TUGAS AKHIR – SS141501

ANALISIS KOMPETITIF SOSIAL MEDIA MENGGUNAKAN METODE KLASIFIKASI *NAÏVE BAYES CLASSIFIER* (NBC) DAN *SUPPORT VECTOR MACHINE* (SVM): STUDI KASUS DI INDUSTRI TRANSPORTASI UMUM TAKSI

PUTRI AYU SEKAR KARIMAH
NRP 062116 4500 0009

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

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

TUGAS AKHIR – SS141501

ANALISIS KOMPETITIF SOSIAL MEDIA MENGGUNAKAN METODE KLASIFIKASI *NAÏVE BAYES CLASSIFIER* (NBC) DAN *SUPPORT VECTOR MACHINE* (SVM): STUDI KASUS DI INDUSTRI TRANSPORTASI UMUM TAKSI

PUTRI AYU SEKAR KARIMAH
NRP 062116 4500 0009

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

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

FINAL PROJECT – SS141501

SOCIAL MEDIA COMPETITIVE ANALYSIS USING CLASSIFICATION METHODS *NAÏVE BAYES CLASSIFIER* (NBC) AND *SUPPORT VECTOR MACHINE* (SVM): CASE STUDY ON PUBLIC TAXI TRANSPORTATION INDUSTRY

PUTRI AYU SEKAR KARIMAH
SN 062116 4500 0009

Supervisors
Dr. Dra. Kartika Fithriasari, M.Si

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

LEMBAR PENGESAHAN

ANALISIS KOMPETITIF SOSIAL MEDIA MENGGUNAKAN METODE KLASIFIKASI *NAÏVE BAYES CLASSIFIER* (NBC) DAN *SUPPORT VECTOR MACHINE* (SVM): STUDI KASUS DI INDUSTRI TRANSPORTASI UMUM TAKSI

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

Oleh :
Putri Ayu Sekar Karimah
NRP. 062116 4500 0009

Disetujui oleh Pembimbing:

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

Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2018

ANALISIS KOMPETITIF SOSIAL MEDIA MENGGUNAKAN METODE KLASIFIKASI *NAÏVE BAYES CLASSIFIER* (NBC) DAN *SUPPORT VECTOR MACHINE* (SVM): STUDI KASUS DI INDUSTRI TRANSPORTASI UMUM TAKSI

Nama Mahasiswa	**: Putri Ayu Sekar Karimah**
NRP	**: 062116 4500 0009**
Departemen	**: Statistika FMKSD ITS**
Dosen Pembimbing	**: Dr. Dra. Kartika Fithriasari, M.Si**

Abstrak

Taksi merupakan salah satu angkutan umum yang cukup berbeda dibandingkan dengan angkutan umum lainnya. Adapun fenomena baru yang terjadi pada angkutan umum moda ini yaitu fenomena taksi berbasis online. Sehingga persaingan antara taksi konvensional dan taksi berbasis online tersebut kerap mendapat berbagai tanggapan publik di media sosial. Penelitian ini dilakukan untuk mengetahui tanggapan masyarakat berdasarkan sentimen di media sosial (khususnya Twitter) terhadap pelayanan atau isu yang sedang beredar tentang taksi konvensional dan taksi berbasis online tersebut. Data yang digunakan pada penelitian ini merupakan data dari hasil crawling dari tiga keyword yaitu BlueBird Taxi, GoCar dan GrabCar dengan menggunakan Twitter API (Application Programming Interface). Dari hasil analisis didapatkan bahwa kepuasan pengguna terhadap jasa yang diberikan oleh taksi berbasis online lebih baik jika dibandingkan dengan taksi konvensional. Kemudian dari hasil klasifikasi dengan metode Naïve Bayes Classifier dan Support Vector Machine didapatkan metode yang sesuai adalah metode SVM Kernel RBF dengan rata-rata nilai accuracy pda data testing untuk BlueBird Taxi sebesar 81.6%, untuk GoCar sebesar 77.6%, dan untuk GrabCar sebesar 78.5%. Sedangkan dari hasil Social Network Analysis didapatkan account @Bluebirdgroup merupakan account yang memiliki interaksi

tertinggi dan merupakan account penghubung atau jembatan dari seluruh aliran informasi.

Kata Kunci : Naïve Bayes Classifier, Social Network Analysis, Support Vector Machine, Twitter.

SOCIAL MEDIA COMPETITIVE ANALYSIS USING CLASSIFICATION METHODS *NAÏVE BAYES CLASSIFIER* (NBC) DAN *SUPPORT VECTOR MACHINE* (SVM): CASE STUDY ON PUBLIC TAXI TRANSPORTATION INDUSTRY

Student Name	**: Putri Ayu Sekar Karimah**
Student Number	**: 062116 4500 0009**
Department	**: Statistics**
Supervisors	**: Dr. Dra. Kartika Fithriasari, M.Si**

Abstract

Taxi is one of public transportation that has few differences among the others. Online-based transportation is a new phenomenon in this transportation mode so the competition between conventional one and online-based one frequently get some responses from social media user. This research's goal is knowing people's responses about service or common issue of conventional taxi and online-based taxi based on social media sentiment especially on Twitter. Data resource is from crawling process of three keywords: BlueBird Taxi, GoCar and GrabCar using Twitter API (Application Programming Interface). The result of this research is service satisfaction of online-based taxi is better than the conventional one. Afterwards, classification using Naïve Bayes Classifier and Support Vector Machine obtained suitable method for this research is SVM Kernel RBF with average accuracy on testing data for BlueBird Taxi is for about 81.6%, GoCar is for about 77.6%, and GrabCar is for about 78.5%. Furthermore, Social Network Analysis obtained that @Bluebirdgroup account has highest interaction and connector or from all information flow.

Keywords : Naïve Bayes Classifier, Social Network Analysis, Support Vector Machine, Twitter.

(This page intentionally left blank)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat rahmat, karunia, taufik, dan hidayah-Nya yang tidak pernah berhenti sehingga penulis dapat menyelesaikan penyusunan laporan Tugas Akhir yang berjudul **"Analisis Kompetitif Sosial Media Menggunakan Metode Klasifikasi *Naïve Bayes Classifier* (NBC) Dan *Support Vector Machine* (SVM): Studi Kasus Di Industri Transportasi Umum Taksi"**.

