data sintesis tersebut dibuat berdasarkan *k-nearest neighbor*, yaitu ketetanggan terdekat data sebanyak *k* untuk setiap data di kelas minoritas. Setelah itu dibuat data sintetis sebanyak persentase duplikasi yang diinginkan antara data minoritas dan *k-nearest neighbor* yang dipilih secara acak (Chawla, dkk, 2002). Ilustrasi metode SMOTE dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi Metode SMOTE
(Sumber: Sastrawan, Baizal, & Bijaksana, 2010)

Dari Gambar 2.3, misal diberikan data dengan *p* variabel yaitu $\mathbf{x}^T = \left[x_1, x_2, ..., x_p \right]$ dan $\mathbf{z}^T = \left[z_1, z_2, ..., z_p \right]$ maka jarak *Euclidean* secara umum dapat dilihat melalui persamaan (2.2).

$$d(\mathbf{x}, \mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + ... + (x_p - z_p)^2} \quad . \quad (2.2)$$

Sehingga pembangkitan data *synthetic* dapat dilakukan dengan rumus pada persamaan (2.3).

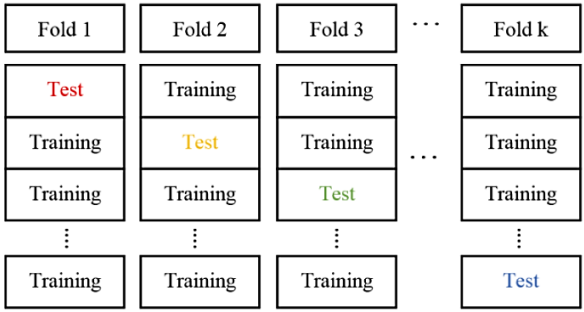
$$x_{syn} = x_i + (x_{knn} - x_i) \times \gamma \ , \quad (2.3)$$

keterangan,

x_{syn} = data hasil replikasi
x_i = data ke-*i* dari kelas minor
x_{knn} = data dari kelas minor yang memiliki jarak terdekat dari x_i
γ = bilangan random antara 0 hingga 1

2.9 *K-fold Cross Validation*

K-fold cross validation merupakan metode yang digunakan untuk membagi atau mempartisi data menjadi data latih (*training data*) dan data uji (*testing data*). Dalam *k-fold cross validation*, data awal akan secara acak dipartisi menjadi *k* lipatan, D_1, D_2,..., D_k, dengan masing-masing berukuran kurang lebih sama.



Gambar 2.4 Ilustrasi *k-fold Cross Validation*
(Sumber: Mayasari, 2018)

Training dan *testing* dilakukan sebanyak *k* kali. Dalam iterasi *i*, partisi D_i dijadikan sebagai data *testing* sementara partisi sisanya secara kolektif digunakan untuk melatih model. Misal, untuk iterasi pertama, himpunan bagian D_2, D_3,..., D_k secara kolektif ber-

fungsi sebagai data *training* untuk mendapat model pertama kemudian akan diuji pada D_1, dan begitu seterusnya (Han, Kamber, & Pei, 2006). Gambar 2.4 merupakan ilustrasi *k-fold cross validation*.

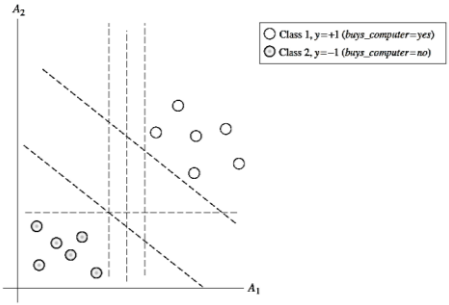
2.10 *Support Vector Machine*

Support Vector Machine merupakan metode pembelajaran *supervised* yang pertama kali diperkenalkan pada tahun 1995 oleh Vapnik. Metode ini sangat berhasil dalam melakukan prediksi baik pada klasifikasi maupun regresi. Klasifikasi dilakukan dengan mencari *hyperplane* atau bidang pembatas yang memisahkan antara suatu kelas dengan kelas lain, yang dalam kasus ini garis tersebut berperan memisahkan *tweet* bersentimen positif dengan *tweet* yang bersentimen negatif. SVM melakukan pencarian nilai *hyperplane* dengan menggunakan *support vector* dan nilai margin (Han, Kamber, & Pei, 2006).

Metode *Support Vector Machine* (SVM) merupakan teori pembelajaran statistik dan dapat memberikan hasil yang lebih baik dari metode lain (Prasetyo, 2012). SVM dapat bekerja dengan baik pada data berdimensi tinggi. Selain itu, SVM menggunakan teknik kernel dan hanya sejumlah data yang terpilih yang berkontribusi membangun model klasifikasi. Hal tersebut menjadi kelebihan SVM, karena tidak semua data latih (*training data*) akan dilihat untuk dilibatkan dalam setiap iterasi pelatihannya.

2.10.1 SVM *Linear Separable*

Melalui Gambar 2.5 dapat dilihat bahwa data dengan dua dimensi terpisah secara linier karena garis lurus dapat tergambar untuk memisahkan semua *tuple* untuk kelas +1 dari semua *tuple* milik kelas -1. Terdapat banyak garis pemisah (*hyperplane*) yang dapat tergambar. *Hyperplane* pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*.



Gambar 2.5 Contoh Pola Data Terpisah Secara Linier
(Sumber: Han, Kamber, & Pei, 2006)

Dalam penelitian ini, misalkan setiap *tweet* ke-*i* memiliki sepasang *j* prediktor berupa bobot kata $\mathbf{x}_i^T = \left[x_{i1}, x_{i2}, ..., x_{ij} \right]$ dengan *i* = 1, 2,..., *n* dimana *n* merupakan banyak *tweet*, berpasangan dengan $y_i \in \{-1, +1\}$ maka data kumpulan *tweet* dapat dinyatakan dalam persamaan (2.4).

$$D_n = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \in X \times \{-1, +1\} \ . \qquad (2.4)$$

Jika $\mathbf{x}_i$ adalah anggota kelas (+1) maka diberi label pada target dengan y_i = +1 dan jika $\mathbf{x}_i$ adalah anggota kelas (-1) maka diberi label y_i = -1 sehingga data *tweet* yang diberikan berupa pasangan $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ merupakan himpunan dari *training data* dari dua kelas yang akan diklasifikasikan menggunakan metode *Support Vector Machine*.

Separating hyperplane atau bidang pemisah yang membagi ruang menjadi dua daerah digambarkan melalui persamaan (2.5).

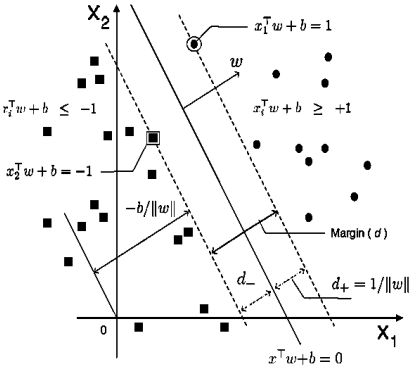
$$f(x) = \mathbf{x}^T \mathbf{w} + b = 0 \ , \qquad (2.5)$$

dengan **w** adalah vektor bobot yang berukuran $j \times 1$ dan *b* adalah bias atau posisi bidang relatif terhadap pusat koordinat yang berupa skalar.

Hyperplane dapat dikatakan linier jika merupakan fungsi linier dalam input $\mathbf{x}_i$. *Support vector* merupakan data yang berada

pada *margin*. Sehingga dalam kasus data *linear separable*, fungsi pemisah untuk kedua kelas digambarkan pada persamaan (2.6).

$$\mathbf{x}_i^{\,T}\mathbf{w}+b\geq+1 \quad \text{untuk} \quad y_i=+1\,,$$
$$\mathbf{x}_i^{\,T}\mathbf{w}+b\leq-1 \quad \text{untuk} \quad y_i=-1. \qquad (2.6)$$



Gambar 2.6 Ilustrasi *Hyperplane* untuk Data *Linear Separable* (Sumber: Hardle, Prastyo, & Hafner, 2014)

Berdasarkan Gambar 2.6, bidang pembatas $\mathbf{x}_i^{\,T}\mathbf{w}+b\geq+1$ memiliki bobot **w** dan jarak tegak lurus dari titik pusat koordinat adalah $\frac{|1-b|}{\|\mathbf{w}\|}$, sedangkan bidang pembatas $\mathbf{x}_i^{\,T}\mathbf{w}+b\leq-1$ memiliki jarak tegak lurus dari titik pusat koordinat sebesar $\frac{|-1-b|}{\|\mathbf{w}\|}$. Sehingga besar jarak antara *margin* dan bidang pemisah (*separating hyperplane*) adalah $\frac{1}{\|\mathbf{w}\|}$ dan nilai maksimum *margin* antara bidang pembatas adalah $\frac{1-b-(-1-b)}{\|\mathbf{w}\|}=\frac{2}{\|\mathbf{w}\|}$. Dengan demikian maka untuk mencari *hyperplane* yang optimal adalah dengan memaksimumkan nilai $\frac{2}{\|\mathbf{w}\|}$ atau ekuivalen dengan meminimumkan nilai

$\frac{1}{2}\|\mathbf{w}\|^2$. Kedua bidang pembatas yang ditunjukkan pada persamaan (2.6) dapat digabung menjadi persamaan (2.7).

$$y_i\left(\mathbf{x}_i^T\mathbf{w}+b\right)-1\geq 0 \quad ;\ i=1,...,n. \tag{2.7}$$

Sehingga untuk mencari *hyperplane* optimal dari kasus *linear separable* adalah dengan meminimumkan nilai $\frac{1}{2}\|\mathbf{w}\|^2$ serta konstrain yang ditunjukkan pada persamaan (2.7). Optimasi tersebut dapat diselesaikan dengan *Lagrange Multiplier* dengan rumus yang ditunjukkan persamaan (2.8).

$$L\left(\mathbf{w},b,\boldsymbol{\alpha}\right)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{n}\alpha_i\left[y_i\left(\mathbf{x}_i^T\mathbf{w}+b\right)-1\right], \tag{2.8}$$

dimana $\alpha_i \geq 0$ yang merupakan nilai koefisien *Lagrange*. Nilai optimal dari formula *Lagrangian* untuk *primal problem* dalam kasus ini adalah dengan meminimumkan *L* terhadap **w** dan *b*, sementara untuk *dual problem* adalah dengan memaksimalkan *L* terhadap **α**, sehingga dapat diringkas menjadi persamaan (2.9).

$$\max_{\boldsymbol{\alpha}} L_D=\max_{\boldsymbol{\alpha}}\left(\min_{\mathbf{w},b} L\left(\mathbf{w},b,\boldsymbol{\alpha}\right)\right). \tag{2.9}$$

Penyelesaian *primal problem* atau meminimumkan *L* terhadap **w** dan *b* diperoleh hasil seperti yang ditunjukkan pada persamaan (2.10) dan (2.11).

$$\frac{\partial L_P}{\partial \mathbf{w}}=\frac{\partial L\left(\mathbf{w},b,\boldsymbol{\alpha}\right)}{\partial \mathbf{w}}=\mathbf{w}-\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i=0, \tag{2.10}$$

$$\frac{\partial L_P}{\partial b}=\frac{\partial L\left(\mathbf{w},b,\boldsymbol{\alpha}\right)}{\partial b}=\sum_{i=1}^{n}\alpha_i y_i=0. \tag{2.11}$$

Melalui persamaan (2.10) diperoleh bahwa $\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i$, maka persamaan (2.8) menjadi suatu persamaan yang ditunjukkan oleh persamaan (2.12).

$$\begin{aligned} \frac{1}{2}\|\mathbf{w}\|^2 &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j \ . \\ \sum_{i=1}^{n}\alpha_i\left[y_i\left(\mathbf{x}_i{}^T\mathbf{w}+b\right)-1\right] &= \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i{}^T\sum_{j=1}^{n}\alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{n}\alpha_i \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j - \sum_{i=1}^{n}\alpha_i \ . \end{aligned} \quad (2.12)$$

Sehingga persamaan (2.9) dapat dilanjutkan dengan mensubstitusikan persamaan (2.12) menjadi persamaan (2.13) sebagai penyelesaian *dual problem* yaitu memaksimumkan *L* terhadap **α**.

$$\max_{\boldsymbol{\alpha}} L_D = \max_{\boldsymbol{\alpha}}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j\right), \quad (2.13)$$

dengan $\sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \geq 0$.

Dengan demikian akan diperoleh nilai α_i yang dapat digunakan untuk menghitung nilai **w**. Setiap data latih yang memiliki nilai $\alpha_i > 0$ disebut sebagai *support vector*, sementara data latih sisanya yang memiliki nilai $\alpha_i = 0$ disebut sebagai *non-support vector*. Selanjutnya dapat dilakukan klasifikasi dengan aturan klasifikasi yang ditunjukkan pada persamaan (2.14).

$$g(x) = sign\left(\mathbf{x}_i{}^T\mathbf{w}+b\right), \quad (2.14)$$

dimana $\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$ dan $b = \frac{1}{2}\left(x_{+1}+x_{-1}\right)\mathbf{w}$ dengan x_{+1} dan x_{-1} merupakan dua *support vector* yang mengikuti kelas berbeda untuk persamaan $y\left(\mathbf{x}^T\mathbf{w}+b\right)=1$, sehingga melalui persamaan (2.15), nilai dari fungsi klasifikasi dapat dihitung (Hardle, Prastyo, & Hafner, 2014).

$$f(x) = \mathbf{x}^T\mathbf{w}+b \ . \quad (2.15)$$

2.10.2 SVM *Linear Non-Separable*

Kasus data yang tidak terpisah secara linier diasumsikan bahwa kelas pada *input space* tidak dapat terpisah secara sempurna maka *constraint* pada persamaan (2.6) tidak dapat terpenuhi. Untuk

mengatasi masalah ini maka pada persamaan (2.6) tersebut akan ditambah dengan variabel *slack*, ξ_i yang mewakili pelanggaran dari *strict separation* yang memungkinkan suatu titik berada dalam *margin error*, $0 \le \xi_i \le 1$, atau menjadi salah diklasifikasikan, $\xi_i > 1$. Oleh karena itu fungsi pemisah untuk kasus data *linear non-separable* menjadi seperti pada persamaan (2.16).

$$\begin{aligned} \mathbf{x}_i^T\mathbf{w}+b &\ge +1-\xi_i \text{ untuk } y_i = +1\,, \\ \mathbf{x}_i^T\mathbf{w}+b &\le -(1-\xi_i) \text{untuk } y_i = -1\,. \end{aligned} \tag{2.16}$$

Pencarian *hyperplane* terbaik dengan penambahan variabel *slack* ξ_i sering disebut sebagai *soft margin hyperplane*. Kedua fungsi pemisah yang ditunjukkan pada persamaan (2.16) dapat digabung menjadi persamaan (2.17) berikut.

$$y_i\left(\mathbf{x}_i^T\mathbf{w}+b\right) \ge 1-\xi_i \text{ dengan } \xi_i \ge 0\,;\ i=1,...,n. \tag{2.17}$$

Dengan demikian, maka fungsi tujuan mencari *hyperplane* terbaik berubah menjadi seperti persamaan (2.18).

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\left[\sum_{i=1}^{n}\xi_i\right] \text{ dengan } \xi_i \ge 0;\ C>0\,, \tag{2.18}$$

dimana C (*cost*) adalah parameter penentu besar pelanggaran akibat kesalahan klasifikasi data dan nilainya ditentukan oleh pengguna. Apabila nilai C besar maka *margin* akan menjadi lebih kecil yang mengindikasikan bahwa tingkat kesalahan akan menjadi lebih kecil.

Sehingga optimasi untuk mencari *hyperplane* optimal dari kasus *linear non-separable* ini dapat diselesaikan dengan *Lagrange Multiplier* dengan rumus yang ditunjukkan persamaan (2.19).

$$\begin{aligned} L(\mathbf{w},b,\xi,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left[y_i\left(\mathbf{x}_i^T\mathbf{w}+b\right)-1+\xi_i\right] \\ -\sum_{i=1}^{n}\mu_i\xi_i, \end{aligned} \tag{2.19}$$

dimana $\alpha_i \ge 0$ dan $\mu_i \ge 0$ yang merupakan *Lagrange Multiplier*. Nilai optimal dari formula *Lagrangian* untuk *primal problem* dalam kasus ini adalah dengan meminimumkan L terhadap $\mathbf{w}$, b,

dan ξ_i, sementara untuk *dual problem* adalah dengan memaksimalkan L terhadap $\boldsymbol{\alpha}$, sehingga dapat diringkas menjadi persamaan (2.20).

$$\max_{\boldsymbol{\alpha}} L_D = \max_{\boldsymbol{\alpha}} \left(\min_{\mathbf{w},b,\xi} L(\mathbf{w},b,\xi,\boldsymbol{\alpha}) \right) . \tag{2.20}$$

Penyelesaian *primal problem* atau meminimumkan L terhadap $\mathbf{w}$, b, dan ξ_i, diperoleh hasil seperti yang ditunjukkan pada persamaan (2.21), (2.22), serta (2.23).

$$\frac{\partial L_P}{\partial \mathbf{w}} = \frac{\partial L(\mathbf{w},b,\xi,\boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 , \tag{2.21}$$

$$\frac{\partial L_P}{\partial b} = \frac{\partial L(\mathbf{w},b,\xi,\boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 , \tag{2.22}$$

$$\frac{\partial L_P}{\partial \xi_i} = \frac{\partial L(\mathbf{w},b,\xi,\boldsymbol{\alpha})}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 , \tag{2.23}$$

dengan kondisi $\alpha_i \geq 0,\ \mu_i \geq 0,\ \alpha_i \left[y_i \left(\mathbf{x}_i^{\,T} \mathbf{w} + b \right) - 1 + \xi_i \right] = 0$ dan $\mu_i \xi_i = 0$.

Dari persamaan (2.22) diperoleh bahwa $\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$, maka persamaan (2.19) menjadi suatu persamaan yang ditunjukkan oleh persamaan (2.24).

$$\begin{aligned} L_D(\boldsymbol{\alpha}) &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\,T} \mathbf{x}_j - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^{\,T} \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j \\ &\quad + C \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \mu_i \xi_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\,T} \mathbf{x}_j \\ &\quad + \sum_{i=1}^{n} \xi_i \left(C - \alpha_i - \mu_i \right) . \end{aligned} \tag{2.24}$$

Berdasarkan persamaan (2.23) maka $\sum_{i=1}^{n} \xi_i \left(C - \alpha_i - \mu_i\right) = 0$ maka persamaan (2.25) sebagai penyelesaian *dual problem* yaitu memaksimumkan *L* terhadap **α**.

$$\max_{\boldsymbol{\alpha}} L_D = \max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right), \quad (2.26)$$

dengan $\sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C$.

2.10.3 *Kernel Trick* dan *Non-Linear Separable*

Pada umumnya masalah dalam domain dunia nyata (*real world problem*) jarang yang bersifat *linear separable* dan kebanyakan bersifat *non linear*. Untuk menyelesaikan masalah *non linear*, SVM dimodifikasi dengan memasukkan fungsi *Kernel*. *Kernel trick* merupakan perhitungan *scalar product* melalui sebuah fungsi *kernel*. Fungsi dari *kernel* digambarkan melalui persamaan (2.27).

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \varphi\left(\mathbf{x}_i\right)^T \varphi\left(\mathbf{x}_j\right) . \quad (2.27)$$

Sehingga pencarian *hyperplane* optimum pada kasus ini dapat dilakukan menggunakan persamaan (2.28).

$$\max_{\boldsymbol{\alpha}} L_D = \max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \right), \quad (2.28)$$

dengan $\sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C$.

Kernel trick memberikan kemudahan karena pada proses pembelajaran *Support Vector Machine*, untuk menentukan *support vector* maka pengguna hanya perlu mengetahui fungsi *kernel* yang dipakai tanpa mengetahui wujud dari fungsi non-linier. *Kernel Radial Basis Function* mampu memberikan hasil terbaik pada proses klasifikasi untuk data yang tidak bisa secara linier dipisahkan (Kurniasari, 2018). Beberapa *kernel* yang umum dipakai pada SVM ditunjukan pada Tabel 2.2.

Dalam penelitian ini, *kernel trick* yang digunakan adalah Linier dan *Radial Basis Function* (RBF). *Kernel* linier memiliki

parameter *C* (*Cost*) sementara *kernel* RBF memiliki parameter *C* dan γ (*gamma*). Sehingga aturan klasifikasi pada persamaan (2.14) berubah menjadi seperti persamaan (2.29) jika digunakan *kernel trick*.

$$g(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right), \tag{2.29}$$

dengan *i* menunjukkan indeks *support vector* serta fungsi *kernel* $K(\mathbf{x},\mathbf{x}_i)$ dapat dimasukkan masing-masing fungsi *kernel* untuk linier maupun RBF pada Tabel 2.2.

Tabel 2.2 *Kernel* pada *Support Vector Machine*

Kernel	**Rumus**
Linier	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
Polinomial	$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\delta \mathbf{x}_i^T \mathbf{x}_j + r\right)^p, \delta > 0$
Radial Basis Function (RBF)	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right), \gamma > 0$
Sigmoid	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\delta \mathbf{x}_i^T \mathbf{x}_j + r\right)$

2.11 Evaluasi Model Klasifikasi

Evaluasi model klasifikasi dapat menggunakan akurasi klasifikasi. Skor evaluasi digunakan untuk mengukur kinerja model yang dihasilkan oleh metode klasifikasi. Salah satu ukuran untuk mengukur akurasi klasifikasi adalah menggunakan *confussion matrix*. *Confussion matrix* adalah sebuah tabel yang menyatakan jumlah data *testing* yang benar diklasifikasikan dan jumlah data *testing* yang salah diklasifikasikan. Tabel 2.3 merupakan bentuk *confussion matrix*.

Tabel 2.3 *Confusion Matrix*

Kelas Sebenarnya	**Kelas Prediksi**	
	Negatif (0)	Positif (1)
Negatif (0)	TN	FP
Positif (1)	FN	TP

True Negative (TN) adalah jumlah data dari kelas 0 yang benar diklasifikasikan dan diklasifikasikan sebagai kelas 0. *True Positive* (TP) adalah jumlah dokumen dari kelas 1 yang benar diklasifikasikan ke kelas 1. *False Negative* (FN) adalah jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0. Dan *False POsitive* (FP) yaitu jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1. Perhitungan kebaikan model klasifikasi dapat menggunakan rumus (2.30).

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} . \quad (2.30)$$

Akurasi prediksi pada klasifikasi bisa jadi tidak tepat digunakan jika data tidak seimbang, sehingga dibutuhkan indikator lain sebagai evaluator ketepatan klasifikasi yang dihasilkan seperti pada persamaan (2.31) dan (2.32).

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}} , \quad (2.31)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} . \quad (2.32)$$

Kurva *Receiver Operating Characteristics* (ROC) merupakan teknik standar dalam meringkas kinerja *classifier* pada rentang tingkat kesalahan antara *true positive* dan *false positive.* Pada kurva ROC, sumbu *X* menggambarkan presentase *false positive* sementara sumbu *Y* menggambarkan presentase *true positive*, yang dapat diperoleh melalui rumus pada persamaan (2.33). Titik ideal pada kurva ROC berada di (0,100), dimana semua kelas positif telah tepat diklasifikasikan dan tidak ada kelas negatif yang salah diklasifikasikan di kelas positif (Chawla, 2005).

$$\%\text{FP} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad \text{dan} \quad \%\text{TP} = \frac{\text{TP}}{\text{TP} + \text{FN}} . \quad (2.33)$$

Area di bawah kurva ROC atau biasa disebut sebagai AUC (*Area Under Curve*) adalah indikator yang menunjukkan kinerja kurva ROC dengan meringkas performa *classifier* menjadi satu metrik tunggal. Dalam penerapannya nilai AUC bervariasi antara 0,5 hingga 1 dimana jika nilai AUC semakin mendekati 1 maka

akan semakin bagus (Bekkar, Djemaa, & Alitouche, 2013). Perhitungan AUC digambarkan melalui rumus pada persamaan (2.34).

$$\text{AUC} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right). \tag{2.34}$$

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data yang diperoleh langsung menggunakan *Twitter API* (*Application Programming Interface*). Data yang dikumpulkan berupa kumpulan *tweet* atau kicauan pengguna *Twitter* yang diajukan kepada akun *Twitter* resmi milik AirAsia (@AirAsia_indo) dan LionAir (@lionairgroup) dari 21 Februari 2019 hingga 7 Mei 2019.

3.2 Variabel Penelitian

Variabel penelitian yang digunakan adalah data bobot kata dan kelas sentimen seperti yang ditunjukan oleh Tabel 3.1.

Tabel 3.1 Variabel Penelitian

Variabel Penelitian	Keterangan	Skala
Y	Kelas Sentimen: 0 = Negatif 1 = Positif	Nominal
X	Bobot Kata	Rasio

3.3 Struktur Data

Tabel 3.2 adalah struktur data *tweet* yang diambil melalui bantuan *Twitter API* (*Application Programming Interface*) yang ditujukan kepada akun resmi AirAsia (@AirAsia_indo) dan LionAir (@lionairgroup).

Tabel 3.2 Struktur Data Awal

No	***Created At***	***Text***	**Sentimen**
1	*Created At*$_1$	*Text*$_1$	Sentimen$_1$
2	*Created At*$_2$	*Text*$_2$	Sentimen$_2$
.	.	.	.
.	.	.	.
n	*Created At*$_n$	*Text*$_n$	Sentimen$_n$

Kolom *Created At* pada Tabel 3.2 menunjukkan waktu *tweet* disebarkan. *Text* merupakan kalimat pada *tweet* dan kolom Sentimen menunjukan kelas sentimen dari *tweet* (positif/negatif) serta *n* merupakan banyak *tweet*.

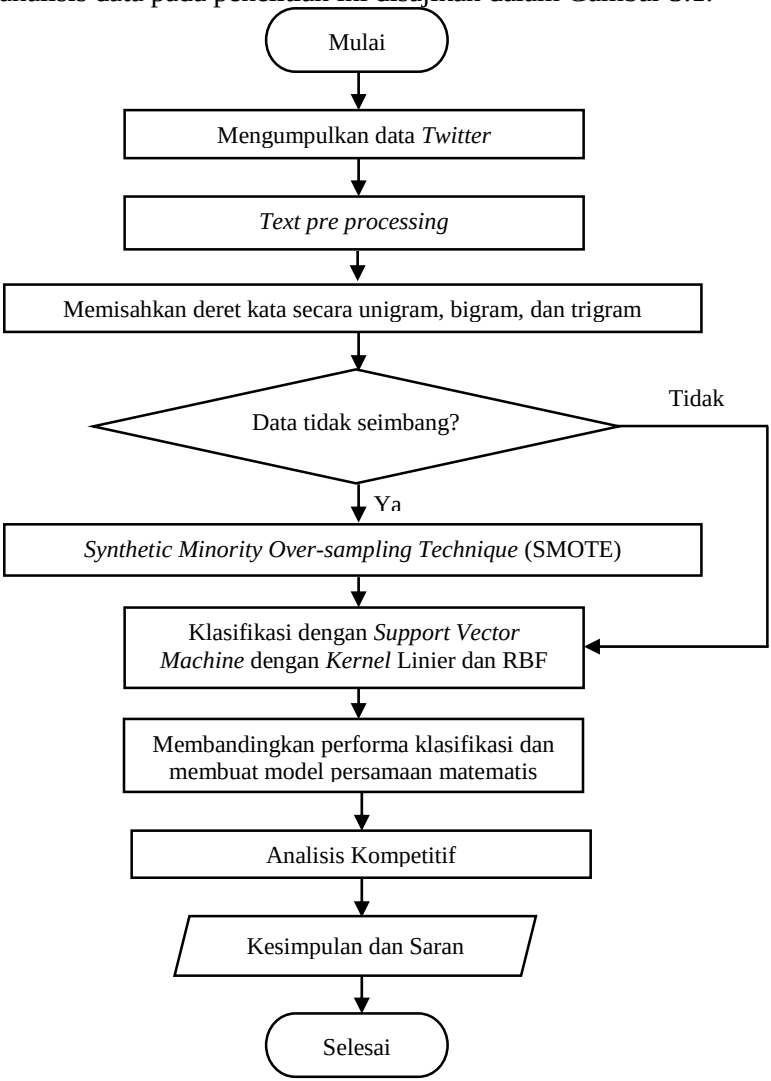
3.4 Langkah Analisis

Langkah analisis yang dilakukan dalam penelitian ini adalah sebagai berikut.

1. Mengumpulkan data berupa *tweet* yang diajukan kepada akun *Twitter* resmi milik AirAsia (@AirAsia_indo) dan LionAir (@lionairgroup) kemudian menyimpan hasil pencarian tersebut ke dalam suatu *database*.
2. Memberi label berupa sentimen (positif atau negatif) pada data *tweet* secara manual.
3. Melakukan *text pre-processing* (*cleaning*, *case folding*, penghilangan *stopwords*, *stemming*, dan *tokenizing* dengan *n*-gram).
4. Melakukan pembobotan pada *term* menggunakan TF-IDF.
5. Mempartisi data menggunakan *k-fold cross validation*.
6. Melakukan *Synthetic Minority Over-sampling Technique* (SMOTE) untuk mengatasi kasus data tidak seimbang.
7. Melakukan klasifikasi teks menggunakan metode *Support Vector Machine*.
    a. Menentukan parameter optimum *C* untuk *kernel* linier serta parameter *C* dan γ untuk *kernel* RBF.
    b. Menghitung skor evaluasi klasifikasi.
8. Menghitung dan membandingkan skor evaluasi model hasil klasifikasi *Support Vector Machine* berdasarkan setiap jenis *n*-gram yang digunakan.
9. Membuat model persamaan berdasarkan hasil klasifikasi terbaik untuk masing-masing maskapai penerbangan.
10. Menganalisis kompetitif AirAsia dan LionAir dengan mendeskripsikan karakteristik sentimen.
11. Menarik kesimpulan dan saran.

3.5 Diagram Alir

Diagram alir sebagai gambaran sederhana dari langkah analisis data pada penelitian ini disajikan dalam Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

(Halaman ini sengaja dikosongkan)

BAB IV
ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas analisis hasil pengolahan data yang telah dilakukan. Metode yang digunakan dalam penelitian ini adalah klasifikasi teks dengan *Support Vector Machine*. Data yang digunakan merupakan *tweet* yang ditujukan kepada akun *Twitter* resmi AirAsia dan LionAir. Pengumpulan data dilakukan dengan bantuan *Twitter* API. Sebelum melakukan analisis utama, data terlebih dahulu dilakukan tahap *pre processing* sebagai pembersihan data. Kemudian akan dilakukan perbandingan kebaikan hasil klasifikasi menggunakan nilai *area under curve* (AUC). Selain itu akan dilakukan pula analisis kompetitif.

4.1 Karakteristik *Tweet* yang Ditujukan Kepada AirAsia dan LionAir

Mengetahui serta memahami karakteristik dari data sebelum diolah merupakan hal yang penting, sebab dapat membantu peneliti untuk melakukan tahap selanjutnya dengan tepat. Oleh karena itu, dalam penelitian dilakukan analisis karakteristik terhadap data *tweet* yang ditujukan kepada akun *Twitter* resmi kedua maskapai.

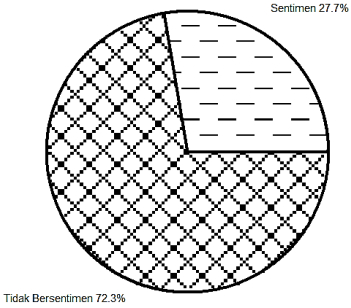
4.1.1 AirAsia

Sebanyak 5213 *tweet* dikicaukan oleh pengguna *Twitter* dengan menyebut akun resmi milik AirAsia yaitu @AirAsia_indo.

Tabel 4.1 Contoh Kata untuk Pelabelan Sentimen AirAsia

Februari		Maret		April		Mei	
Positif	**Negatif**	**Positif**	**Negatif**	**Positif**	**Negatif**	**Positif**	**Negatif**
terima	tiket	gratis	tiket	rute	pener-bangan	rute	pilot
kasih	kenapa	kursi	aplikasi	buka	tiket	buka	tiket
tiket	refund	tiket	ota	kasih	uang	dong	bayar
pener-bangan	error	ikutan	kon-sumen	terima	refund	semoga	harga
...	...	...	...	...	...	...	...
senang	lama	merapat	susah	asyik	dibatalkan	sukses	kapok
...	...	...	...	...	...	...	...
mendunia	susah	sikat	bodoh	terbaik	frustasi	mantap	rampok

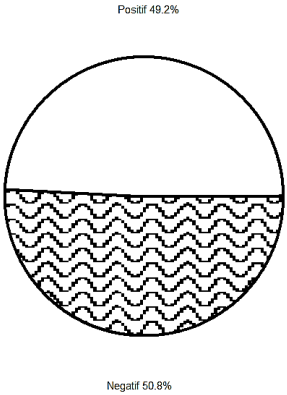
Tabel 4.1 menunjukkan beberapa kata yang digunakan untuk melabeli *tweet* dengan sentimen. Seperti yang telah dijelaskan pada Bab II bahwa konsep pelabelan sentimen secara subjektif yang dilakukan pada penelitian ini dapat dianalogikan dengan konsep analisis regresi. Jika dalam penelitian ini digunakan jenis sentimen positif/negatif yang berperan sebagai variabel respon, maka analisis regresi logistik biner merupakan analogi yang lebih tepat. Pada keempat periode bulan, kata yang sering muncul cenderung selalu berubah karena terdapat kejadian-kejadian berbeda. Misalnya pada bulan Maret, AirAsia mengadakan promo gratis kursi berupa kuis sehingga sering muncul kata 'gratis', 'kursi', hingga 'ikutan' di kelas positif. Sementara di kelas negatif, sering muncul kata 'tiket', 'aplikasi', hingga 'konsumen' dimana pada periode tersebut AirAsia mencabut perjanjian kerjasama dengan hampir seluruh *online travel agency* di Indonesia yang kemudian ramai dikeluhkan pengguna *Twitter*. Sesuai konsep analisis regresi maka seluruh kata yang termasuk di dalam kelas tertentu pada periode tertentu akan berkombinasi dengan berperan sebagai variabel prediktor sehingga sebuah *tweet* dapat dinyatakan jenis sentimennya.