Penulis menyadari bahwa dalam penyusunan Tugas Akhir ini tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Dra. Kartika Fithriasari, M.Si selaku dosen pembimbing yang telah banyak memberikan bimbingan, pengarahan, dan motivasi sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Bapak Prof. Drs. Nur Iriawan, M.Ikom, Ph.D dan Ibu Irhamah, M.Si, Ph.D selaku dosen penguji Tugas Akhir yang telah memberikan banyak saran dan masukan, serta nasehat dalam penyusunan laporan Tugas Akhir ini.
3. Bapak Dr. Suhartono selaku Kepala Departemen Statistika FMKSD ITS yang telah memberikan banyak fasilitas yang menunjang kelancaran penyelesaian Tugas Akhir.
4. Bapak Dr. Sutikno, M.Si selaku Koordinator Program Studi S1 atas ketelatenannya dalam memberikan bantuan dan informasi yang diberikan selama ini.
5. Ibu Dr. Ismaini Zain, M.Si selaku dosen wali yang selalu memberikan pengarahan dan motivasi selama perkuliahan penulis.
6. Seluruh civitas akademika Departemen Statistika FMKSD ITS yang telah membantu dalam melancarkan Tugas Akhir ini.

7. Kedua orang tua dan keluarga yang selalu memberikan doa, kasih sayang, semangat, dan dukungan terbesar dalam hidup penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
8. Teman-teman Lintas Jalur angkatan 2016 yang telah menemani perjuangan selama dua tahun ini dan atas segala bantuan serta kerjasamanya sehingga memberikan kelasncaran bagi penulis dalam menyelesaikan studi di Departemen Statistika ITS.
9. Teman-teman dekat yang telah banyak memberikan dukungan, semangat, serta keceriaan sehingga penulis dapat menikmati segala proses di Departemen Statistika ITS dengan lancar dan sukses.
10. Semua pihak yang telah turut serta dalam membantu dan mendukung terselesainya Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.

Dengan berakhirnya pengerjaan laporan Tugas Akhir ini, penulis berharap laporan ini dapat memberikan manfaat kepada penulis pada khususnya dan pembaca pada umumnya. Penulis menyadari dalam penulisan laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, peneliti mengharap adanya perbaikan dalam penulisan laporan di masa mendatang.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i
TITLE PAGE ii
LEMBAR PENGESAHAN iii
ABSTRAK v
ABSTRACT vii
KATA PENGANTAR ix
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xv
DAFTAR LAMPIRAN xvii

BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Perumusan Masalah 8
1.3 Tujuan Penelitian 8
1.4 Manfaat Penelitian 9
1.5 Batasan Penelitian 9

BAB II TINJAUAN PUSTAKA
2.1 *Text Mining* 11
2.2 *Crawling* Data Twitter dengan Twitter API 12
2.3 *Sentiment Analysis* 13
2.4 Klasifikasi Teks 14
2.5 Praproses Teks 15
2.6 *Term Frequency Inverse Document Frequency* (TF-IDF) 17
2.7 *K-fold Cross Validation* 18
2.8 *Naïve Bayes Classifier* (NBC) 18
2.9 *Support Vector Machine* (SVM) 21
2.9.1 SVM pada *Linearly Separable Data* 22
2.9.2 SVM pada *Non Linearly Separable Data* dengan Metode Kernel 25
2.10 Ketepatan Klasifikasi 29
2.11 *Word Cloud* 31

2.12 *Social Network Analysis* (SNA) 32
2.13 Media Sosial 36
2.14 Transportasi Umum Taksi 37

BAB III METODOLOGI PENELITIAN
3.1 Sumber Data 39
3.2 Struktur Data 39
3.3 Langkah Analisis 41
3.4 Diagram Alir 43

BAB IV ANALISIS DAN PEMBAHASAN
4.1 Karakteristik Data 45
4.2 Praproses Teks 54
4.3 Klasifikasi Menggunakan *Naïve Bayes Classifier* (NBC) 57
4.4 Klasifikasi Menggunakan *Support Vector Machine* (SVM) 61
4.4.1 SVM Menggunakan Kernel Linear 62
4.4.2 SVM Menggunakan Kernel RBF 65
4.4.3 Pengukuran Performa SVM 68
4.5 Perbandingan Hasil Klasifikasi Antara Metode NBC dan SVM 70
4.6 Visualisasi *Word Cloud* 71
4.7 *Social Network Analysis* 74
4.7.1 *Social Network Analysis* BlueBird Taxi 75
4.7.2 *Social Network Analysis* GoCar 78
4.7.3 *Social Network Analysis* GrabCar 82
4.7.4 *Social Network Analysis* Taksi Konvensional dan Taksi Berbasis *Online* 85

BAB V PENUTUP
5.1 Kesimpulan 91
5.2 Saran 92

DAFTAR PUSTAKA 95
LAMPIRAN 101
BIODATA PENULIS 125

DAFTAR TABEL

Halaman

Tabel 2.1 Fungsi Kernel pada SVM 28
Tabel 2.2 *Confusion Matrix* 29
Tabel 2.3 Interpretasi Nilai AUC.....31
Tabel 3.1 Struktur Data Penelitian Sebelum *Pre-processing* .39
Tabel 3.2 Struktur Data Setelah *Pre-processing*.....40
Tabel 3.3 Struktur Data *Social Network Analysis*.....41
Tabel 4.1 Contoh Isi *Tweet* Tiap Kategori.....49
Tabel 4.2 Simulasi Praproses Teks 55
Tabel 4.3 Struktur Data Setelah Praproses Teks.....56
Tabel 4.4 Frekuensi Kata Kunci Untuk Masing-Masing Taksi 57
Tabel 4.5 Model Klasifikasi NBC 58
Tabel 4.6 Nilai *Accuracy* Metode NBC dengan 10-*fold CV*...59
Tabel 4.7 *Confusion Matrix* dengan Metode NBC 59
Tabel 4.8 Ketepatan Klasifikasi Metode NBC.....60
Tabel 4.9 Penentuan Parameter SVM Kernel Linear.....62
Tabel 4.10 Nilai *Accuracy* Metode SVM Kernel Linear dengan 10-*fold CV* 63
Tabel 4.11 *Confusion Matrix* dengan Metode SVM Kernel Linear.....64
Tabel 4.12 Ketepatan Klasifikasi Metode SVM Kernel Linear.....64
Tabel 4.13 Penentuan Parameter SVM Kernel RBF.....65
Tabel 4.14 Nilai *Accuracy* Metode SVM Kernel RBF dengan 10-*fold CV*.....66
Tabel 4.15 *Confusion Matrix* dengan Metode SVM Kernel RBF.....67
Tabel 4.16 Ketepatan Klasifikasi Metode SVM Kernel RBF...68
Tabel 4.17 Perbandingan Ketepatan Klasifikasi Antara SVM Kernel Linear dan Kernel RBF 69
Tabel 4.18 Persamaan *Hyperplane* pada Setiap Taksi.....70
Tabel 4.19 Perbandingan Metode NBC dan SVM.....71