Gambar 4.1 *Pie Chart Tweet* untuk AirAsia

Kategori tidak bersentimen pada Gambar 4.1 adalah untuk *tweet* yang tidak digunakan dalam penelitian ini. Termasuk di dalamnya antara lain *tweet* yang berbahasa selain bahasa Indonesia, memiliki ambiguitas, serta tidak mengandung sentimen positif maupun negatif. Dapat dilihat bahwa *tweet* bersentimen yang

digunakan dalam penelitian ini sebesar 27,7% dari seluruh data yang telah dikumpulkan.



Gambar 4.2 *Pie Chart Tweet* Bersentimen untuk AirAsia

Berdasarkan Gambar 4.2 dapat diketahui bahwa persentase *tweet* yang mengandung sentimen positif dan negatif hampir seimbang. *Tweet* positif berjumlah 734 (50,8%) dan *tweet* negatif sebanyak 711 (49,2%).

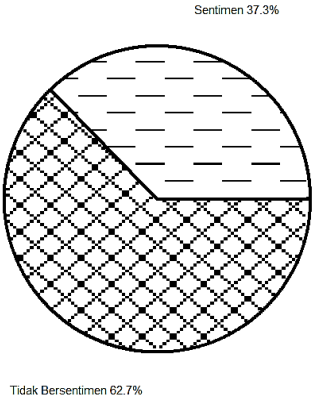
4.1.2 LionAir

Tabel 4.2 menunjukkan kata-kata yang digunakan untuk melabeli *tweet* yang ditujukan kepada LionAir dengan sentimen. Konsep pelabelan sentimen secara subjektif yang dilakukan pada penelitian ini dapat dianalogikan dengan konsep analisis regresi logistik biner karena digunakan jenis sentimen positif dan negatif yang berperan sebagai variabel respon. Pada keempat periode bulan, kata yang sering muncul cenderung selalu berubah karena terdapat kejadian-kejadian berbeda. Misalnya pada bulan Mei, ramai diperbincangkan di *Twitter* mengenai video yang menunjukkan pilot LionAir melakukan hal tidak pantas. Sehingga dapat dilihat pada Tabel 4.2 bahwa sering muncul kata ‘pilot’, ‘lion’, ‘pecat’, hingga ‘hukum’ di kelas negatif. Sesuai konsep analisis regresi maka seluruh kata yang termasuk di dalam kelas tertentu pada periode tertentu akan berkombinasi dengan berperan sebagai variabel prediktor sehingga sebuah *tweet* yang ditujukan kepada LionAir dapat diperoleh jenis sentimennya.

Tabel 4.2 Contoh Kata untuk Pelabelan Sentimen LionAir

Februari		Maret		April		Mei	
Positif	**Negatif**	**Positif**	**Negatif**	**Positif**	**Negatif**	**Positif**	**Negatif**
boar-ding	jam	lion	lion	solo	lion	dukung	pilot
pass	pesawat	pesawat	bagasi	lion	tiket	mantap	lion
untung	tiket	terima	koper	rute	maska-pai	semoga	pecat
baik	bandara	kasih	naik	terima	harga	suka	anjing
...	...	...	...	...	...	...	...
asik	avtur	semoga	hilang	pontianak	rugi	ayo	hukum
...	...	...	...	...	...	...	...
selamat	sibuk	bertahan	kampret	jogja	bobrok	gagah	anarkis

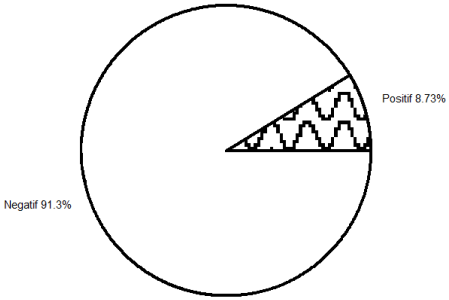
Pada Gambar 4.3 dapat dilihat bahwa 37,3% dari seluruh *tweet* yang dikumpulkan merupakan *tweet* yang bersentimen. Sementara 62,7% sisanya adalah *tweet* tidak bersentimen. *Tweet* tidak bersentimen merupakan data yang tidak digunakan dalam penelitian ini, antara lain *tweet* yang tidak berbahasa Indonesia, memiliki ambiguitas, serta tidak mengandung sentimen positif maupun negatif.



Gambar 4.3 *Pie Chart Tweet* untuk LionAir

Sementara Gambar 4.4 menunjukan persentase *tweet* yang bersentimen. *Tweet* yang mengandung sentimen negatif lebih men-

dominasi dengan persentase sebesar 91,3% atau sebanyak 1306 *tweet*. Sementara 125 *tweet* sisanya merupakan *tweet* positif.



Gambar 4.4 *Pie Chart Tweet* Bersentimen untuk LionAir

4.2 *Analisis Data*

Setelah mengetahui karakteristik data, selanjutnya dilakukan proses analisis data awal yang meliputi *pre processing* hingga pemberian bobot untuk setiap *term* sebagai variabel independen.

4.2.1 *Pre Processing*

Seperti yang telah dijelaskan pada Bab II bahwa tahap *pre processing* merupakan salah satu tahap yang penting berupa pembersihan data sehingga siap dianalisis lebih lanjut. *Pre Processing* meliputi *cleaning*, *case folding*, *stemming*, penghapusan *stopwords*, dan *tokenizing*. Penjelasan mengenai hasil dari setiap tahap dalam *pre processing* akan dijelaskan melalui Tabel 4.3 dengan menyertakan data *tweet* awal dari setiap maskapai.

Tabel 4.3 Penjelasan Tahap *Pre Processing*

Maskapai	Hasil	Keterangan
1. Data Awal		
AirAsia	@epipong @kumparan @AirAsia_indo Padahal kami butuh AirAsia ada di Traveloka atau di https://t.co/99Wh3iolVQ untuk kemudahan kita dan harga yang murah namun dengan pemesanan yang lebih mudah	
LionAir	@tijabar Belum ada penurunan tiket rute Palembang - Jakarta. Beberapa bulan lalu bisa dapat 300 ribuan. Sekarang paling murah 680 ribu. Kenaikan 100%. @lionairgroup	

Tabel 4.3 Penjelasan Tahap *Pre Processing* (lanjutan)

2. *Cleaning*		
Menghapus simbol-simbol yang tidak diperlukan.		
AirAsia	Padahal kami butuh AirAsia ada di Traveloka atau di untuk kemudahan kita dan harga yang murah namun dengan pemesanan yang lebih mudah	Menghapus simbol *mention* (*@username*), dan *link URL*.
LionAir	Belum ada penurunan tiket rute Palembang Jakarta Beberapa bulan lalu bisa dapat ribuan Sekarang paling murah ribu Kenaikan	Menghapus simbol *mention* (*@username*), angka, dan tanda baca.
3. *Case Folding*		
Mengubah semua karakter teks menjadi huruf kecil.		
AirAsia	padahal kami butuh airasia ada di traveloka atau di untuk kemudahan kita dan harga yang murah namun dengan pemesanan yang lebih mudah	
LionAir	belum ada penurunan tiket rute palembang jakarta beberapa bulan lalu bisa dapat ribuan sekarang paling murah ribu kenaikan	
4. *Stemming*		
Pemotongan kata menjadi kata dasarnya		
AirAsia	padahal kami butuh airasia ada di traveloka atau di untuk mudah kita dan harga yang murah namun dengan pesan yang lebih mudah	- ‘kemudahan’ = ‘mudah’ - ‘pemesanan’ = ‘pesan’
LionAir	belum ada turun tiket rute palembang jakarta beberapa bulan lalu bisa dapat ribu sekarang paling murah ribu naik	- ‘penurunan’ = ‘turun’ - ‘ribuan’ = ‘ribu’ - ‘kenaikan’ = ‘naik’

Tabel 4.3 Penjelasan Tahap *Pre Processing* (lanjutan)

5.	***Stopwords*** Menghapus *term* yang tidak berhubungan dengan subyek utama.		
	AirAsia	butuh airasia traveloka mudah harga murah pesan lebih mudah	Menghapus ‘padahal’, ‘kami’, ‘ada’, ‘di’, ‘atau’, ‘untuk’, ‘kita’, ‘dan’, ‘yang’, ‘namun’, ‘dengan’, ‘belum’, ‘beberapa’, ‘bulan’, ‘lalu’, ‘bisa’, ‘dapat’, ‘sekarang’, ‘paling’.
	LionAir	turun tiket rute palembang jakarta ribu murah ribu naik	
6.	***N*-gram** Memecah kalimat menjadi *n* kata berurutan.		
	AirAsia	Unigram	butuh | airasia | traveloka | mudah | harga | murah | pesan | lebih | mudah
		Bigram	butuh airasia | airasia traveloka | traveloka mudah | mudah harga | harga murah | murah pesan | pesan lebih | lebih mudah
		Trigram	butuh airasia traveloka | airasia traveloka mudah | traveloka mudah harga | mudah harga murah | harga murah pesan | murah pesan lebih | pesan lebih mudah
	LionAir	Unigram	turun | tiket | rute | palembang | jakarta | ribu | murah | ribu | naik
		Bigram	turun tiket | tiket rute | rute palembang | palembang jakarta | jakarta ribu | ribu murah | murah ribu | ribu naik
		Trigram	turun tiket rute | tiket rute palembang | rute palembang jakarta | palembang jakarta ribu | jakarta ribu murah | ribu murah ribu | murah ribu naik

Setelah memecah kalimat menjadi *n* kata berurutan dengan konsep *n*-gram, maka selanjutnya adalah menghitung frekuensi kemunculan serta bobot kata menggunakan metode TF-IDF.

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

Setelah diperoleh kata baik secara unigram, bigram, maupun trigram, maka langkah selanjutnya adalah menghitung frekuensi kemunculan masing-masing kata di setiap *tweet*. Frekuensi tersebut kemudian sering disebut sebagai TF (*Term Frequency*). Setelah diperoleh TF maka dapat dilakukan perhitungan TF-IDF (*Term Frequency-Inverse Document Fruquency*) untuk mendapat nilai bobot kata. Nilai bobot kata tersebut yang akan diolah sebagai variabel prediktor dalam klasifikasi. Tabel 4.4 menunjukkan ilustrasi perhitungan frekuensi kemunculan (TF) serta bobot (TF-IDF) kata unigram.

Tabel 4.4 Ilustrasi Perhitungan DF dan IDF Unigram

Tweet	TF						
	Aceh	**...**	**Aplikasi**	**...**	**Naik**	**...**	**Tanggap**
1	1	...	0	...	0	...	0
2	0	...	1	...	0	...	2
...	...	...	...	...	...	...	...
1444	0	...	0	...	1	...	0
1445	0	...	1	...	...		0
DF	8	...	183	...	25	...	21
IDF	$\log\left(\frac{1445}{8}\right) = 2{,}26$	...	$\log\left(\frac{1445}{183}\right) = 0{,}90$	...	$\log\left(\frac{1445}{25}\right) = 1{,}76$	...	$\log\left(\frac{1445}{21}\right) = 1{,}84$

DF *(Document Frequency*) pada Tabel 4.4 merupakan jumlah dokumen yang mengandung *term* (kata) tertentu. Sementara bobot kata (TF-IDF) didapat melalui nilai TF yang dikalikan dengan nilai IDF. Semakin besar nilai TF-IDF suatu kata mengindikasikan bahwa semakin sedikit frekuensi kemunculannya dalam *tweet*. Nilai TF-IDF tersebut yang akan diolah dalam klasifikasi teks selanjutnya.

4.3 Klasifikasi dengan *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan metode pembelajaran *suppervised* yang telah banyak digunakan dalam permasalahan klasifikasi teks. Klasifikasi dilakukan dengan mencari

hyperplane yang memisahkan antara suatu kelas dengan kelas lain. Dalam kasus ini, garis tersebut berperan memisahkan *tweet* bersentimen positif dengan *tweet* negatif. Satu keuntungan dari metode SVM adalah cukup *robust* untuk data berdimensi tinggi dan jarang membutuhkan *feature selection* karena SVM akan langsung memilih *support vector* yang diperlukan untuk klasifikasi (Allahyari, dkk., 2017).

Pada penelitian ini, data *tweet* untuk kedua maskapai penerbangan yang dianalisis menggunakan metode SVM adalah data yang telah dilakukan pembobotan menggunakan metode TF-IDF. Kemudian data akan dibagi menjadi data latih (*training data*) dan data uji (*testing data*) menggunakan *k-fold cross validation* dengan *k* sebanyak 10. SVM yang digunakan adalah *kernel* linier dan RBF (*Radial Basis Function*).

Misal dicobakan klasifikasi dengan *kernel* RBF, parameter $C = 10^2$ dan $\gamma = 2^{-5}$. Berikut adalah contoh bentuk model klasifikasi (fungsi *hyperplane*) dari data latih yang diperoleh yang ditunjukkan persamaan (4.1).

$$f(\mathbf{x}) = \sum_{i=1}^{1000}\left(100K\left(\mathbf{x},\mathbf{x}_{sv(i)}\right)+\ldots+\alpha_{1000}y_{1000}K\left(\mathbf{x},\mathbf{x}_{sv(1000)}\right)\right)+0,05\ , \tag{4.1}$$

dengan fungsi *kernel* $K\left(\mathbf{x},\mathbf{x}_{sv(i)}\right)=\exp\left(-2^{-5}\left\|\mathbf{x}-\mathbf{x}_{sv(i)}\right\|^2\right)$. Parameter *C* berperan sebagai batas atas koefisien *Lagrange* (α) karena pada optimasi diberi syarat $\sum_{i=1}^{n}\alpha_i y_i = 0,\ 0 \le \alpha_i \le C$. Sementara *i* merupakan indeks *support vector*, dan vektor $\mathbf{x}_i$ merupakan data latih yang berperan sebagai *support vector*. Sehingga untuk menguji *testing data*, dapat memasukkan data uji **x** di persamaan (4.1). Berikut pada Tabel 4.5 adalah ilustrasi perhitungan AUC serta nilai rata-ratanya dari 10 *fold* berdasarkan *confussion matrix* hasil klasifikasi terhadap data uji.

Tabel 4.5 Ilustrasi Perhitungan AUC

***Fold* ke-**	**Kelas Aktual**	**Kelas Prediksi**		**AUC**
		Negatif	Positif	
1	Negatif	117	12	$\frac{1}{2}\left(\frac{10}{10+5}+\frac{117}{117+12}\right)=0{,}7868$
	Positif	5	10	
2	Negatif	111	18	$\frac{1}{2}\left(\frac{11}{11+4}+\frac{111}{111+18}\right)=0{,}7969$
	Positif	4	11	
3	Negatif	115	15	$\frac{1}{2}\left(\frac{12}{12+1}+\frac{115}{115+15}\right)=0{,}9038$
	Positif	1	12	
4	Negatif	112	20	$\frac{1}{2}\left(\frac{7}{7+4}+\frac{112}{112+20}\right)=0{,}7424$
	Positif	4	7	
5	Negatif	116	14	$\frac{1}{2}\left(\frac{8}{8+5}+\frac{116}{116+14}\right)=0{,}7538$
	Positif	5	8	
6	Negatif	114	16	$\frac{1}{2}\left(\frac{11}{11+2}+\frac{114}{114+16}\right)=0{,}8615$
	Positif	2	11	
7	Negatif	123	13	$\frac{1}{2}\left(\frac{3}{3+4}+\frac{123}{123+13}\right)=0{,}6665$
	Positif	4	3	
8	Negatif	123	4	$\frac{1}{2}\left(\frac{11}{11+5}+\frac{123}{123+4}\right)=0{,}8280$
	Positif	5	11	
9	Negatif	111	22	$\frac{1}{2}\left(\frac{7}{7+3}+\frac{111}{111+22}\right)=0{,}7673$
	Positif	3	7	
10	Negatif	107	24	$\frac{1}{2}\left(\frac{11}{11+1}+\frac{107}{107+24}\right)=0{,}8667$
	Positif	1	11	
Rata-rata				0,7974

4.3.1 SVM *Kernel* Linier

Pada *kernel* SVM jenis linier, parameter yang digunakan adalah *C* (*Cost*). Dalam penelitian ini, paramater *C* akan dicobakan dari nilai 10^{-2} hingga 10^{4}. Kemudian akan dipilih yang memberikan nilai ketepatan klasifikasi tertinggi untuk menjadi parameter *C* terbaik (Huang, Lin, & Huang, 2007). Tabel 4.4 adalah hasil evaluasi ketepatan klasifikasi oleh metode SVM dengan *kernel* linier berupa rata-rata nilai AUC (*Area Under Curve*) dari 10 *fold*.

Dari *range* nilai parameter *C* yang dicobakan untuk setiap jenis *n*-gram, $C = 10^{0}$ menjadi parameter terbaik dimana memberikan nilai rata-rata AUC yang paling tinggi yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Nilai Rata-rata AUC SVM *Kernel* Linier untuk AirAsia

C	Unigram	Bigram	Trigram
10^{-2}	0,6019	0,5690	0,5000
10^{-1}	0,8002	0,6740	0,5945
10^{0}	0,8511	0,7473	0,6265
10^{1}	0,8240	0,7407	0,6217
10^{2}	0,8072	0,7290	0,6217
10^{3}	0,8088	0,7282	0,6217
10^{4}	0,8063	0,7282	0,6217

Pada jenis *n*-gram berupa unigram (n = 1) di Tabel 4.6, diperoleh nilai rata-rata AUC sebesar 0,8511. Sementara pada jenis bigram dan trigram berturut-turut adalah 0,7473 dan 0,6265. Dengan demikian, maka jenis unigram dengan parameter $C = 10^0$ adalah yang terbaik pada kasus klasifikasi *tweet* yang ditujukan kepada AirAsia menggunakan *kernel* linier.

Seperti yang telah diketahui di awal bahwa jumlah data kelas positif dan negatif untuk LionAir adalah *imbalance*, dimana kelas positif jauh lebih sedikit dengan persentase 8,7%. Sehingga dalam penelitian ini, digunakan metode SMOTE (*Synthetic Minority Over-sampling Technique*) untuk mengatasi hal tersebut. SMOTE bekerja dengan membangkitkan data sintetis untuk kelas minoritas, dalam hal ini adalah kelas positif, untuk menyeimbangkan jumlah data kelas mayoritas.

Tabel 4.7 Nilai Rata-rata AUC SVM *Kernel* Linier untuk LionAir

C	Unigram		Bigram		Trigram	
	Tanpa SMOTE	SMOTE	Tanpa SMOTE	SMOTE	Tanpa SMOTE	SMOTE
10^{-2}	0,5000	0,6050	0,5000	0,5477	0,5000	0,5127
10^{-1}	0,5000	0,7773	0,5000	0,5779	0,5000	0,5127
10^{0}	0,6509	0,7885	0,5791	0,7312	0,5094	0,5550
10^{1}	0,7329	0,7543	0,5926	0,7160	0,5162	0,5482
10^{2}	0,7274	0,7503	0,5926	0,7171	0,5162	0,5478
10^{3}	0,7274	0,7434	0,5926	0,7168	0,5162	0,5478
10^{4}	0,7274	0,7450	0,5926	0,7168	0,5162	0,5478

Berdasarkan Tabel 4.7 dapat dilihat bahwa nilai rata-rata AUC yang dihasilkan dengan metode SMOTE lebih tinggi dari me-

tode SVM *kernel* linier tanpa dilakukan SMOTE untuk semua jenis *n*-gram dan *range* nilai parameter *C* yang digunakan. Hal tersebut menunjukkan bahwa pada kasus ini metode SMOTE dapat meningkatkan skor ketepatan klasifikasi ketika data tidak seimbang. Seperti pada kasus untuk AirAsia, parameter *C* terbaik adalah 10^0 karena mampu memberikan nilai rata-rata AUC tertinggi. Untuk jenis unigram, rata-rata AUC yang dihasilkan sebesar 0,7885. Sementara nilai rata-rata AUC sebesar 0,7312 dan 0,5550 berturut-turut dihasilkan dari jenis bigram dan trigram. Sehingga pada SVM jenis *kernel* linier untuk kasus LionAir akan dipilih jenis *n*-gram berupa unigram dengan parameter $C = 10^0$ dengan mengaplikasikan metode SMOTE untuk mengatasi kasus *imbalanced data*.

4.3.2 SVM *Kernel* RBF (*Radial Basis Function*)

Jenis kernel lainnya yang digunakan dalam penelitian ini adalah *Radial Basis Function* (RBF) yang menggunakan parameter *C* dan γ. Seperti pada *kernel* linier, parameter *C* akan dicobakan dengan *range* nilai 10^{-2} hingga 10^4, sementara untuk γ (*gamma*) menggunakan *range* nilai 2^{-15} hingga 2^3 (Hsu, Chang, & Lin, 2016).

Tabel 4.8 Nilai Rata-rata AUC SVM *Kernel* RBF untuk AirAsia

C	Unigram ($\gamma = 2^1$)	Bigram ($\gamma = 2^{-3}$)	Trigram ($\gamma = 2^1$)
10^{-2}	0,5000	0,5000	0,5000
10^{-1}	0,6227	0,6064	0,6095
10^0	0,8536	0,7046	0,6336
10^1	0,8619	0,7810	0,6336
10^2	0,8619	0,7732	0,6336
10^3	0,8619	0,7724	0,6336
10^4	0,8619	0,7724	0,6336

Melalui Tabel 4.8 dapat dilihat bahwa parameter terbaik untuk jenis unigram adalah $C = 10^1$ dan γ sebesar 2^1 dengan rata-rata AUC sebesar 0,8619. Sementara untuk jenis bigram, nilai rata-rata AUC tertinggi, yaitu 0,7810 diberikan oleh parameter *C* dan γ berturut-turut sebesar 10^1 dan 2^{-3}. Serta parameter terbaik $C = 10^1$ dan $\gamma = 2^1$ untuk jenis trigram dengan menghasilkan nilai rata-rata AUC 0,6336. Dengan demikian, maka dengan menggunakan *kernel* RBF untuk kasus AirAsia dipilih parameter $C = 10^1$ dan $\gamma =$

2^1 dengan jenis *n*-gram berupa unigram karena menghasilkan nilai rata-rata AUC paling tinggi.

Seperti sebelumnya, pada kasus LionAir akan dicobakan pula dengan metode SMOTE untuk mengatasi *imbalanced data*. Tabel 4.9 menunjukkan bahwa nilai rata-rata AUC mengalami peningkatan jika dilakukan metode SMOTE. Pada jenis unigram, nilai rata-rata AUC tertinggi diberikan oleh parameter $C = 10^2$ dan $\gamma = 2^{-9}$ dengan metode SMOTE sebesar 0,7974 dimana nilai tersebut meningkat dibanding tanpa menggunakan SMOTE. Parameter terbaik untuk jenis bigram adalah C sebesar 10^4 dan $\gamma = 2^{-9}$ sementara nilai rata-rata AUC paling tinggi untuk jenis trigram, yaitu 0,5727, diperoleh dari parameter $C = 10^1$ dan $\gamma = 2^{-3}$. Berdasarkan hal tersebut maka pada kasus LionAir dengan *kernel* RBF dipilih parameter terbaik $C = 10^2$ dan $\gamma = 2^{-9}$ ditambah metode SMOTE untuk mengatasi kasus data tidak seimbang.

Tabel 4.9 Nilai Rata-rata AUC SVM *Kernel* RBF untuk LionAir

C	Unigram ($\gamma = 2^{-9}$)		Bigram ($\gamma = 2^{-9}$)		Trigram ($\gamma = 2^{-3}$)	
	Tanpa SMOTE	SMOTE	Tanpa SMOTE	SMOTE	Tanpa SMOTE	SMOTE
10^{-2}	0,5000	0,6050	0,5000	0,5477	0,5000	0,5674
10^{-1}	0,5000	0,6050	0,5000	0,5477	0,5000	0,5674
10^{0}	0,5000	0,6050	0,5000	0,5477	0,5000	0,5666
10^{1}	0,5000	0,6692	0,5000	0,5525	0,5166	0,5727
10^{2}	0,5542	0,7974	0,5294	0,6582	0,5162	0,5727
10^{3}	0,7256	0,7664	0,5934	0,7289	0,5162	0,5727
10^{4}	0,7285	0,7524	0,5926	0,7304	0,5162	0,5727

4.3.3 Perbandingan Hasil Kedua *Kernel* SVM

Setelah mendapatkan parameter terbaik untuk kedua *kernel*, yaitu linier dan RBF, maka selanjutnya adalah membandingkan hasil evaluasi ketepatan klasifikasi keduanya. Tabel 4.10 menunjukkan bahwa perolehan nilai rata-rata AUC dari SVM *kernel* RBF (*Radial Basis Function*) lebih tinggi dibanding *kernel* linier untuk kedua kasus. Selain itu, Tabel 4.10 menunjukkan pada kasus LionAir yang memiliki *imbalance data*, nilai rata-rata AUC klasifikasi dengan data yang telah diaplikasikan SMOTE lebih tinggi dari tanpa diaplikasikan SMOTE.

Tabel 4.10 Perbandingan Hasil Klasifikasi

		Rata-rata AUC	
		Kernel Linier	**Kernel RBF**
AirAsia	Tanpa SMOTE	0,8511 ($C = 10^0$)	0,8619 ($C = 10^1, \gamma = 2^1$)
LionAir	Tanpa SMOTE	0,6509	0,5542
	SMOTE	0,7885 ($C = 10^0$)	0,7974 ($C = 10^2, \gamma = 2^{-9}$)

Berdasarkan hal tersebut maka disarankan untuk mengklasifikasikan *tweet* bersentimen positif atau negatif dengan metode SVM *kernel* RBF dengan parameter $C = 10^1$ dan $\gamma = 2^1$ untuk AirAsia. Sementara untuk LionAir, disarankan menggunakan metode SVM *kernel* RBF dengan parameter $C = 10^2$ dan $\gamma = 2^{-9}$. Baik AirAsia maupun LionAir disarankan pula untuk menggunakan jenis *n*-gram berupa unigram berdasarkan hasil analisis yang telah dibahas sebelumnya yang menunjukkan bahwa nilai rata-rata AUC dengan jenis unigram lebih tinggi dibanding bigram dan trigram.

4.3.4 Pemodelan Hasil Klasifikasi Terbaik

Setelah diperoleh hasil klasifikasi terbaik dengan membandingkan seluruh hasil klasifikasi yang dicobakan, maka selanjutnya adalah pemodelan dengan parameter terbaik. Tabel 4.11 merupakan hasil pengolahan data untuk AirAsia menggunakan metode terbaik dengan 10-*fold cross validation*.

Tabel 4.11 Nilai Evaluasi SVM *Kernel* RBF Parameter Terbaik dengan 10-*fold CV* untuk AirAsia

***fold* ke-**	**AUC**	**Akurasi**	**Presisi**	**Recall**
1	0,8532	0,8690	0,9200	0,7541
2	0,8981	0,8966	0,9412	0,8533
3	0,8881	0,8897	0,9077	0,8551
4	0,8932	0,8897	0,9342	0,8659
5	0,7954	0,8069	0,8800	0,6667
6	0,8374	0,8345	0,8939	0,7763
7	0,8605	0,8611	0,8923	0,8169
8	0,8304	0,8264	0,9167	0,7333
9	0,8697	0,8681	0,9231	0,8108
10	0,8930	0,9028	0,9444	0,8226
Rata-rata	0,8619	0,8645	0,9154	0,7955

Berdasarkan Tabel 4.11 tersebut maka dapat diketahui bahwa pada *fold* kedua diperoleh nilai AUC tertinggi yaitu 0,8981 dengan akurasi sebesar 0,8966. Serta presisi dan *recall* berturut-turut sebesar 0,9412 dan 0,8533. Keempat skor evaluasi ketepatan klasifikasi sentimen untuk *tweet* yang ditujukan kepada akun resmi AirAsia tersebut diperoleh berdasarkan *confussion matrix* data uji yang ditampilkan pada Tabel 4.12.

Tabel 4.12 *Confusion Matrix fold* kedua SVM *Kernel* RBF Parameter Terbaik untuk AirAsia

Kelas Sebenarnya	**Kelas Prediksi**	
	Negatif	Positif
Negatif	66	4
Positif	11	64

Pada Tabel 4.12 dapat dilihat bahwa kelas negatif yang tepat diklasifikan ke kelas negatif sebanyak 66 *tweet*, sementara yang salah diklasifikasikan ke kelas positif sebanyak 4 *tweet*. *Tweet* bersentimen positif yang salah diklasifikasikan ke kelas negatif sebanyak 11 *tweet*, sementara yang tepat diklasifikasikan ke kelas positif adalah 64 *tweet*.

Sehingga dengan menggunakan metode SVM *kernel* RBF dengan parameter $C = 10^1$ dan $\gamma = 2^1$ maka fungsi *kernel* RBF yang digunakan adalah seperti yang ditunjukan pada persamaan (4.2).

$$K\left(\mathbf{x}, \mathbf{x}_{sv(i)}\right) = \exp\left(-2^1 \left\|\mathbf{x} - \mathbf{x}_{sv(i)}\right\|^2\right) . \tag{4.2}$$

Sementara fungsi *hyperplane* sebagai fungsi pemisah antar kelas positif dan negatif, diperoleh dari *training data* pada *fold* kedua yang ditunjukkan melalui persamaan (4.3).

$$f\left(\mathbf{x}\right) = \sum_{i=1}^{1114} \alpha_i y_i K\left(\mathbf{x}, \mathbf{x}_{sv(i)}\right) - 0,3765 \ , \tag{4.3}$$

dengan $\mathbf{x}$ merupakan vektor data input dan $\mathbf{x}_{sv(i)}$ adalah data *support vector* ke-*i*, dimana $i = 1,2,...,1114$. Kedua vektor memiliki ukuran (515×1) dimana 515 merupakan jumlah fitur yang digunakan. Nilai α pada persamaan (4.3) merupakan *lagrange multiplier* dari *support vector*. Sementara nilai *y* merupakan label kelas (+1 dan -

1) dimana +1 untuk kelas sentimen positif dan -1 untuk kelas sentimen negatif. Nilai -0,3765 adalah nilai bias.

Sementara itu, berdasarkan Tabel 4.13 yang merupakan hasil pengolahan data menggunakan metode terbaik dengan 10-*fold cross validation* untuk LionAir, pada *fold* ketiga dihasilkan nilai AUC paling tinggi dibanding *fold* lainnya, yaitu 0,9038. Nilai akurasi, presisi, dan *recall* masing-masing sebesar 0,8881, 0,4444, dan 0,9231. Keempat skor evaluasi ketepatan klasifikasi sentimen pada tabel 4.13 diperoleh berdasarkan *confusion matrix* data uji yang ditampilkan pada Tabel 4.14.

Tabel 4.13 Nilai Evaluasi SVM *Kernel* RBF Parameter Terbaik dengan 10-*fold CV* untuk LionAir

***fold* ke-**	**AUC**	**Akurasi**	**Presisi**	**Recall**
1	0,7868	0,8819	0,4545	0,6667
2	0,7969	0,8472	0,3793	0,7333
3	0,9038	0,8881	0,4444	0,9231
4	0,7424	0,8322	0,2593	0,6364
5	0,7538	0,8671	0,3636	0,6154
6	0,8615	0,8741	0,4074	0,8462
7	0,6665	0,8811	0,1875	0,4286
8	0,8280	0,9371	0,7333	0,6875
9	0,7673	0,8252	0,2414	0,7000
10	0,8667	0,8252	0,3143	0,9167
Rata-rata	0,7974	0,8659	0,3785	0,7154

Tabel 4.14 *Confusion Matrix fold* kedua SVM *Kernel* RBF Parameter Terbaik untuk LionAir

Kelas Sebenarnya	**Kelas Prediksi**	
	Negatif	Positif
Negatif	115	15
Positif	1	12

Dapat dilihat melalui Tabel 4.14, sebanyak 15 *tweet* negatif salah diklasifikan ke kelas positif. *Tweet* dengan kelas positif yang tepat diklasifikasikan ke kelasnya sebanyak 12 *tweet*, sementara hanya satu *tweet* salah diklasifikasikan ke kelas negatif. Sehingga dengan menggunakan metode SVM *kernel* RBF dengan parameter $C = 10^2$ dan γ sebesar 2^{-9} serta data terlebih dahulu diaplikasikan metode SMOTE, maka fungsi *kernel* RBF yang digunakan adalah

seperti yang ditunjukan pada persamaan (4.4). Serta fungsi *hyper-plane* yang diperoleh dari *training data* pada *fold* ketiga yang ditunjukkan melalui persamaan (4.5).

$$K\left(\mathbf{x},\mathbf{x}_{sv(i)}\right)=\exp\left(-2^{-9}\left\|\mathbf{x}-\mathbf{x}_{sv(i)}\right\|^2\right)=\exp\left(-0,001953125\left\|\mathbf{x}-\mathbf{x}_{sv(i)}\right\|^2\right). \quad (4.4)$$

$$f\left(\mathbf{x}\right)=\sum_{i=1}^{1355}\alpha_i y_i K\left(\mathbf{x},\mathbf{x}_{sv(i)}\right)-81,2941\ , \quad (4.5)$$

Data *tweet* yang ditujukan kepada akun *Twitter* resmi Lion-Air di masa mendatang dapat diklasifikasi dengan memasukkan nilai **x** sebagai input sesuai pada persamaan (4.5). Vektor $\mathbf{x}_{sv(i)}$ merupakan data *support vector* ke-*i* yang akan berjalan mulai dari 1 hingga 1355. Vektor **x** dan $\mathbf{x}_{sv(i)}$ memiliki ukuran (970×1). Nilai α merupakan *lagrange multiplier* dari *support vector*. Nilai *y* merupakan label kelas (+1 dan -1) dan nilai -81,2941 adalah nilai bias.