Tabel 4.20 Data Perbandingan Jaringan BueBird Taxi..... 75
Tabel 4.21 Analisis *Centrality* dari Jaringan BlueBird Taxi.... 77
Tabel 4.22 Data Perbandingan Jaringan GoCar..... 79
Tabel 4.23 Analisis *Centrality* dari Jaringan GoCar..... 81
Tabel 4.24 Data Perbandingan Jaringan GrabCar..... 82
Tabel 4.25 Analisis *Centrality* dari Jaringan GrabCar..... 84
Tabel 4.26 Data Perbandingan Jaringan..... 85
Tabel 4.27 Analisis *Centrality* 88

DAFTAR GAMBAR

Halaman

Gambar 2.1 *OAuth Workflow* 13
Gambar 2.2 Ilustrasi Pembagian Data 18
Gambar 2.3 *Linearly Separable Data* 22
Gambar 2.4 *Non Linearly Separable Data* 25
Gambar 2.5 Visualisasi Data dengan *Word Cloud* 32
Gambar 2.6 Contoh *Sociogram* Sederhana 33
Gambar 3.1 Diagram Alir Penelitian 43
Gambar 4.1 Perbandingan Antar *Official Account* Twitter Tiap Taksi 45
Gambar 4.2 Perbandingan Banyak *Tweet* Dari Tiap Taksi ... 46
Gambar 4.3 Kategori *Tweet* Pengguna 47
Gambar 4.4 *Tweet* Sentimen 52
Gambar 4.5 Perbandingan *Tweet* Sentimen Antar Jenis Taksi 53
Gambar 4.6 *Scatterplot* Variabel Ngobrol dan Variabel Mitra 61
Gambar 4.7 *Word Cloud* BlueBird Taxi Sentimen Positif (a) dan Sentimen Negatif (b) 72
Gambar 4.8 *Word Cloud* GoCar Sentimen Positif (a) dan Sentimen Negatif (b) 73
Gambar 4.9 *Word Cloud* GrabCar Sentimen positif (a) dan Sentimen Negatif (b) 74
Gambar 4.10 Visualisasi *Network* BlueBird Taxi 76
Gambar 4.11 Visualisasi *Network* GoCar 80
Gambar 4.12 Visualisasi *Network* GrabCar 83
Gambar 4.13 Visualisasi *Network* 87

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

Halaman

Lampiran 1. *Syntax Crawling* Data Menggunakan RStudio 101
Lampiran 2. Data *Tweet* BlueBird Taxi dari Twitter API 102
Lampiran 3. Data *Tweet* GoCar dari Twitter API 103
Lampiran 4. Data *Tweet* GrabCar dari Twitter API 104
Lampiran 5. *Syntax Input* dan Praproses Data Menggunakan Python 3.6 105
Lampiran 6. Frekuensi Kata Kunci 107
Lampiran 7. *Syntax Naïve Bayes Classifier* (NBC) Menggunakan Python 3.6 108
Lampiran 8. *Confusion Matrix* Metode NBC pada BlueBird Taxi 110
Lampiran 9. *Confusion Matrix* Metode NBC pada GoCar .. 110
Lampiran 10. *Confusion Matrix* Metode NBC pada GrabCar 111
Lampiran 11. *Syntax Support Vector Machine* (SVM) Menggunakan Python 3.6 112
Lampiran 12. *Confusion Matrix* Metode SVM Kernel Linear pada BlueBird Taxi 114
Lampiran 13. *Confusion Matrix* Metode SVM Kernel Linear pada GoCar..... 115
Lampiran 14. *Confusion Matrix* Metode SVM Kernel Linear pada GrabCar 115
Lampiran 15. *Confusion Matrix* Metode SVM Kernel RBF pada BlueBird Taxi..... 116
Lampiran 16. *Confusion Matrix* Metode SVM Kernel RBF pada GoCar 117
Lampiran 17. *Confusion Matrix* Metode SVM Kernel RBF pada GrabCar..... 117
Lampiran 18. *Output* Persamaan *Hyperplane* BlueBird Taxi..... 118
Lampiran 19. *Output* Persamaan *Hyperplane* GoCar 119

Lampiran 20. *Output* Persamaan *Hyperplane* GrabCar 119
Lampiran 21. *Syntax Word Cloud* Menggunakan Python 3.6 120
Lampiran 22. *Output Social Network Analysis* BlueBird Taxi 121
Lampiran 23. *Output Social Network Analysis* GoCar..... 121
Lampiran 24. *Output Social Network Analysis* GrabCar..... 122
Lampiran 25. *Output Social Network Analysis* Taksi Konvensional dan Taksi Berbasis *Online* 122
Lampiran 26. Surat Pernyataan Data..... 123

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dari masa ke masa sangat pesat dan mempunyai peran yang sangat penting dalam kehidupan masyarakat baik individu maupun kelompok organisasi. Teknologi informasi memegang peranan diberbagai sendi kehidupan manusia karena dapat menghubungkan dan menyajikan berbagai informasi melalui *web*. *Web* atau *website* merupakan halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses seluruh dunia selama terkoneksi dengan jaringan internet (Buntoro, Adji, & Purnamasari, 2014). Kehadiran internet telah membawa revolusi pada cara manusia melakukan komunikasi. Dengan penggunaan teknologi informasi ini sebagai sarana komunikasi memungkinkan setiap orang berkomunikasi dengan pihak lain yang terhubung dengan internet walaupun lokasi tempat tinggal mereka berjauhan. Menurut Kementrian Komunikasi dan Informatika Republik Indonesia (2017) mengungkapkan bahwa pengguna internet di Indonesia pada tahun 2016 mencapai 132,7 juta orang dari total penduduk Indonesia sebanyak 256,2 juta. Dari angka tersebut, 40 persennya menggunakan internet untuk mengakses jejaring sosial.

Media jejaring sosial memberikan peran yang sangat besar khususnya perkembangan teknologi dalam bidang komunikasi yang menjadi tak terbatas. Jejaring sosial merupakan sebuah media sosial dimana seseorang dapat menuangkan segala sesuatu yang ada dipikiran baik untuk berbagi informasi, berbagi pengalaman suka maupun duka, berbagi informasi promosi, berbagi informasi kegiatan dan lain sebagainya. Jejaring sosial sendiri merupakan sumber sebagai media sosial yang digunakan oleh orang-orang untuk berkomunikasi dengan para pengguna lainnya sebagai sarana bersosialisasi dengan orang lain dengan cara berinteraksi antara satu dengan lainnya sehingga manusia dapat mengeluarkan pendapat dan menyampaikan informasi serta dapat menerima

pendapat orang lain baik dengan cara langsung (tatap muka) maupun melalui media (Hendriyani, 2013).

Dari sekian banyak jejaring sosial di internet, salah satu jejaring sosial yang saat ini cukup menarik banyak perhatian para pengguna internet jejaring sosial adalah Twitter. Twitter merupakan jejaring sosial *microblogging* yang dipergunakan sebagai media komunikasi dan penyampaian informasi yang didalamnya memungkinkan terjadinya pertukaran informasi. Pertukaran informasi yang dilakukan melalui *sharing* dengan pengguna maupun penyebar informasi lainnya, sehingga Twitter dapat memberikan informasi dalam berbagai hal. Twitter memiliki jumlah pengguna yang cukup besar di Indonesia yaitu mencapai 19,5 juta pengguna dari total 330 juta pengguna di dunia. Pengguna Twitter di Indonesia menempati peringkat 5 pengguna Twitter terbesar di dunia dibawah USA, Brazil, Jepang, dan Inggris (Kemenkominfo, 2017). Sejak tahun 2010 hingga kuartal ketiga tahun 2016, pengguna Twitter mengalami pertumbuhan yang signifikan hingga mencapai 313 juta *account*.

Dengan tampilan yang mudah penggunaannya, berisi maksimal 140 karakter, masyarakat umum bisa menyebutkan dengan kata '*tweet*' atau kicauan. Kata yang terkandung dalam Twitter adalah bahasa alami manusia yang merupakan bahasa dengan struktur kompleks. Kebiasaan masyarakat mengutarakan pendapatnya melalui media sosial terutama Twitter dalam menanggapi kejadian atau hal-hal yang terjadi di lingkungannya dapat menjadi salah satu acuan untuk mengetahui sentimen masyarakat terhadap lingkungan atau kota tempat tinggal masyarakat tersebut berupa kritik atau saran. Salah satunya menanggapi berbagai macam sentimen masyarakat terhadap pelayanan yang diberikan oleh penyedia jasa transportasi, misalnya saja seperti di industri transportasi umum taksi.

Fenomena transportasi umum berbasis aplikasi dinilai banyak pihak telah merubah tatanan dan sistem komunikasi yang berlangsung pada sektor transportasi umum. Modernisasi sistem komunikasi transportasi berlangsung sejalan dengan kebutuhan

konsumen masa kini yang lebih cepat, mudah, dan murah. Saat ini konsumen semakin memilih untuk menggunakan jasa transportasi melalui sebuah perangkat lunak pada telepon pintarnya yang disebut aplikasi transportasi. Pada tahun 2014 hingga tahun 2016 berbagai perusahaan besar transportasi berbasis aplikasi muncul di Indonesia yaitu diantaranya Go-Jek, Grab, dan Uber.

Pemesanan jasa moda transportasi umum khususnya berupa taksi yang dimanfaatkan tiga perusahaan tersebut dikenal di Indonesia dengan sebutan taksi *online*. Pemesanan secara *online* melalui aplikasi menjadi cara baru dalam komunikasi antara pengemudi dan konsumen untuk pemesanan jasa taksi. Dengan bantuan internet dan melalui aplikasi ini konsumen lebih dimudahkan karena tidak harus dilakukan dengan cara menyetop ataupun telepon ke *operator*. Konsumen dapat memesan taksi terdekat dari manapun serta informasi mengenai pengemudi dan harga dapat langsung diketahui oleh konsumen. Selain itu, taksi berbasis *online* juga sering memberikan promosi khusus pada konsumennya. Media promosi pun dilakukan melalui layanan *website*, serta sosial media seperti Facebook, Twitter, Instagram, dan lainnya. Sehingga pertukaran informasi dan komunikasi tersebut dapat tumbuh semakin besar.

Pertumbuhan taksi *online* yang begitu cepat membuat perusahaan taksi plat kuning yang selama ini telah beroperasi merasa tersaingi dan semakin kebingungan. Perusahaan taksi ini kini disebut konsumen sebagai taksi konvensional. Berbagai perusahaan taksi konvensional di Indonesia melakukan protes kepada pemerintah untuk melarang beroperasinya taksi berbasis *online*. Pemerintah Indonesia melalui menteri perhubungan serta menteri komunikasi dan informasi telah berkomunikasi untuk menyikapi permasalahan ini. Hingga diterapkannya peraturan yang mengatur pelaksanaan taksi berbasis *online* yaitu Peraturan Menteri Perhubungan Nomor 108 Tahun 2017 tentang Penyelenggaraan Angkutan Orang dengan Kendaraan Bermotor Tidak Dalam Trayek yang telah berlaku sejak tanggal 1 Februari 2018.

Beberapa perusahaan taksi konvensional di Indonesia kini juga ada yang telah berinovasi. Diantaranya ialah Perusahaan *BlueBird Taxi* yang telah lebih dulu menggunakan aplikasi taksi yaitu 'Aplikasi Taxi Mobile Reservation' yang kini berubah nama menjadi aplikasi 'My Blue Bird'. Bahkan sejak tanggal 1 Februari 2017, PT Blue Bird Tbk telah resmi menjalin kerja sama dengan Go-Jek. Hal tersebut merupakan sebuah peningkatan untuk perusahaan taksi konvensional dapat bersaing dengan penyedia pelayanan taksi *online* di era digital sekarang ini. Sehingga persaingan antara taksi konvensional dan taksi berbasis *online* tersebut juga sering mendapat berbagai tanggapan publik di media sosial. Sehingga perlu dilakukan penelitian tentang analisis kompetitif sosial media (khususnya Twitter) yang bertujuan untuk mengetahui tanggapan masyarakat berdasarkan sentimen di media sosial (Twitter) terhadap pelayanan atau isu yang sedang beredar tentang taksi konvensional dan taksi berbasis *online* tersebut.