Berikut pada Tabel 4.15 adalah contoh perbandingan kelas sentimen awal dan hasil klasifikasi dengan model terbaik untuk masing-masing maskapai.

Tabel 4.15 Perbandingan Kelas Sentimen Awal dan Hasil Klasifikasi

	Data	Kelas Awal	Kelas Prediksi
Air Asia	@AirAsia_indo ini gimana ya, saya sudah transfer tp belum bisa konfirmasi pembayarannya	Negatif	Negatif
	Udah lengkap, praktis, dan gak nguras dompet deh buruan pesan tiket di Aplikasi @AirAsia_indo dijaman amin eh *dijamin aman #MengejarKursiGratis	Positif	Positif
Lion Air	@zeazeejiggy @HaweDit @MuhadklyAcho @LionAirID @lionairgroup Kantorku udah ngeluarin kebijakan ga boleh dinas pake lion, kak untungnya	Negatif	Negatif

Tabel 4.15 Perbandingan Kelas Sentimen Awal dan Hasil Klasifikasi (lanjutan)

Lion Air	kini sudah OnTime, @lionairgroup ciptakan perjalanan Udara yang lebih berkesan capaian tingkat ketepatan waktu (on time) 85,2% sepanjang April 2019 dengan total 12.300 penerbangan dari rata-rata 400-420 frekuensi terbang per hari https://t.co/8csAxtPSzj	Positif	Positif

4.3.5 Ilustrasi Prediksi dengan Model SVM Terbaik

Klasifikasi termasuk metode *supervised* dimana memiliki *input* dan *output* yang dapat dibuat menjadi suatu model hubungan matematis. Sehingga mampu melakukan prediksi berdasarkan data yang telah ada sebelumnya. Setelah mendapat model terbaik yang diperoleh melalui metode *Spport Vector Machine* maka berikut adalah ilustrasi perhitungan klasifikasi sebagai prediksi di masa mendatang. Ilustrasi menggunakan model persamaan terbaik untuk AirAsia pada persamaan (4.3).

Tabel 4.16 Contoh Data Uji

Data Uji ke-	X_1	X_2	...	X_{515}
1	0	0	...	0,402011
2	0,500973	0	...	0
3	0	0	...	0
4	0	0,406908	...	0

Misal diberikan empat data uji dengan masing-masing memiliki 515 fitur yang dapat dilihat melalui Tabel 4.16. Melalui Tabel 4.16 maka dapat digambarkan oleh vektor untuk data uji pertama pada persamaan (4.6). Serta vektor $\mathbf{x}_{sv(i)}$ pada persamaan (4.7) adalah data *support vector* ke-*i*, dimana i = 1, 2,..., 1114.

$$\mathbf{x}^T = \begin{bmatrix} 0 & 0 & \cdots & 0,402011 \end{bmatrix} \quad (4.6)$$

$$\mathbf{x}_{sv(1)}{}^T = \begin{bmatrix} 0 & 0,634291 & \cdots & 0 \end{bmatrix}, \mathbf{x}_{sv(2)}{}^T = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}, \ldots,$$

$$\mathbf{x}_{sv(1114)}{}^T = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \quad (4.7)$$

Dengan mengaplikasikan parameter γ optimum sebesar 2^1 pada persamaan (4.2) maka hasil perhitungan pada fungsi *kernel* RBF untuk data uji pertama adalah seperti pada persamaan (4.8).

$$K\left(\mathbf{x}, \mathbf{x}_{sv(1)}\right) = 0,018316,\ K\left(\mathbf{x}, \mathbf{x}_{sv(2)}\right) = 0,021429,$$
$$\ldots,$$
$$K\left(\mathbf{x}, \mathbf{x}_{sv(1114)}\right) = 0,028862 \qquad (4.8)$$

Nilai hasil perhitungan fungsi *kernel* pada persamaan (4.8) yang berupa skalar. Melalui model terbaik pada persamaan (4.3) diperoleh pula α_i dan y_i. Sehingga diperoleh hasil seperti pada Tabel 4.17.

Tabel 4.17 Ilustrasi Perhitungan Fungsi *Hyperplane* Data Uji Pertama

i	α_i	y_i	$K(\mathbf{x}, \mathbf{x}_{sv(i)})$	$\alpha_i\, y_i\, K(\mathbf{x}, \mathbf{x}_{sv(i)})$
1	0,3918	-1	0,018316	-0,00718
2	0,0791	-1	0,021429	-0,00169
...	...	...	...	...
1114	0,2201	-1	0,028862	-0,00636

Dengan menerapkan fungsi *hyperplane* optimum pada persamaan (4.3) dimana nilai bias sebesar -0,3765, maka hasil perhitungan untuk memeroleh nilai $f(\mathbf{x})$ data uji pertama hingga keempat ditunjukkan pada Tabel 4.18.

Tabel 4.18 Ilustrasi Perhitungan Klasifikasi

i	$\alpha_i\, y_i\, K(\mathbf{x}, \mathbf{x}_{sv(i)})$			
	Data Uji 1	**Data Uji 2**	**Data Uji 3**	**Data Uji 4**
1	-0,00718	-0,00718	-0,00718	-0,02944
2	-0,00169	-0,00145	-0,00145	-0,00169
...	...	...	...	...
1114	-0,00636	-0,00580	-0,00403	-0,00693
Total	1,376571	-0,62358	1,376419	-0,62393
***f*(x)**	1,000071	-1,00008	0,999919	-1,00043
Klasifikasi	+1	-1	+1	-1
Keterangan	Positif	Negatif	Positif	Negatif

Dari nilai total masing-masing data uji yang merupakan hasil perhitungan $\sum_{i=1}^{1114} \alpha_i y_i K\left(\mathbf{x}, \mathbf{x}_i\right)$ pada Tabel 4.18, maka dapat dilihat melalui nilai $f(\mathbf{x})$ bahwa data uji pertama dan ketiga memiliki nilai

$f(\mathbf{x})$ lebih dari 0 maka hasil klasifikasi kedua data uji tersebut adalah kelas +1 (Positif). Sebaliknya, data uji kedua dan keempat masuk ke dalam klasifikasi kelas -1 (Negatif) karena nilai $f(\mathbf{x})$ yang diperoleh kurang dari 0.

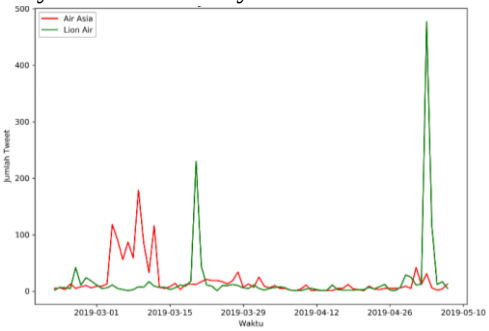
4.4 Analisis Kompetitif

Maskapai penerbangan yang beroperasi di Indonesia, AirAsia dan LionAir merupakan maskapai berbiaya rendah dan saling mengklaim bahwa masing-masing adalah rival. Dalam penelitian ini, akan digambarkan bagaimana karakteristik *tweet* yang dikicaukan pengguna *Twitter* kepada akun resmi masing-masing.

Tabel 4.19 Perbandingan Aktivitas Akun *Twitter* Resmi

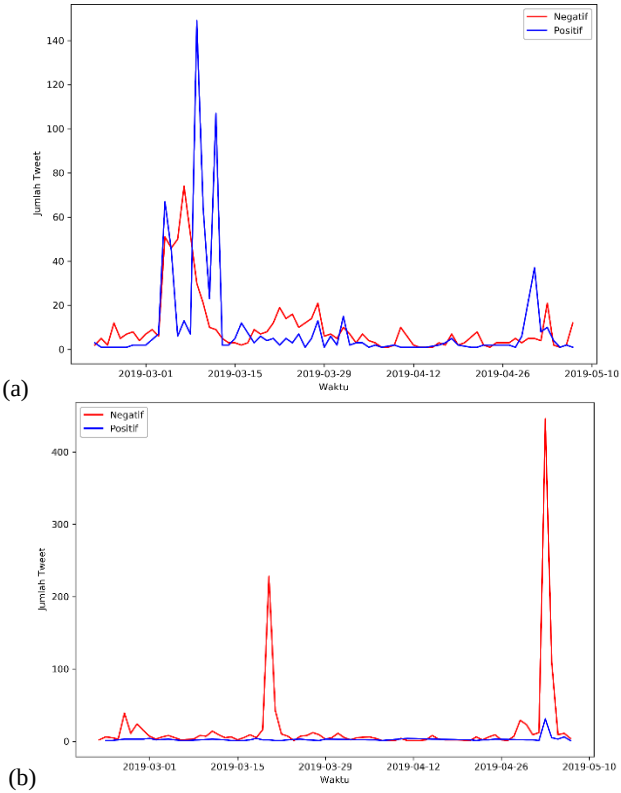
Maskapai	Bergabung di *Twitter*	Jumlah *Tweet*	*Follower*
AirAsia (@airasia_indo)	Februari 2010	52358	4099476
LionAir (@lionairgroup)	Agustus 2012	1741	11099

Pada Tabel 4.19 dapat dilihat bahwa baik jumlah *tweet* maupun *follower* (pengikut) akun resmi AirAsia lebih banyak dibanding LionAir. Jumlah *Tweet* untuk AirAsia per 18 Mei 2019 mencapai lebih dari 52 ribu, sekitar 30 kali lebih banyak dibanding LionAir. Hal tersebut mengindikasikan bahwa AirAsia lebih aktif mengabarkan, menyapa, menanggapi konsumen dibanding LionAir. Sikap AirAsia tersebut juga dapat mengindikasikan mengapa jumlah pengikut akun resminya mencapai 4 juta, sekitar 370 kali lipat lebih banyak dibanding LionAir meskipun selisih waktu bergabung keduanya di *Twitter* hanya dua tahun.



Gambar 4.5 Jumlah *Tweet* Berdasarkan Waktu

Dapat dilihat melalui Gambar 4.5, jumlah *tweet* bersentimen dari waktu ke waktu baik untuk AirAsia maupun LionAir cenderung sama. Namun ada beberapa waktu, jumlah *tweet* bersentimen yang ditujukan kepada AirAsia dan LionAir mengalami peningkatan cukup tajam. Ada dua waktu dimana jumlah *tweet* bersentimen untuk LionAir mencapai masing-masing lebih dari 200 dan 400. Jumlah tersebut lebih tinggi dibanding jumlah *tweet* bersentimen terbanyak yang dimiliki AirAsia.



Gambar 4.6 Jumlah *Tweet* Berdasarkan Waktu dan Sentimen (a) AirAsia (b) LionAir

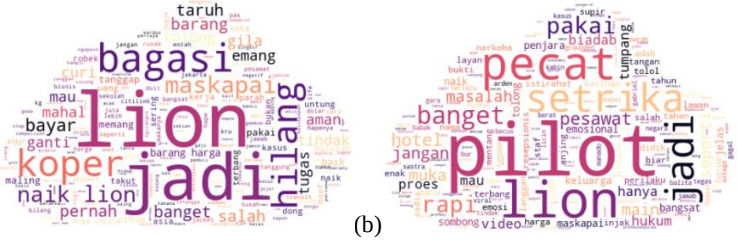
Gambar 4.6 (a) menunjukkan bahwa jumlah *tweet* yang menyebut akun resmi AirAsia melonjak cukup tajam pada tanggal

9 Maret 2019, dimana *tweet* yang mengandung sentimen positif lebih mendominasi. Di sisi lain, LionAir justru mengalami pelonjakan jumlah *tweet* bersentimen negatif cukup tajam pada tangal 20 Maret 2019 dan 3 Mei 2019 yang dapat dilihat melalui Gambar 4.6 (b) dan *tweet* negatif lebih mendominasi.

Gambar 4.7 *Wordcloud* untuk AirAsia

Berdasarkan Gambar 4.7, AirAsia mengadakan kuis pada tanggal 9 Maret 2019. Terlihat bahwa terdapat kata 'kursi gratis' dengan ukuran paling besar. Pemenang kuis akan mendapat promo kursi gratis, sehingga pada waktu itu pengguna *Twitter* beramai-ramai serta mengajak pengguna lainnya untuk mengikuti kuis. Hal tersebut tentu dapat meningkatkan reputasi AirAsia dan mendapat pandangan baik dari pengguna *Twitter* khususnya konsumen.

Pada kasus LionAir, setelah divisualisasikan dengan *wordcloud*, terlihat pada Gambar 4.8 (a) bahwa pada tanggal 20 Maret 2019 terjadi kasus pembobolan koper yang dikicaukan oleh salah satu pengguna *Twitter*. Tak lama setelah dikicaukan, banyak pengguna *Twitter* lainnya yang membagikan kicauan tersebut dan menceritakan pengalaman yang sama sehingga peristiwa tersebut menjadi ramai diperbincangkan di *Twitter*.



(a) (b)

Gambar 4.8 *Wordcloud* untuk LionAir (a) 20 Maret 2019 (b) 3 Mei 2019

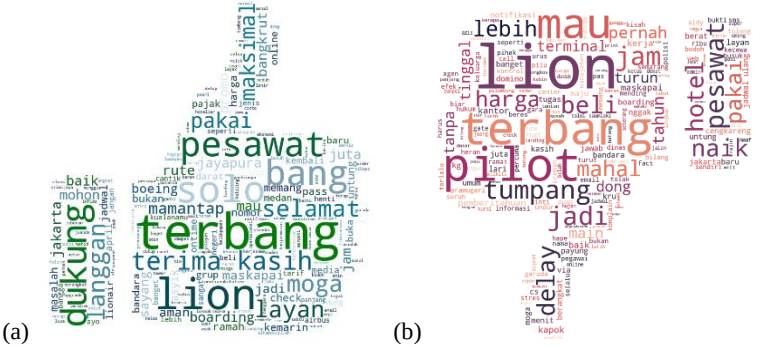
Sementara pada Gambar 4.8 (b), kasus tidak menyenangkan oleh LionAir terjadi lagi pada tanggal 3 Mei 2019. Kala itu, tersebar video *cctv* salah satu hotel yang menjadi perbincangan di *Twitter,* dimana salah satu pilot LionAir yang memukul salah satu pegawai hotel yang diketahui dengan alasan karena hasil setrika seragam pilot yang menurutnya kurang rapi. Dengan cepat pengguna *Twitter* mengecam tindakan pilot tersebut dengan mengicaukan kalimat yang kurang pantas sehingga muncul kata ‘pecat’, ‘masalah’, ‘biadab’, hingga ‘sombong’. Sehingga pada tanggal tersebut *tweet* yang menyebut akun resmi LionAir lebih didominasi oleh *tweet* bersentimen negatif. Berdasarkan dua kasus yang menjadi ramai diperbincangkan di *Twitter* tersebut maka pantas bahwa persentase jumlah *tweet* untuk LionAir yang mengandung sentimen negatif lebih besar dibanding sentimen positif. Kondisi tersebut diindikasikan dapat menurunkan reputasi LionAir di mata pengguna *Twitter* khususnya konsumen.

Pada *wordcloud tweet* bersentimen positif yang dapat dilihat melalui Gambar 4.9 (a), pengguna *Twitter* yang menyebut akun resmi AirAsia sering menggunakan kata ‘terbang’, ‘buka’, ‘rute’, dan ‘dong’. Hal tersebut menunjukkan bahwa sering dikicaukan permintaan untuk membuka rute penerbangan di daerah-daerah tertentu oleh pengguna *Twitter*. Oleh sebab itu, maka muncul pula kata-kata berupa daerah-daerah di Indonesia maupun luar Indonesia yang cukup sering dikicaukan seperti ‘jakarta’, ‘jogja’, ‘surabaya’, ‘bandung’, ‘makassar’, hingga ‘jeju’.

Gambar 4.9 *Wordcloud* untuk AirAsia (a) Positif (b) Negatif

Selain kata-kata tersebut, terdapat kata ‘terima’, ‘kasih’, ‘keren’, dan ‘suka’ yang dapat mengindikasikan bahwa reputasi AirAsia cukup baik di mata pengguna *Twitter*. Sementara pada *wordcloud* untuk *tweet* yang mengandung sentimen negatif pada Gambar 4.9 (b), kata ‘harga’ dan ‘bayar’ memiliki frekuensi kemunculan yang lebih banyak. Hal tersebut menunjukkan bahwa pengguna *Twitter* sering mengeluhkan harga penerbangan yang ditawarkan AirAsia serta metode pembayaran di *website* AirAsia. Terdapat pula kata ‘halo’, ‘tolong’, dan ‘telepon’ yang menunjukkan bahwa pengguna *Twitter* meminta tolong dengan menyebut akun remsi AirAsia terkait sesuatu, ada pula yang mengeluhkan karena telepon *call center* yang sering tidak bisa dihubungi.

Pada *wordcloud tweet* bersentimen untuk LionAir yang ditampilkan melalui Gambar 4.10 (a), terlihat bahwa kata ‘terbang’, ‘dukung’, ‘selamat’ dan ‘terima kasih’ mendominasi di kumpulan *tweet* yang mengandung sentimen positif. Hal tersebut menunjukkan bahwa pengguna *Twitter* masih memberikan dukungan untuk penerbangan yang ditawarkan meskipun pandangan kepada LionAir kurang baik dilihat melalui persentase *tweet* negatif yang lebih banyak.

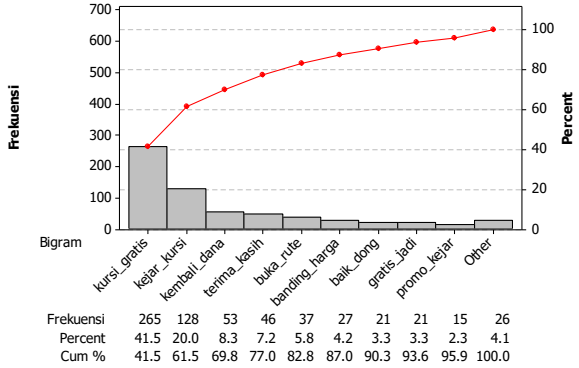


(a) (b)

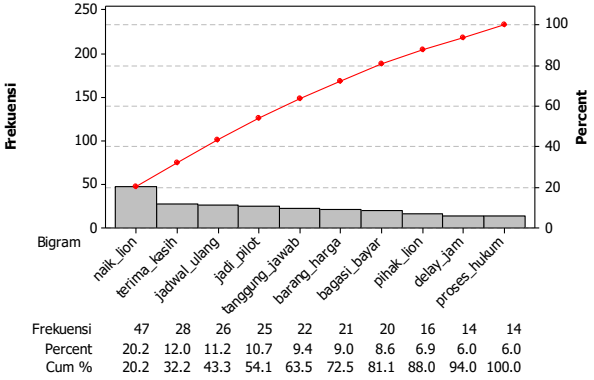
Gambar 4.10 *Wordcloud* untuk LionAir (a) Positif (b) Negatif

Sementara pada *wordcloud* untuk *tweet* negatif pada Gambar 4.10 (b) terlihat bahwa kata ‘pilot’ dan ‘terbang’ sering dikicaukan pengguna *Twitter* saat menyebut akun resmi LionAir. Selain itu, terdapat pula kata ‘tumpang’, ‘delay’, ‘harga’, dan

'mahal' dengan ukuran yang cukup besar. Hal tersebut menunjukkan bahwa penumpang sering mengeluh karena *delay* (keterlambatan penerbangan) serta harga penerbangan yang mahal.



(a)

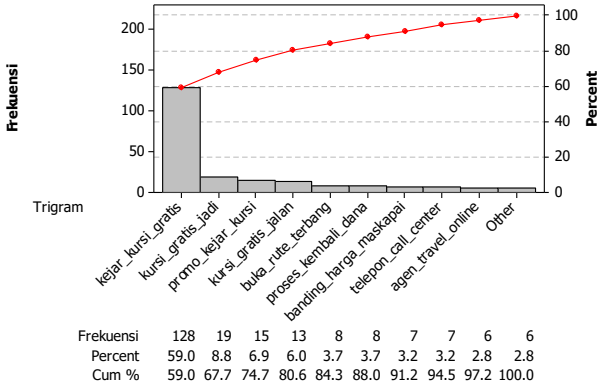


(b)

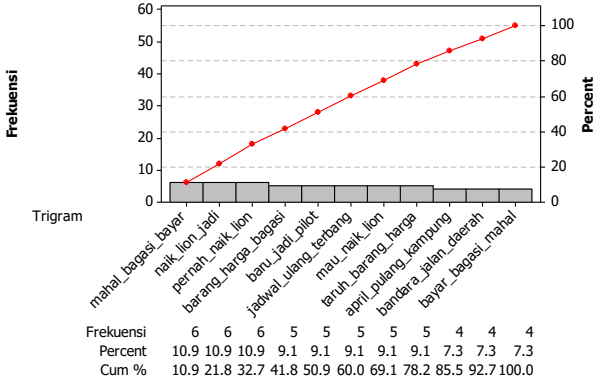
Gambar 4.11 *Pareto Chart* Bigram (a) AirAsia (b) LionAir

Terdapat beberapa kumpulan dua kata yang ditunjukkan oleh *pareto chart* pada Gambar 4.11. Dengan menggunakan 10 kata dengan frekuensi terbanyak, pada *pareto chart* bigram untuk AirAsia terdapat 'kursi_gratis' dengan persentase kemunculan paling besar, diikuti oleh 'kejar_kursi', 'kembali_dana', dan 'terima_kasih'. Hal tersebut terjadi karena AirAsia sering mengadakan promo kursi gratis. Kemudian pengguna *Twitter* beramai-ramai

mengikuti dan menjadi sering diperbincangkan. Selain itu, terdapat pula ‘kembali_dana’ yang menunjukkan beberapa pengguna *Twitter* cukup sering membahas mekanisme pengembalian dana oleh pihak AirAsia. Di sisi lain, untuk *pareto chart* bigram LionAir pada Gambar 11 (a), terdapat ‘naik_lion’ yang memiliki frekuensi kemunculan terbanyak. Selain itu terdapat ‘terima_kasih’, ‘jadwal _ulang’, ‘jadi_pilot’, dan ‘tanggung_jawab’ juga cukup sering dikicaukan. Hasil *pareto chart* bigram pada Gambar 4.11 tersebut dapat didukung dengan hasil *pareto chart* trigram yang ditunjukkan Gambar 4.12.



(a)



(b)

Gambar 4.12 *Pareto Chart* Trigram (a) AirAsia (b) LionAir

Pada Gambar 12 (a), dapat dilihat bahwa terdapat kata trigram ‘kejar_kursi_gratis’ yang muncul paling sering, dengan persentase 59% jika digunakan 10 kata dengan frekuensi tertinggi. Hal ini mendukung hasil *pareto chart* sebelumnya yang dapat mengindikasikan bahwa antusias pengguna *Twitter* cukup tinggi unuk mengikuti kuis berhadiah kursi gratis. Sementara pada kasus LionAir, kata trigram ‘mahal_bagasi_bayar’ paling sering dikicaukan di *Twitter*. Selain itu, terdapat pula ‘jadwal_ulang_terbang’ yang menunjukkan pengguna *Twitter* cukup sering membahas mekanisme *reschedule* LionAir.

Dilihat melalui Gambar 4.11 dan Gambar 4.12, pengguna *Twitter* cenderung lebih sering berkicau tentang AirAsia dibanding LionAir. Hal tersebut berdasarkan tiga deretan kata dengan frekuensi terbanyak. Kemunculan tiga deretan kata bigram untuk AirAsia pada Gambar 4.11 memiliki persentase kumulatif mendekati 70%, sementara LionAir hanya 43,3%. Begitu pula dengan Gambar 4.12. Persentase kumulatif kemunculan tiga deretan kata trigram untuk AirAsia mencapai 74,7%, sementara LionAir kurang dari 40%.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis dan pembahasan yang telah dilakukan pada bab 4, maka diperoleh kesimpulan sebagai berikut.

1. Persentase kelas sentimen untuk AirAsia cenderung seimbang dimana sebesar 49,2% *tweet* memiliki sentimen positif dan 50,8% sisanya negatif. Sebaliknya, LionAir memiliki persentase kelas sentimen yang tidak seimbang dengan 8,7% *tweet* merupakan *tweet* positif dan 91,3% adalah *tweet* negatif.
2. Penerapan metode *Support Vector Machine kernel* linier pada data *tweet* bersentimen untuk AirAsia menghasilkan nilai rata-rata AUC terbaik sebesar 0,8511, 0,7473, dan 0,6265 dimana masing-masing secara urut untuk *n*-gram jenis unigram, bigram, dan trigram dengan parameter $C = 10^0$. Pada *kernel* RBF dihasilkan rata-rata AUC terbaik untuk unigram ($C = 10^1$, $\gamma = 2^1$), bigram ($C = 10^1$, $\gamma = 2^{-3}$), dan trigram ($C = 10^1$, $\gamma = 2^1$) adalah 0,8619, 0,7810, dan 0,6336. Sementara itu, karena persentase kelas sentimen pada LionAir tidak seimbang maka diterapkan SMOTE untuk mengatasi hal tersebut. Dengan menggunakan *kernel* linier, rata-rata AUC terbaik sebesar 0,7885, 0,7312, dan 0,5550 dimana masing-masing secara urut untuk *n*-gram jenis unigram, bigram, dan trigram dengan parameter $C = 10^0$. Pada *kernel* RBF dihasilkan rata-rata AUC terbaik untuk unigram ($C = 10^2$, $\gamma = 2^{-9}$), bigram ($C = 10^4$, $\gamma = 2^{-9}$), dan trigram ($C = 10^1$, $\gamma = 2^{-3}$) adalah 0,7974, 0,7304, dan 0,5727. Hasil dari penerapan SMOTE dapat meningkatkan nilai AUC.
3. Jika hasil klasifikasi dibandingkan, maka dipilih metode *kernel* RBF menjadi metode terbaik karena mampu menghasilkan nilai AUC tertinggi bagi kedua maskapai. Digunakan parameter $C = 10^1$ dan $\gamma = 2^1$ dengan jenis *n*-gram berupa

unigram untuk AirAsia. Sementara LionAir menggunakan parameter $C = 10^2$ dan $\gamma = 2^{-9}$ (unigram) dan data latih sebelumnya telah diterapkan SMOTE.

4. Berdasarkan analisis kompetitif maka dapat disimpulkan bahwa reputasi atau pandangan publik khususnya pengguna *Twitter* untuk AirAsia lebih baik dibanding LionAir. Mulai dari aktivitas masing-masing akun resmi hingga karakteristik *tweet* yang menunjukkan bahwa AirAsia lebih aktif dan *tweet* positif yang lebih mendominasi dibanding LionAir.

5.2 Saran

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

1. Jika menggunakan *n*-gram maka lebih diperhatikan kembali mengenai *text pre processing* khususnya pada tahap penghapusan *stopwords* untuk menghasilkan skor evaluasi klasifikasi yang lebih tinggi.
2. Dapat dicobakan kombinasi *n*-gram, misalnya unigram dan bigram atau unigram dan trigram sebagai fitur.

DAFTAR PUSTAKA

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S., & Williams, H. E. (2007). Stemming Indonesian: A Confix-Stripping Approach. *ACM Transactions on Asian Language Information Processing, 6*(4), 1-33.

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2013). Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl, 5*(3), 1-30.

Allahyari, M., Safaei, S., Pouriyeh, S., Trippe, E., Assefi, M., Gutierrez, J., & Kochut, K. (2017). *A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques*. Canada: KDD Bigdas.

Arifin, A. Z., Mahendra, I. A., & Ciptaningtyas, H. T. (2009). Enhanced Confix Stripping Stemmer and Ants Algorithm for Classifying News Document in Indonesian Language. *The 5th International Conference on Information & Communication Technology and Systems*, (hal. 149-157).

Bekkar, M., Djemaa, D. K., & Alitouche, D. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications, 3*(10), 27-38.

Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *Integer Journal, 2*(1), 32-41.

Chandani, V., Wahono, R. S., & Purwanto. (2015). Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film. *Journal of Intelligent Systems, 1*(1), 56-60.

Chawla, N. V. (2005). *Data Mining and Knowledge Discovery Handbook* (*2nd Edition*). New York: Springer Science+Business Media.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research 16*, 321-357.

Fauzia, M. (2018). *Peringkat Maskapai Penerbangan Indonesia Berdasarkan Keamanan*. Dipetik November 28, 2018, dari Kompas: https://ekonomi.kompas.com/read/2018/11/01/105134326/peringkat-maskapai-penerbangan-indonesia-berdasarkan-keamanan

Han, J., Kamber, M., & Pei, J. (2006). *Data Mining: Concepts and Techniques* (*2nd Edition*). Waltham: Morgan Kaufmann.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (*3rd Edition*). Waltham: Morgan Kaufmann.

Hardle, W. K., Prastyo, D. D., & Hafner, C. (2014). Support Vector Machines with Evolutionary Feature Selection for Default Prediction. *The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics*, 346-373.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2016). *A Practical Guide to Support Vector Classification*. Diambil kembali dari Department of Computer Science & Information Engineering: https://www.csie.ntu.edu.tw/~cjlin/papers/-guide/guide.pdf

Huang, C. M., Lin, D. K., & Huang, S. Y. (2007). Model selection for support vector machines via uniform design. *Computational Statistics & Data Analysis*, *52*, 335-346.

Indhiarta, W. C. (2017). *Penggunaan N-gram pada Analisa Sentimen Pemilihan Kepala Daerah Jakarta Menggunakan Algoritma Naive Bayes.* Universitas Muhammadiyah Surakarta. Surakarta: Electronic Thesis and Disertations.

Iriawan, N., Fithriasari, K., Ulama, B. S., Suryaningtyas, W., Pangastuti, S. S., Cahyani, N., & Qadrini, L. (2018). On The Comparison: Random Forest, SMOTE-Bagging, and Bernoulli Mixture to Classify Bidikmisi Dataset in East Java. *2018 International Conference on Computer Engineering, Network and Intelligent Multimedia, CENIM 2018.* Surabaya: Institute of Electrical and Electronics Engineers.

Kurniasari, S. R. (2018). *Implementasi SVM dan Asosiasi untuk Sentiment Analysis Data Ulasan The Phoenix Hotel Yogyakarta pada Situs TripAdvisor*. Yogyakarta: Jurusan Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia.

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

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval.* Cambrigde: Cambrigde University Press.

Montgomery, D. C. (2009). *Introduction to Statistical Quality Control* (*6th Edition*). United States of America: John Wiley & Sons, Inc.

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundation and Trends in Information Retrieval, 2*(1-2), 1-135.

Prasetyo, E. (2012). *Data Mining: Konsep dan Aplikasi Menggunakan MATLAB.* Yogyakarta: Andi Offset.

Pujadayanti, I., Fauzi, M. A., & Sari, Y. A. (2018). Prediksi Rating Otomatis pada Ulasan Produk Kecantikan dengan Metode Naive Bayes dan N-gram. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2*(11), 4421-4427.

Qeis, M. I. (2015). Aplikasi Wordcloud Sebagai Alat Bantu Analisis Wacana. *International Conference on Language, Culture, and Society - ICLCS LIPI 2015.* Jakarta.

Rahman, F. (2018). *Klasifikasi Emosi untuk Teks Berbahasa Indonesia pada Pengguna Twitter Mengenai Presiden Joko Widodo.* Surabaya: Program Studi Sarjana Departemen Statistika Fakultas Matematika, Komputasi, dan Sains Data ITS.

Ratnasari, B. C., & Sankhyaadi, A. (2018). *Kali Ke-6 AirAsia Sabet Gelar Maskapai Berbiaya Hemat Terbaik di Asia*. Dipetik Maret 9, 2019, dari Kumparan: https://kumparan.com/-

@kumparantravel/kali-ke-6-airasia-sabet-gelar-maskapai-berbiaya-hemat-terbaik-di-asia-1536316191247795070

Reyhana, Z., Fithriasari, K., Atok, M., & Iriawan, N. (2018). Linking Twitter Sentiment Knowledge with Infrastructure Development. *Malaysian Journal of Industrial and Applied Mathematics*, 91-102.

Siringoringo, R. (2018). Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan k-Nearest Neighbor. *Jurnal ISD, 3*(1), 44-40.

Skytrax. (2018). *World's Best Low-Cost Airlines 2018*. Dipetik Maret 9, 2019, dari Skytrax: World Airline Awards: https://www.worldairlineawards.com/worlds-best-low-cost-airlines-2018/

Subastian, B., & Nurjanah, R. (2018). *Lion Air, Singa Langit yang Terempas*. Dipetik Maret 9, 2019, dari Kumparan: https://-kumparan.com/@kumparannews/lion-air-singa-langit-yang-terempas-1541383250707985089

Sulistyo, W. (2008). Klasifikasi Dokumen Berbahasa Inggris Berdasarkan Weighted-Term. *Jurnal Teknologi Informasi-Aiti, 5*(1), 87-100.