Pengguna Twitter dapat mengemukakan pendapatnya terhadap suatu produk dan mengomentari suatu pelayanan melalui *tweet*. *Tweet* pada setiap pengguna Twitter dapat berpengaruh dalam pembentukan citra suatu produk atau jasa, karena semakin banyak suatu topik tertentu diulas dalam *tweet* pengguna maka topik tersebut dapat menjadi *trending topic* di Twitter. Keberadaan Twitter telah digunakan secara luas dari lapisan masyarakat dan dapat dilihat sebagai sebuah refleksi yang baik dimana keberadaan Twitter dapat merepresentasikan apa yang sedang menjadi *trend* pembicaraan dan hal apa yang sedang menarik untuk dibahas. Kebiasaan masyarakat mem-*posting tweet* dalam menilai sebuah pelayanan dapat menjadi acuan untuk mengetahui sentimen terhadap penyedia jasa. Twitter telah menyediakan *Application Programming Interface* (API) yaitu sekumpulan fungsi atau protokol yang disediakan untuk pengguna dalam rangka mengembangkan sebuah aplikasi (Blanchette, 2008). Twitter API memungkinkan pengguna untuk mengakses dan mendapatkan informasi mengenai *tweet*, profil pengguna, data *follower*, dan lainnya. Hal tersebut menjadikan Twitter sebagai *microblog* yang

banyak diminati perusahaan, organisasi, maupun individu dalam mendapatkan opini publik mengenai suatu topik tertentu.

Pada dasarnya, setiap proses utama akan melakukan tahapan secara umum yaitu ketika peneliti ingin menentukan orientasi sentimen masyrakat dari Twitter. Analisis sentimen atau *opinion mining* merupakan metode analisis berbasis komputasi mengenai pendapat, sentimen, dan emosi (Liu, 2010). Analisis sentimen digunakan untuk melihat kecenderungan suatu sentimen atau pendapat, apakah pendapat tersebut cenderung beropini positif atau negatif. Peneliti akan melakukan *input* berupa teks Twitter atau yang biasa disebut dengan *tweet*. Kemudian akan dilakukan seleksi *tweet* yang telah di *input* peneliti sebagai bagian dari proses penentuan sentimen yang mengandung *tweet* positif atau negatif secara manual. Sebelum *tweet* dapat dikategorikan, maka data *tweet* tersebut sebaiknya diproses terlebih dahulu. Bentuk data yang tersedia pada *tweet* adalah berupa teks. Apabila dibandingkan dengan jenis data yang lain, sifat data berbentuk teks tidak terstruktur dan sulit ditangani. *Text mining* adalah cara agar teks dapat diolah dengan menggunakan komputer untuk menghasilkan analisis yang bermanfaat (Witten, Frank, & Hall, 2011). Sehingga setelah data diolah sedemikian rupa melalui *text processing* data, maka data telah siap untuk diolah lebih lanjut untuk klasifikasi sentimen dengan menggunakan metode klasifikasi teks.

Terdapat banyak metode klasifikasi dalam ilmu statsitika yang digunakan untuk analisis sentimen, namun metode yang sering digunakan diantaranya adalah *Naïve Bayes Classifier* (NBC), *K-Nearest Neighbour*, dan *Support Vector Machines* (SVM). Namun pada penelitian ini akan menggunakan metode NBC dan SVM. Metode NBC telah banyak digunakan dalam penelitian mengenai *text mining*, beberapa kelebihan NBC diantaranya adalah salah satu algoritma klasifikasi yang sederhana namun memiliki akurasi yang tinggi (Rish, 2001). Sedangkan pemilihan metode SVM karena kemampuan generalisasi dalam mengklasifikasikan suatu *pattern* atau pola. Teknik ini berakar pada teori pembelajaran statistik dan telah menunjukkan hasil

empiris yang menjanjikan dalam berbagai aplikasi praktis dari pengenalan digit tulisan tangan sampai kategorisasi teks. SVM juga bekerja sangat baik pada data dengan berbagai banyak dimensi dan menghindari kesulitan dari permasalahan dimensionalitas (Tan, Stenbach, & Kumar, 2006).

Penelitian yang berkaitan dengan metode NBC dan SVM untuk kasus data Twitter telah dilakukan oleh Aliandu (2013) yaitu tentang *sentiment analysis* dengan judul *Twitter Used by Indonesian President: An Sentiment Analysis of Timeline*. Penelitian tersebut membahas hasil analisis sentimen pada akun Twitter milik Presiden Indonesia dengan hasil akurasi yang didapatkan sebesar 79,42%, akan tetapi masih terdapat kekurangan dalam memisahkan data *text* yang tidak mengandung sentimen. Analisis serupa juga dilakukan oleh Sunni dan Widyantoro (2012) yang mengulas analisis sentimen menggunakan algoritma *Naïve Bayes Classifier* dengan penggabungan 10 metode praproses dan melihat karakter topik yang muncul menggunakan metode *tf-idf* dan didapatkan tingkat akurasi antara 69,4% hingga 72,8% dengan jumlah data sebanyak 2000 *tweet*. Penelitian dengan metode serupa pernah dilakukan oleh Ariadi & Fithriasari (2015) tentang Klasifikasi Berita Indonesia menggunakan NBC dan SVM dengan *Confix-Stripping Stemmer*. Penelitian tersebut menyimpulkan bahwa SVM kernel linier dan kernel RBF menghasilkan ketepatan klasifikasi yang sama dan bila dibandingkan dengan NBC maka SVM lebih baik. Selain itu, penelitian yang dilakukan oleh Widhianingsih & Fithriasari (2016) untuk aplikasi *text mining* pada automasi klasifikasi artikel dalam majalah online wanita dengan menggunakan metode *Naïve Bayes Classifier* (NBC), *Artificial Neural Network* (ANN), dan Regresi Logistik Multinomial didapatkan bahwa tingkat akurasi model NBC adalah sebesar 80,71%, model ANN sebesar 75%, dan Regresi Logistik Multinomial sebesar 57,86%. Sehingga didapatkan metode NBC memiliki performansi yang paling baik. Kemudian penelitian yang dilakukan oleh Asiyah & Fithriasari (2016) yang membandingkan metode *Support Vector Machine* (SVM) dan *K-Nearest Neighbor*

(KNN) mendapatkan hasil bahwa apabila dibandingkan dengan KNN, maka SVM lebih baik daripada KNN dengan hasil nilai *akurasi*, *recall*, *precision*, dan *F-Measure* sebesar 93,2%, 93,2%, 93,63%, dan 93,14%.