Susilowati, E., Sabariah, M. K., & Gozali, A. A. (2015). Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas pada Twitter. *e-Proceeding of Engineering, 2*, hal. 1478-1484. Bandung.

Tjiptono, F. (2005). *Service Quality Satisfication.* Yogyakarta: Andi.

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of Sentiment Reviews using n-gram Machine Learning Approach. *Expert Systems With Applications 57*, 117-126.

WeAreSocial. (2019, Januari 31). *Digital 2019 Indonesia.* Diambil kembali dari Data Reportal: https://datareportal.com/-reports/digital-2019-indonesia

LAMPIRAN

Lampiran 1. *Syntax Twitter Crawling* Menggunakan RStudio

```
setwd('C:/Users/User/Documents/ crawling')

# CRAWLING TWITTER ============================
library("rtweet")
library(magrittr)

# MEMBUAT TOKEN ================================
appname <- "nama app pengguna"
key <- "consumer key pengguna"
secret <- "consumer secret pengguna"

twitter_token <- create_token(app = appname, consumer_key = key,
                              consumer_secret = secret)

# AMBIL TWEET ==================================
tweet_airasiaindo <- search_tweets("@AirAsia_indo", n = 18000,
                                   include_rts = FALSE,
                                   token = twitter_token) %>%
                     select(status_id, created_at, screen_name, text)

tweet_airasiaindo$created_at2 <- tweet_airasiaindo$created_at
                                 %>% str_sub(1,10)
tweet_airasiaindo$screen_name2 <- tweet_airasiaindo$screen_name
                                  %>% str_to_lower
tweet_airasiaindo <-
tweet_airasiaindo[order(tweet_airasiaindo$status_id),] %>%
subset(!duplicated(tweet_airasiaindo$status_id))

# SIMPAN =======================================
write.csv(tweet_airasiaindo, file = "tweet_airasiaindo.csv")
```

Catatan: untuk LionAir diganti “@lionairgroup” sebagai *keyword*

Lampiran 2. Data *Tweet* AirAsia dan LionAir

AirAsia

No	*Created At*	*Text*	*Username*	Sentimen
1	2/21/2019	@AirAsia_indo Kalau pemesanan tiket grup kemana min..? Terima kasih respon cepat nya...	muhtarom_habib	1
2	2/21/2019	@jsamodra @AirAsia @AirAsia_indo Dasar kamu antek Jepang. #TigaA.	sandrinarin	0
.	.	.		.
.	.	.		.
1445	5/7/2019	@35ury4 @PartaiSocmed @traveloka @AirAsia_indo @Citilink Kita customer yg udah bayar mahal malah berasa jongos #PecatBudiKarya	sidewii	0

LionAir

No	*Created At*	*Text*	*Username*	Sentimen
1	2/21/2019	@tijabar Belum ada penurunan tiket rute Palembang - Jakarta. Beberapa bulan lalu bisa dapat 300 ribuan. Sekarang paling murah 680 ribu. Kenaikan 100%. @kemenhub151 @lionairgroup @IndonesiaGaruda @SriwijayaAir @Citilink @BatikAirINA	laskapalembang	0

Lampiran 2. Data *Tweet* AirAsia dan LionAir (lanjutan)

No	*Created At*	*Text*	*Username*	Sentimen
2	2/21/2019	@susipudjiastuti Siap Bu Susi... Cuma cargo dari Saumlaki ke Jakarta by Lion Air kok sampai 120.000/kg-nya? @InfoMenhub @lionairgroup	winsamb	0
.	.	.		.
.	.	.		.
1431	5/7/2019	@LionAirID @lionairgroup saya miris liat solusi dari kalian kaya gini. Uang 13 juta itu bukan daun bapak/ibu yg terhormat. Anak 3.5 tahun mana bisa bawa barang seberat 7kg hati nurani kalian dimana sih? Heran seenaknya aja kaya cuci tangan ga mau tahu	xoxodpk	0

Lampiran 3. *Syntax Input* dan *Pre Processing* Menggunakan Python

```
# IMPORT LIBRARY ================================
import pandas as pd
import seaborn as sns
import string
import nltk
from nltk.tokenize import word_tokenize
import re
import sys
import os
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from wordcloud import WordCloud, STOPWORDS
```

Lampiran 3. *Syntax Input* dan *Pre Processing* Menggunakan Python (lanjutan)

```
# INPUT DATA ==================================
airasia = pd.read_csv(r'C:\Documents\ airasia_ sentimen.csv', header
= 0, encoding = 'latin-1')
input_df = pd.DataFrame(data = airasia)
pd.options.display.max_colwidth = 2000
airasiatweet = input_df['text']

# PRE PROCESSING ==============================
# Cleaning

# a.menghapus Link
airasiaclearlink = []
for line in airasiatweet:
    result = re.sub(r"http\S+","",line)
    airasiaclearlink.append(result)

# b. menghapus Username
airasiaclearusername = []
for line in airasiaclearlink:
    result = re.sub(r"@\S+","",line)
    airasiaclearusername.append(result)

# c.menghapus Hashtag
airasiaclearhashtag = []
for line in airasiaclearusername:
    result = re.sub(r"3(\w+)","",line)
    airasiaclearhashtag.append(result)

# d. menghapus angka
airasiaclearnum = []
for line in airasiaclearhashtag:
    result = re.sub("\d"," ",line)
    airasiaclearnum.append(result)
```

Lampiran 3. *Syntax Input* dan *Pre Processing* Menggunakan Python (lanjutan)

```
# e. menghapus baris baru
airasiaclearline = []
for line in airasiaclearnum:
    result = re.sub("\n"," ",line)
    airasiaclearline.append(result)

# f. menghapus emoticon
airasiaclearemoticon = []
for line in airasiaclearline:
    result = re.sub(r'<.*?>',"",line)
    airasiaclearemoticon.append(result)

# g. menghapus punctuation
airasiaclearpunctuation = []
for line in airasiaclearemoticon:
    result = re.sub(r"[^\w\s]"," ",line)
    airasiaclearpunctuation.append(result)

# h.menghapus space berlebih
airasiaclear = []
for line in airasiaclearpunctuation:
    result = re.sub(r'\s+',' ',line)
    airasiaclear.append(result)

# Case Folding
airasialower = []
for line in airasiaclear:
    a = line.lower()
    airasialower.append(a)

# Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
data_stemmed = map(lambda x: stemmer.stem(x), airasialower)
datastemmed = list(data_stemmed)
```

Lampiran 3. *Syntax Input* dan *Pre Processing* Menggunakan Python (lanjutan)

```
# Sinonim Replace
Sinonim replace
kata = {"aamiin":"amin", "amiin":"amin", "acc ":"akun ",
"account":"akun", "adain":"ada",
….
"yuks":"ayo"}

def replace_all(teks, dic):
    for i, j in dic.items():
        teks = teks.replace(i, j)
    return teks

import collections
from collections import OrderedDict

dic = OrderedDict(kata)
datachange = []
for line in datastemmed:
    result = replace_all(line, dic)
    datachange.append(result)

# Stopword + Tokenizing
stopwords = open(r'C: \Documents \airasia_stopword.txt').read()

datafinal = []
df = []
for line in datachange:
    wt_data = word_tokenize(line)
    wt_data = [word for word in wt_data if not word in stopwords and
not word[0].isdigit()]
    datafinal.append(wt_data)
    df.append(" ".join(wt_data))

airasiafinal = df
text = pd.DataFrame(df)
input_df['finaltext'] = text
```

Lampiran 3. *Syntax Input* dan *Pre Processing* Menggunakan Python (lanjutan)

```
# DTM dan TF-IDF
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

# Unigram
vectorizer = CountVectorizer(min_df = 3)
DTM = vectorizer.fit_transform(airasiafinal)
DTM_ = pd.DataFrame(DTM.toarray(), columns =
vectorizer.get_feature_names())
DTM_df = pd.concat([airasiatweet, DTM_], axis = 1)
DTM_.to_csv('DTMa.csv')
DTM_df.to_csv('DTMa_df.csv')
terms = vectorizer.get_feature_names()
print(len(terms))
for i, feature in enumerate(vectorizer.get_feature_names()):
   print(i, feature)
vectorizer = TfidfVectorizer(min_df = 3)
TFIDF = vectorizer.fit_transform(airasiafinal)
TFIDF_ = pd.DataFrame(TFIDF.toarray(), columns =
vectorizer.get_feature_names())
TFIDF_.to_csv('TFIDFa.csv')

# Bigram
vectorizer = CountVectorizer(min_df = 2, ngram_range = (2,2))
DTM_BG = vectorizer.fit_transform(airasiafinal)
term = vectorizer.get_feature_names()
term1 = [word.replace(' ','_') for word in term]
DTM_BG_ = pd.DataFrame(DTM_BG.toarray(), columns = term1)
DTM_BG_.to_csv('DTMa_BG.csv')
vectorizer = TfidfVectorizer(min_df = 2, ngram_range = (2,2))
TFIDF_BG = vectorizer.fit_transform(airasiafinal)
term = vectorizer.get_feature_names()
term1 = [word.replace(' ','_') for word in term]
TFIDF_BG_ = pd.DataFrame(TFIDF_BG.toarray(), columns =
term1)
TFIDF_BG_.to_csv('TFIDFa_BG.csv')
```

Lampiran 3. *Syntax Input* dan *Pre Processing* Menggunakan Python (lanjutan)

```
# Trigram
vectorizer = CountVectorizer(min_df = 3, ngram_range = (3,3))
DTM_TG = vectorizer.fit_transform(airasiafinal)
term = vectorizer.get_feature_names()
term1 = [word.replace(' ','_') for word in term]
DTM_TG_ = pd.DataFrame(DTM_TG.toarray(), columns = term1)
DTM_TG_.to_csv('DTMa_TG.csv')

vectorizer = TfidfVectorizer(min_df = 2, ngram_range = (3,3))
TFIDF_TG = vectorizer.fit_transform(airasiafinal)
term = vectorizer.get_feature_names()
term1 = [word.replace(' ','_') for word in term]
TFIDF_TG_ = pd.DataFrame(TFIDF_TG.toarray(), columns =
term1)
TFIDF_TG_.to_csv('TFIDaF_TG.csv')
```

Lampiran 4. *Syntax Wordcloud* dan Karakteristik Data Menggunakan Python

```
airasia['sentimen'].replace((0, 1),('Negatif', 'Positif') , inplace = True)
# Pie Chart ======================================
sentimen = airasia.groupby('sentimen').size().reset_index(name =
'counts')
fig, ax = plt.subplots(figsize = (12, 7), subplot_kw = dict(aspect =
"equal"), dpi = 80)

data = sentimen['counts']
categories = sentimen['sentimen']
def func(pct, allvals):
   absolute = int(pct/100.*np.sum(allvals))
   return "{:.1f}% ({:d})".format(pct, absolute)
wedges, texts, autotexts = ax.pie(data,
                          autopct = lambda pct: func(pct, data),
                          textprops = dict(color = "w"),
                          colors = plt.cm.Dark2.colors,
                          startangle = 140)
```

Lampiran 4. *Syntax Wordcloud* dan Karakteristik Data Menggunakan Python (lanjutan)

```
ax.legend(wedges, categories, title = "Sentimen", loc = "center left",
bbox_to_anchor = (1, 0, 0.5, 1))
plt.setp(autotexts, size = 10, weight = 700)
ax.set_title("Pie Chart Sentimen Air Asia")
plt.savefig('airasia_piechart', dpi = 300)
plt.show()

# Time Series Plot 1 ==================================
airasia['created_at2'] = pd.to_datetime(airasia['created_at2'])
a = pd.DataFrame(airasia.text.groupby(by =
airasia['created_at2'].dt.date).count())

t = pd.Series(data = a.text.values, index = a.index.values)
t.plot(figsize = (10,7), color = 'r')
plt.xlabel("Waktu")
plt.ylabel("Jumlah Tweet")
plt.title("Jumlah Tweet Air Asia", fontsize = 20)
plt.savefig('airasia_jumlahtweet', dpi = 300)

# Time Series Plot 2 ==================================
neg = pd.DataFrame(airasia['created_at2'].loc[airasia['sentimen'] ==
'Negatif'])
neg_ = pd.DataFrame(neg.groupby(by =
neg['created_at2'].dt.date).count())

pos = pd.DataFrame(airasia['created_at2'].loc[airasia['sentimen'] ==
'Positif'])
pos_ = pd.DataFrame(pos.groupby(by =
pos['created_at2'].dt.date).count())

negg = pd.Series(data = neg_.created_at2.values, index =
neg_.index.values)
poss = pd.Series(data = pos_.created_at2.values, index =
pos_.index.values)
```

Lampiran 4. *Syntax Wordcloud* dan Karakteristik Data Menggunakan Python (lanjutan)

```
negg.plot(figsize=(10,7), label="Negatif", legend=True, color ='r')
poss.plot(figsize=(10,7), label="Positif", legend=True, color = 'b')
plt.xlabel("Waktu")
plt.ylabel("Jumlah Tweet")
plt.title("Jumlah Tweet Air Asia Berdasarkan Sentimen", fontsize =
20)
plt.savefig('airasia_jumlahtweet_bysentimen', dpi = 300)

# Wordcloud ====================================
def generate_wordcloud(words, mask, color, save):
    word_cloud = WordCloud(width = 1600, height = 800,
                    background_color = 'white', colormap = color,
                    mask = mask).generate(words)
    plt.figure(figsize = (20, 10))
    plt.imshow(word_cloud, interpolation = 'bilinear')
    plt.axis('off')
    plt.tight_layout(pad = 0)
    plt.savefig(save, dpi = 300)
    plt.show()

airasiafinal_pos = input_df['finaltext'].loc[input_df['sentimen'] == 1]
airasiafinal_neg = input_df['finaltext'].loc[input_df['sentimen'] == 0]

a = str(airasiafinal_pos)
positif = re.sub(r"""","",a)
mask = np.array(Image.open("thumbsupfix.jpg"))
save = 'airasia_positif'
generate_wordcloud(positif, mask, 'ocean', save)

b = str(airasiafinal_neg)
negatif = re.sub(r"""","",b)
#negatif
mask = np.array(Image.open("thumbsdownfix.jpg"))
save = 'airasia_negatif'
generate_wordcloud(negatif, mask, 'rocket', save)
```

Lampiran 5. *Syntax Support Vector Machine* (SVM) Menggunakan Python

```
# IMPORT LIBRARY ================================
from sklearn.metrics import precision_score, recall_score,
accuracy_score, confusion_matrix
from sklearn.model_selection import cross_val_score, KFold
from sklearn.svm import SVC
import imblearn
from sklearn.metrics import roc_curve, auc
from imblearn.over_sampling import SMOTE
from sklearn import metrics
from imblearn.pipeline import make_pipeline, Pipeline

# MENDEFINISIKAN VARIABEL ======================
Y = pd.DataFrame.as_matrix(input_df['sentimen'])
Y1 = np.ravel(Y)
tfidf_new = TFIDF_.values
X_new, Y_new = tfidf_new, Y1

# SVM KERNEL LINIER ============================

# Mencari Parameter Optimum ========================
# Tanpa SMOTE
c_range = [0.01, 0.1, 1, 10, 100, 1000, 10000]
n = 10
kfold = KFold(n_splits = n, random_state = 2000, shuffle = True)

all_ = dict()
auc = dict()
conma = dict()

for i,c in enumerate(c_range):
    all_[i] = []
    auc[i] = []
    conma[i] = []
    svm = SVC(kernel = 'linear', C = c)
```

Lampiran 5. *Syntax Support Vector Machine* (SVM) Menggunakan Python (lanjutan)

```
for train, test in kfold.split(X_new, Y_new):
        (X_new[train], X_new[test])
        (Y_new[train], Y_new[test])
        svm.fit(X_new[train], Y_new[train])
         prediksi = svm.predict(X_new[test])
        conma[i].append((confusion_matrix(Y_new[test],
prediksi)).astype(float))
for i in range(len(c_range)):
    all_[i] = []
    auc[i] = []
    for j in range(n):
        all_[i].append(sum(sum(conma[i][j])))
        auc[i].append(0.5*((conma[i][j][1,1]/(conma[i][j][1,0] +
conma[i][j][1,1])) + (conma[i][j][0,0]/(conma[i][j][0,0] +
conma[i][j][0,1]))))
auc1 = np.empty(len(c_range))
for i in range(len(c_range)):
    auc1[i] = np.mean(auc[i])
print(auc1)

# SVM KERNEL RBF ==================================

# Mencari Parameter Optimum ========================
# SMOTE
c_range = [0.01, 0.1, 1, 10, 100, 1000, 10000]
n = 10
kfold = KFold(n_splits = n, random_state = 2000, shuffle = True)
smote = SMOTE(random_state = 123)
all_ = dict()
auc = dict()
conma = dict()
for i,c in enumerate(c_range):
    all_[i] = []
    auc[i] = []
    conma[i] = []
    svm = SVC(kernel = 'rbf', C = c, gamma = 2**(1))
```

Lampiran 5. *Syntax Support Vector Machine* (SVM) Menggunakan Python (lanjutan)

```
    for train, test in kfold.split(X_new, Y_new):
        (X_new[train], X_new[test])
        (Y_new[train], Y_new[test])
        X_trainsmote, Y_trainsmote = smote.fit_sample(X_new[train],
Y_new[train])
        svm.fit(X_trainsmote, Y_trainsmote)
        prediksi = svm.predict(X_new[test])
        conma[i].append((confusion_matrix(Y_new[test],
prediksi)).astype(float))

for i in range(len(c_range)):
    all_[i] = []
    auc[i] = []
    for j in range(n):
        all_[i].append(sum(sum(conma[i][j])))
        auc[i].append(0.5*((conma[i][j][1,1]/(conma[i][j][1,0] +
conma[i][j][1,1])) + (conma[i][j][0,0]/(conma[i][j][0,0] +
conma[i][j][0,1]))))

auc1 = np.empty(len(c_range))
for i in range(len(c_range)):
    auc1[i] = np.mean(auc[i]

print(auc1)

# Evaluasi Klasifikasi Parameter Optimum ==========
# Tanpa SMOTE
sv = SVC(kernel = 'rbf', C = 10, gamma = 2**(1))
kfold = KFold(n_splits = 10, random_state = 2000, shuffle = True)

for train, test in kfold.split(X_new, Y_new):
    (X_new[train], X_new[test])
    (Y_new[train], Y_new[test])
```

Lampiran 5. *Syntax Support Vector Machine* (SVM) Menggunakan Python (lanjutan)

```
    #training
    svm_ = sv.fit(X_new[train], Y_new[train])
    prediksi_ = svm_.predict(X_new[train])
    fpr_,tpr_,__ = metrics.roc_curve(Y_new[train], prediksi_)
    auc__ = metrics.auc(fpr_,tpr_)

    #testing
    svm = sv.fit(X_new[train], Y_new[train])
    prediksi = svm.predict(X_new[test])
    fpr,tpr,_ = metrics.roc_curve(Y_new[test], prediksi)
    auc_ = metrics.auc(fpr,tpr)

    print(confusion_matrix(Y_new[train], prediksi_))
    print("AUC Tr Score: ", auc__)
    print(prediksi.astype(int))
    print(confusion_matrix(Y_new[test], prediksi))
    print("AUC Score: ", auc_)
    print("Akurasi: ", (accuracy_score(Y_new[test], prediksi)))
    print("Presisi: ", (precision_score(Y_new[test], prediksi)))
    print("Recall: ", (recall_score(Y_new[test], prediksi)))

# SMOTE
smote = SMOTE(random_state = 123)

sv = SVC(kernel = 'rbf', C = 100, gamma = 2**(-9))
kfold = KFold(n_splits = 10, random_state = 2000, shuffle = True)

for train, test in kfold.split(X_new, Y_new):
    (X_new[train], X_new[test])
    (Y_new[train], Y_new[test])
    X_trainsmote, Y_trainsmote =
smote.fit_sample(X_new[train],Y_new[train])
```

Lampiran 5. *Syntax Support Vector Machine* (SVM) Menggunakan Python (lanjutan)

```
#training
svm_ = sv.fit(X_trainsmote, Y_trainsmote)
prediksi_ = svm_.predict(X_trainsmote)
fpr_,tpr_,__ = metrics.roc_curve(Y_trainsmote, prediksi_)
auc__ = metrics.auc(fpr_,tpr_)

#testing
svm = sv.fit(X_trainsmote, Y_trainsmote)
prediksi = svm.predict(X_new[test])
fpr,tpr,_ = metrics.roc_curve(Y_new[test], prediksi)
auc_ = metrics.auc(fpr,tpr)

print(confusion_matrix(Y_trainsmote, prediksi_))
print("AUC Tr Score: ", auc__)
print(prediksi.astype(int))
print(confusion_matrix(Y_new[test], prediksi))
print("AUC Score: ", auc_)
print("Akurasi: ", (accuracy_score(Y_new[test], prediksi)))
print("Presisi: ", (precision_score(Y_new[test], prediksi)))
print("Recall: ", (recall_score(Y_new[test], prediksi)))
```

Lampiran 6. Rata-rata AUC untuk Pencarian Parameter γ Terbaik pada *kernel* RBF AirAsia

C	γ									
	2^{-15}	2^{-13}	2^{-11}	2^{-9}	2^{-7}	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}
	Unigram									
10^{-2}	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
10^{-1}	0,50	0,50	0,50	0,50	0,50	0,51	0,63	0,71	0,62	0,59
10^{0}	0,50	0,60	0,50	0,60	0,50	0,75	0,84	0,86	0,85	0,66
10^{1}	0,50	0,74	0,60	0,74	0,69	0,85	0,84	0,85	0,86	0,66
10^{2}	0,52	0,76	0,80	0,76	0,84	0,84	0,84	0,85	0,86	0,66
10^{3}	0,76	0,76	0,85	0,76	0,84	0,82	0,83	0,85	0,86	0,66
10^{4}	0,84	0,76	0,82	0,76	0,82	0,82	0,83	0,85	0,86	0,66

Lampiran 6. Rata-rata AUC untuk Pencarian Parameter γ Terbaik pada *kernel* RBF (lanjutan)

C	γ									
	2^{-15}	2^{-13}	2^{-11}	2^{-9}	2^{-7}	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}
Bigram										
10^{-2}	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
10^{-1}	0,50	0,50	0,50	0,50	0,50	0,50	0,61	0,63	0,63	0,61
10^{0}	0,50	0,50	0,50	0,50	0,60	0,65	0,70	0,76	0,76	0,67
10^{1}	0,50	0,50	0,57	0,63	0,69	0,73	0,78	0,78	0,77	0,67
10^{2}	0,50	0,61	0,67	0,72	0,76	0,77	0,77	0,78	0,77	0,67
10^{3}	0,66	0,71	0,75	0,77	0,76	0,77	0,77	0,78	0,77	0,67
10^{4}	0,74	0,75	0,74	0,74	0,77	0,76	0,77	0,78	0,50	0,67
Trigram										
10^{-2}	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
10^{-1}	0,50	0,50	0,50	0,50	0,50	0,50	0,52	0,57	0,61	0,54
10^{0}	0,50	0,50	0,50	0,50	0,50	0,57	0,60	0,63	0,63	0,60
10^{1}	0,50	0,50	0,50	0,54	0,60	0,62	0,63	0,63	0,63	0,60
10^{2}	0,50	0,52	0,59	0,61	0,62	0,63	0,63	0,63	0,63	0,60
10^{3}	0,58	0,60	0,62	0,62	0,62	0,63	0,63	0,63	0,63	0,60
10^{4}	0,61	0,62	0,62	0,62	0,62	0,62	0,63	0,63	0,63	0,60

LionAir (SMOTE)

C	γ									
	2^{-15}	2^{-13}	2^{-11}	2^{-9}	2^{-7}	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}
Unigram										
10^{-2}	0,60	0,60	0,60	0,60	0,60	0,60	0,61	0,53	0,51	0,51
10^{-1}	0,60	0,60	0,60	0,60	0,60	0,60	0,61	0,61	0,58	0,56
10^{0}	0,60	0,60	0,60	0,60	0,60	0,76	0,71	0,70	0,63	0,59
10^{1}	0,60	0,60	0,60	0,67	0,77	0,78	0,75	0,72	0,63	0,59
10^{2}	0,60	0,61	0,78	0,80	0,77	0,74	0,75	0,73	0,63	0,59
10^{3}	0,74	0,79	0,79	0,77	0,75	0,75	0,75	0,73	0,63	0,59
10^{4}	0,80	0,76	0,75	0,75	0,74	0,74	0,75	0,73	0,63	0,59

Lampiran 6. Rata-rata AUC untuk Pencarian Parameter γ Terbaik pada *kernel* RBF (lanjutan)

C	γ									
	2^{-15}	2^{-13}	2^{-11}	2^{-9}	2^{-7}	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}
Bigram										
10^{-2}	0,55	0,55	0,55	0,55	0,55	0,55	0,59	0,64	0,61	0,61
10^{-1}	0,55	0,55	0,55	0,55	0,55	0,55	0,59	0,65	0,65	0,65
10^{0}	0,55	0,55	0,55	0,55	0,55	0,56	0,68	0,69	0,68	0,67
10^{1}	0,55	0,55	0,55	0,55	0,60	0,71	0,71	0,70	0,68	0,67
10^{2}	0,55	0,54	0,57	0,55	0,71	0,71	0,70	0,70	0,68	0,67
10^{3}	0,57	0,61	0,72	0,66	0,71	0,71	0,70	0,70	0,68	0,67
10^{4}	0,72	0,72	0,71	0,73	0,71	0,71	0,70	0,69	0,67	0,67
Trigram										
10^{-2}	0,51	0,51	0,51	0,51	0,51	0,52	0,57	0,56	0,56	0,56
10^{-1}	0,51	0,51	0,51	0,51	0,51	0,52	0,57	0,56	0,56	0,56
10^{0}	0,51	0,51	0,51	0,51	0,51	0,52	0,57	0,57	0,56	0,56
10^{1}	0,51	0,51	0,51	0,51	0,51	0,57	0,57	0,57	0,56	0,56
10^{2}	0,51	0,51	0,51	0,52	0,56	0,57	0,57	0,57	0,56	0,56
10^{3}	0,51	0,51	0,56	0,55	0,56	0,57	0,57	0,57	0,56	0,56
10^{4}	0,56	0,55	0,55	0,55	0,56	0,57	0,57	0,57	0,56	0,56

Lampiran 7. *Confusion Matrix* dengan Parameter Optimum untuk AirAsia

	Kelas Sebenarnya	**Kelas Prediksi**	
		Negatif	Positif
Fold ke-1	Negatif	80	4
	Positif	15	46
Fold ke-2	Negatif	66	4
	Positif	11	64
Fold ke-3	Negatif	70	6
	Positif	10	59
Fold ke-4	Negatif	58	5
	Positif	11	71
Fold ke-5	Negatif	73	6
	Positif	22	44
Fold ke-6	Negatif	62	7
	Positif	17	59

Lampiran 7. *Confusion Matrix* dengan Parameter Optimum untuk AirAsia (lanjutan)

	Kelas Sebenarnya	**Kelas Prediksi**	
		Negatif	Positif
Fold ke-7	Negatif	66	7
	Positif	13	58
Fold ke-8	Negatif	64	5
	Positif	20	55
Fold ke-9	Negatif	65	5
	Positif	14	60
Fold ke-10	Negatif	79	3
	Positif	11	51

Lampiran 8. *Confusion Matrix* dengan Parameter Optimum untuk LionAir

	Kelas Sebenarnya	**Kelas Prediksi**	
		Negatif	Positif
Fold ke-1	Negatif	117	12
	Positif	5	10
Fold ke-2	Negatif	111	18
	Positif	4	11
Fold ke-3	Negatif	115	15
	Positif	1	12
Fold ke-4	Negatif	112	20
	Positif	4	7
Fold ke-5	Negatif	116	14
	Positif	5	8
Fold ke-6	Negatif	114	16
	Positif	2	11
Fold ke-7	Negatif	123	13
	Positif	4	3
Fold ke-8	Negatif	123	4
	Positif	5	11
Fold ke-9	Negatif	111	22
	Positif	3	7
Fold ke-10	Negatif	107	24
	Positif	1	11

Lampiran 9. *Output* Persamaan *Hyperplane* untuk AirAsia Menggunakan WEKA

```
SMO
Kernel used:
  RBF Kernel: K(x,y) = exp(-2.0*(x-y)^2)

Classifier for classes: 0, 1
BinarySMO
 -       0.3918 * < 0 0.634291 0 ... 0 0> * X]
 -       0.0791 * < 0 0 0 0 ... 0 0> * X]
 +        0.9881 * < 0 0 0 0 ... 0 0> * X]
...
...
 +       0.7563 * < 0 0 0 0 ... 0 0> * X]
 -       0.2202 * < 0 0 0 0 ... 0 0> * X]
 -       0.3765

Number of support vectors: 1114
```

Lampiran 10. *Output* Persamaan *Hyperplane* untuk LionAir Menggunakan WEKA

```
SMO
Kernel used:
  RBF Kernel: K(x,y) = exp(-0.001953125*(x-y)^2)

Classifier for classes: 0, 1
BinarySMO

 -     100       * < 0 0 0 0 ... 0 0> * X]
 +     100       * < 0 0 0 0 ... 0 0> * X]
 -     22.8606 * < 0 0 0 0 ... 0 0> * X]
...
...
 -    100       * < 0 0 0 0 ... 0.42196 0 0 0> * X]
 -     74.4496 * < 0 0 0 0 ... 0 0> * X]
 -     81.2941

Number of support vectors: 1355
```

Lampiran 11. Frekuensi Kata AirAsia

Februari			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
terima	4	tiket	12
kasih	4	kenapa	9
tiket	3	refund	5
penerbangan	2	error	5
...	...	...	...
senang	1	lama	2
...	...	...	...
mendunia	1	susah	1
Maret			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
gratis	153	tiket	156
kursi	133	aplikasi	84
tiket	73	ota	84
ikutan	68	konsumen	65
...	...	...	...
merapat	14	susah	18
...	...	...	...
sikat	1	bodoh	1
April			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
rute	13	penerbangan	17
buka	8	tiket	16
kasih	8	uang	14
terima	7	refund	14
...	...	...	...
asyik	3	dibatalkan	4
...	...	...	...
terbaik	1	frustasi	1

Lampiran 11. Frekuensi Kata AirAsia (lanjutan)

Mei			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
rute	18	pilot	9
buka	16	tiket	7
dong	16	bayar	6
semoga	7	harga	6
...	...	...	...
sukses	3	kapok	2
...	...	...	...
mantap	1	rampok	1

Lampiran 12. Frekuensi Kata LionAir

Februari			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
boarding	3	jam	19
pass	2	pesawat	17
untung	2	tiket	16
baik	2	bandara	14
...	...	...	...
asik	1	avtur	4
...	...	...	...
selamat	1	sibuk	1
Maret			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
lion	13	lion	159
pesawat	6	bagasi	83
terima	6	koper	67
kasih	6	naik	64
...	...	...	...
semoga	4	hilang	9
...	...	...	...
bertahan	1	kampret	1

Lampiran 12. Frekuensi Kata LionAir (lanjutan)

April			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
solo	13	lion	41
lion	8	tiket	32
rute	5	maskapai	23
terima	5	harga	23
...	...	...	...
pontianak	2	rugi	5
...	...	...	...
jogja	1	bobrok	1
Mei			
Positif	**Frekuensi**	**Negatif**	**Frekuensi**
dukung	7	pilot	200
mantap	6	lion	76
semoga	3	pecat	41
suka	3	anjing	40
...	...	...	...
ayo	2	hukum	23
...	...	...	...
gagah	1	anarkis	1

Lampiran 13. Surat Keterangan Pengambilan Data

SURAT PERNYATAAN

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

Nama : Cahya Buana Putri
NRP : 062115 4000 0112

menyatakan bahwa data yang digunakan dalam Tugas Akhir ini merupakan data sekunder yang diambil dari ~~penelitian / buku / Tugas Akhir / Thesis / Publikasi /~~ lainnya yaitu :

Sumber : Twitter API (*Application Program Interface*)
Keterangan : Data *tweet* dengan *keyword* "@AirAsia_indo" dan "@lionairgroup" dari 21 Februari 2019 hingga 7 Mei 2019

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

Surabaya, Mei 2019

Mengetahui,
Pembimbing Tugas Akhir

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

Mahasiswa

Cahya Buana Putri
NRP. 062115 4000 0112

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Penulis dilahirkan di Malang, 2 Juli 1997 dengan nama lengkap Cahya Buana Putri, biasa dipanggil Icha. Penulis menempuh pendidikan formal di SD Negeri Pucang IV Sidoarjo, SMP Negeri 1 Sidoarjo, dan SMA Negeri 1 Sidoarjo. Kemudian penulis diterima sebagai mahasiswa Departemen Statistika ITS pada tahun 2015. Selama masa perkuliahan, penulis aktif di Himpunan Mahasiswa Statistika ITS (HIMASTA-ITS) periode 2016-2017 sebagai Staf Departemen Media Informasi dan pada periode 2017-2018 sebagai Ketua Biro Kampanye Kreatif Departemen Komunikasi dan Informasi. Selain itu, penulis juga berkesempatan untuk berkontribusi di tingkat Institut sebagai Mentor di GERIGI (Generasi Integralistik) ITS 2017. Pada tahun 2018, penulis berkesempatan melakukan Kerja Praktik di PT Telekomunikasi Indonesia yang berlokasi di Gambir, Jakarta Pusat di unit *Data Scientist* – Divisi *Digital Service* selama satu bulan. Bagi pembaca yang ingin berdiskusi, memberikan saran, dan kritik mengenai Tugas Akhir ini dapat disampaikan melalui email cahyabuana27@gmail.com.