Selain itu juga terdapat beberapa penelitian sebelumnya yang berkaitan dengan metode *Social Network Analysis* (SNA) untuk kasus data Twitter telah dilakukan oleh Nuansa (2017) tentang analisis sentimen pengguna Twitter terhadap pemilihan Gubernur DKI Jakarta dan didapatkan hasil yang tinggi untuk *degree centrality* sebesar 0.865 yang menunjukkan pengaruh antar node kata kunci dengan kata kunci yang lain. Penelitian serupa juga dilakukan oleh Oktora dan Alamsyah (2014) mengenai pola interaksi dan aktor yang paling berperan pada Event JGTC 2013 melalui media sosial Twitter didapatkan bahwa terdapat 7624 *node* (akun) yang terlibat dengan 7445 *edge* (interaksi) yang terjadi di *network* tesebut. Penelitian lainnya yang menerapkan metode SNA dilakukan oleh Sucipto dan Alamsyah (2016) untuk mengetahui persepsi kualitas merek pada konten percakapan di media sosial Twitter untuk studi kasus PT Indosat Tbk. dan PT Telkomsel diperoleh hasil yaitu dalam jaringan kualitas merek Indosat terdapat 107 *node* dan 264 *edges* dan pada jaringan kualitas merek Telkomsel terdapat 65 *node* dan 1133 *edges*.

Pada penelitian ini, struktur data yang digunakan terdiri dari variabel independen yaitu kata dasar *tweet* yang telah dilakukan praproses teks dan variabel dependen yaitu klasifikasi sentimen *tweet* (positif atau negatif). Penelitian ini bertujuan melakukan analisis kompetitif antar sosial media dengan melihat sentimen mengenai tanggapan masyarakat terhadap taksi konvensional dan taksi berbasis *online* berbasis *text mining* data *tweet*. Melalui penelitian ini diharapkan dapat memberikan saran kepada penyedia jasa taksi konvensional dan taksi berbasis *online* serta kepada masyarakat sebagai pertimbangan dalam menentukan penyedia jasa taksi dengan pelayanan terbaik.

1.2 Perumusan Masalah

Mengetahui tanggapan dari masyarakat terhadap layanan jasa taksi konvensional dan taksi berbasis *online* merupakan hal yang tidak dapat dikesampingkan. Sehingga akurasi klasifikasi yang tinggi pada analisis sentimen dengan metode klasifikasi yang tepat digunakan menjadi suatu hal yang penting. *Naïve Bayes Classifier* (NBC) merupakan metode klasifikasi yang mengacu pada teorema probabilitas bersyarat serta memiliki algoritma yang sederhana namun memiliki akurasi yang tinggi. Sedangkan, *Support Vector Machine* (SVM) merupakan metode berbasis *machine learning* yang sangat menjanjikan untuk dikembangkan karena memiliki performansi tinggi dan dapat diaplikasikan secara luas untuk klasifikasi dan estimasi. Kedua metode tersebut akan dibandingkan untuk mengetahui metode manakah yang menghasilkan *error* terkecil. Berdasarkan penjelasan tersebut, maka permasalahan utama yang akan dibahas dalam penelitian ini diantaranya adalah berapa tingkat ketepatan klasifikasi sentimen pengguna Twitter terhadap taksi konvensional dan taksi berbasis *online* dengan menggunakan metode *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM), serta apa kata yang sering muncul berdasarkan masing-masing sentimen. Selain itu, dilakukan identifikasi pengguna Twitter yang berpengaruh berdasarkan topik tentang taksi konvensional dan taksi berbasis *online* dengan menggunakan *Social Network Analysis* (SNA).

1.3 Tujuan Penelitian

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

1. Mengetahui karakteristik data yang digunakan berdasarkan data *tweet* tentang taksi konvensional dan taksi berbasis *online*.
2. Mendapatkan hasil ketepatan klasifikasi sentimen pengguna Twitter terhadap taksi konvensional dan taksi berbasis *online* dengan menggunakan metode *Naïve Bayes Classifier*

dan *Support Vector Machine*, serta membandingkan ketepatan klasifikasi antar kedua metode.

3. Mendapatkan kata-kata yang sering muncul berdasarkan masing-masing sentimen menggunakan *word cloud*.
4. Dapat menganalisa hasil representasi *graph* yang terbentuk menggunakan *Social Network Analysis* untuk mengidentifikasi pengguna Twitter yang berpengaruh berdasarkan topik tentang taksi konvensional dan taksi berbasis *online*.

1.4 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat bermanfaat dalam membantu pihak penyedia taksi konvensional dan taksi berbasis *online* yang bersangkutan dalam memahami tanggapan masyarakat mengenai pelayanan yang diberikan serta mempercepat proses klasifikasi tanggapan masyarakat (sentimen) karena telah mendapatkan model dari data *training* sehingga pihak penyedia pelayanan taksi tidak perlu mengklasifikasikan ulang data secara manual. Selain itu, penyedia jasa taksi konvensional dan taksi berbasis *online* dapat melihat perbandingan secara kompetitif sosial media (khususnya Twitter) antar penyedia jasa. Penelitian ini diharapkan dapat membantu pihak-pihak yang ingin mengetahui kata kunci yang identik terhadap taksi konvensional dan taksi berbasis *online* melalui *posting tweet* masyarakat pengguna Twitter. Sebagai tambahan, penelitian ini dapat pula memberikan tambahan informasi kepada publik terhadap taksi konvensional dan taksi berbasis *online* pilihan melalui hasil klasifikasi sentimen.

1.5 Batasan Penelitian

Batasan penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Penelitian menggunakan studi kasus taksi konvensional dan taksi berbasis *online* yaitu BlueBird Taxi sebagai taksi

konvensional, serta GoCar, dan GrabCar sebagai taksi berbasis *online*.

2. Penelitian hanya melakukan analisis sentimen terhadap *tweet* berbahasa Indonesia.
3. Penelitian tidak mengatasi *tweet* yang mempunyai kata atau frasa dengan arti ganda dan berbeda pada sebuah kalimat.
4. Penelitian tidak memperhatikan latar belakang atau demografi dari pemilik *account* Twitter.
5. Klasifikasi sentimen data awal ditentukan secara subyektif oleh peneliti.
6. Data *tweet* yang digunakan merupakan *tweet* yang diunggah pada tanggal 5 Februari 2018 hingga 5 Mei 2018, sehingga substansi *tweet* berfokus pada pelayanan yang diberikan oleh penyedia jasa taksi konvensional dan taksi berbasis *online* setelah adanya penetapan Peraturan Menteri Perhubungan Nomor 108 Tahun 2017, serta isu-isu yang sedang beredar seperti berbagai tindak kriminalitas yang kerap terjadi.

BAB II
TINJAUAN PUSTAKA

2.1 *Text Mining*

Text mining merupakan salah satu cabang ilmu *data mining* yang menganalisis suatu data berupa *text*. Menurut Han, Kamber, dan Pei (2012), *text mining* adalah suatu langkah analisis teks yang dilakukan otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. Ide awal pembuatan *text mining* adalah untuk menemukan pola-pola informasi yang dapat digali dari teks yang tidak terstruktur (Hamzah, 2012). Dengan demikian, *text mining* mengacu juga kepada istilah *text data mining* (Hearst dalam Hamzah, 2012) atau penemuan pengetahuan dari basis data teks (Feldman & Dagan dalam Hamzah, 2012). Saat ini, *text mining* telah mendapatkan perhatian dalam berbagai bidang, antara lain dibidang keamanan, biomedis, pengembangan perangkat lunak dan aplikasi, media *online*, pemasaran, dan akademik. Seperti halnya dalam *data mining*, aplikasi *text mining* pada suatu studi kasus, harus dilakukan sesuai prosedur analisis. Langkah awal sebelum suatu data teks dianalisis menggunakan metode-metode dalam *text mining* adalah melakukan *pre-processing* teks. Sehingga setelah didapatkan data yang siap diolah, analisis *text mining* dapat dilakukan.

Text mining dapat digunakan untuk proses penemuan *rule* baru dengan algoritma pengelompokan, asosiasi, dan *ranking*. Beberapa fungsi tersebut, yang paling banyak dilakukan adalah proses pengelompokan. Terdapat dua jenis metode pengelompokan teks, yaitu *text clustering* dan *text classification*. Menurut Darujati dan Gumelar (2012), *text clustering* berhubungan dengan proses menemukan sebuah struktur kelompok yang belum terlihat (tak terpandu atau *unsupervised*) dari sekumpulan dokumen. Sedangkan, *text classification* dapat dianggap proses untuk membentuk golongan (kelas-kelas) dari dokumen berdasarkan pada kelas kelompok yang sudah diketahui sebelumnya (terpandu

atau *supervised*). Berdasarkan pengertian ini, dapat dinyatakan bahwa proses klasifikasi (*supervised*) merupakan proses yang lebih mudah dilakukan *monitoring*, karena terdapat target kelas yang akan dituju dalam analisisnya. Beberapa contoh metode yang dapat digunakan untuk klasifikasi suatu data teks adalah dengan *Naïve Bayes Classifier* (NBC) dan *Support VectorMachine* (SVM).

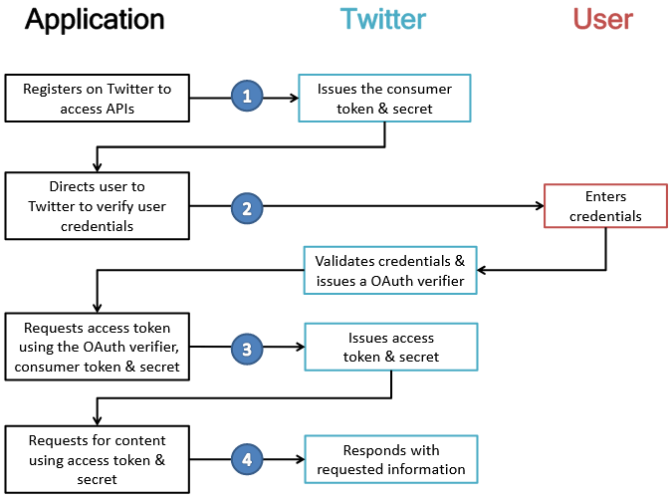
2.2 *Crawling* Data Twitter dengan Twitter API

Berdasarkan Omnicare Group (*Healthcare Digital Marketing Agency*) menunjukkan bahwa terdapat 330 juta pengguna aktif Twitter dengan jumlah *tweet* yang dikirim setiap hari mencapai 500 juta *tweet*. Sebelum melakukan pengolahan data, langkah pertama yang harus dilakukan adalah *crawling* Twitter data dengan menggunakan API (*Application Programming Interface*). Kegunaan dari API adalah untuk mengetahui informasi tentang pengguna, jaringan pengguna yang terdiri dari koneksi, dan tweet yang di bicarakan (Kumar, Morstatter, & Liu, 2013).

API adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan sistem informasi. API juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak sehingga dapat memudahkan seorang programmer untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan developer menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi.

Twitter API (*Application Programming Interface*) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh Twitter untuk mengambil atau memodifikasi data

dari Twitter. Twitter API dapat digunakan untuk membangun aplikasi perangkat lunak, laman *web*, *widget* dan beberapa proyek lain yang berinteraksi dengan Twitter. Fitur yang ditawarkan adalah modifikasi (*get* dan *post*) berbagai data dari pengguna Twitter seperti status, jaringan, media, dan pengaturan (Pratama, 2015). Pengambilan data dari Twitter API memerlukan kode token *Application Key* dan *Application Secret* yang bisa didapatkan dengan mendaftarkan aplikasi yang dibuat di situs Twitter Developers. Terdapat batasan penggunaan pada API ini dalam suatu satuan waktu. Selain itu, data yang diperbolehkan untuk diambil adalah data waktu nyata (*real time*), sehingga data historis atau data yang sudah lampau tidak dapat diambil kembali (Ma'ady, 2014). Langkah-langkah *crawling* data Twitter dengan menggunakan Twitter API ditunjukkan pada Gambar 2.1.



(Sumber: Kumar, Morstatter, & Liu, 2013)

Gambar 2.1 *OAuth Workflow*

2.3 *Sentiment Analysis*

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi

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

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

2.4 Klasifikasi Teks

Klasifikasi teks (*text classification*) atau juga dikenal dengan ketegorisasi teks adalah bagian dari pengelompokkan dokumen atau teks dengan memperhatikan kategori. Secara umum, klasifikasi teks termasuk penggolongan teks berbasis topik dan penggolongan teks berbasis kategori (*genre*) (Ikonomakis, Kotsiantis, & Tampakas, 2005). Teks dapat dikategorikan menjadi artikel ilmiah, laporan surat kabar, tinjauan film, iklan, dan lain-lain. Klasifikasi teks dapat menggunakan banyak teknik pembelajaran mesin (*machine learning*). Pendekatan yang sering diadopsi adalah pohon keputusan (*decision tree*), induksi aturan (*rule induction*), jaringan saraf (*neural network*), *Naïve Bayes Classifier* (NBC), dan *Support Vector Machines* (SVM). Tantangan dari klasifikasi teks adalah sifat data yang tidak terstruktur dan sulit untuk menangani, sehingga diperlukan proses *text mining*. Diharapkan melalui proses *text mining*, informasi yang

ada dapat dikeluarkan secara jelas di dalam teks tersebut dan dapat dipergunakan dalam proses analisis menggunakan alat bantu komputer (Witten, Frank, & Hall, 2011).

2.5 Praproses Teks

Praproses teks merupakan tahapan awal dalam pengolahan teks yang digunakan untuk pengubah bentuk dokumen menjadi data yang terstruktur sesuai kebutuhannya agar dapat diolah lebih lanjut dalam proses *text mining*. Tahapan praproses teks dalam klasifikasi bertujuan untuk meningkatkan akurasi klasifikasi data. Praproses dalam *text mining* cukup rumit karena dalam Bahasa Indonesia terdapat berbagai aturan penulisan kalimat maupun pembentukan kata berimbuhan. Terdapat empat aturan aturan pembentukan kata berimbuhan (afiks) untuk merubah makna kata dasar yaitu sebagai berikut.

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

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

akhiran dalam Bahasa Indonesia dapat dilakukan dengan dua acara.

- Penggabungan atau pengimbuhan yang dilakukan dengan bersamaan pada bentuk dasar, gabungan awal itu dinamakan konfiks. Artinya bentuk dasar yang diimbuhkan awalan-akhiran secara bersamaan itu tidak mempunyai tatanan kata sebelumnya.
- Pengimbuhan awalan-akhiran dalam Bahasa Indonesia yang mempunyai tatanan kata sebelumnya, pengimbuhan ini dinamakan simulfiks. Artinya pengimbuhan awalan-akhiran itu dilakukan secara bertahap, sehingga mempunyai tataran sebelum bentuk kompleks itu terwujud.

d. Sisipan (infiks), merupakan bentuk terikat yang diimbuhkan pada bentuk dasar. Pengimbuhannya ditempatkan ditengah atau diantara bentuk dasar. Infiks dalam Bahasa Indonesia antara lain: -el-, -em-, -er-, -in-.

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

a. *Cleansing*, merupakan proses membersihkan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam *Twitter* adalah karakter HTML, *emoticons*, *hastag (#)*, *username (@username)*, dan *URL* (Buntoro, Adji, & Purnamasari, 2014).
b. *Case Folding*, merupakan proses untuk mengubah semua karakter teks menjadi huruf kecil serta menghilangkan tanda baca dan angka. Cara kerja *case folding* adalah memproses huruf alphabet dari “a” hingga “z” saja sehingga karakter selain huruf tersebut akan dihilangkan seperti tanda baca titik (.), koma (,), dan angka (Weiss, 2010).
c. *Stemming*, merupakan proses untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran) (Ariadi &

Fithriasari, 2015). Pada penelitian ini algoritma *stemming* yang digunakan adalah *Confix Striping Stemmer* yang merupakan pengembangan dari algoritma *Nazief and Adriani's Stemmer*.

d. *Stopwords*, merupakan kosakata yang bukan termasuk kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat (Dragut, Fang, & Sistla, 2009). Kosakata yang dimaksud yaitu seperti kata penghubung dan kata keterangan yang bukan merupakan kata unik, misalnya "dari", "akan", "seorang", dan sebagainya.

e. *Tokenizing*, merupakan proses memecah yang semula berupa kalimat menjadi kata-kata atau memutus urutan string menjadi potongan-potongan seperti kata-kata berdasarkan tiap kata yang menyusunnya. Sehingga dapat dikatakan mengembalikan kata pengubung (Liu, 2010).

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

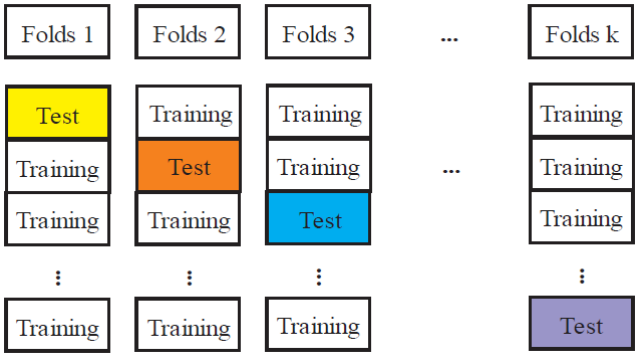
Term Frequency Inverse Document Frequency (TF-IDF) merupakan pembobot yang dilakukan setelah proses *tokenizing* atau ekstraksi artikel (Ariadi & Fithriasari, 2015). Proses metode TF-IDF adalah menghitung bobot dengan cara integrasi antara *term frequency* (*tf*) dan *inverse document frequency* (*idf*). Langkah dalam TF-IDF adalah untuk menemukan jumlah kata yang kita ketahui (*tf*) setelah dikalikan dengan berapa banyak *tweet* dimana suatu kata itu muncul (*idf*). Persamaan dalam menemukan pembobot dengan TF-IDF terdapat pada persamaan (2.1).

$$w_{ij} = tf_{ij} \times idf \text{ dengan } idf = \log\left(\frac{N}{df_j}\right), \qquad (2.1)$$

dimana, w_{ij} adalah bobot dari kata *i* pada artikel ke-*j*, *N* merupakan jumlah seluruh dokumen tf_{ij} adalah jumlah kemunculan kata *i* pada dokumen *j*, df_{ij} adalah jumlah artikel *j* yang mengandung kata *i*. TF-IDF dilakukan agar data dapat dianalisis menggunakan*Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM).

2.7 *K-fold Cross Validation*

K-fold cross validation adalah salah satu metode yang digunakan untuk mempartisi data menjadi data *training* dan data *testing*. Metode ini banyak digunakan peneliti karena dapat mengurangi bias yang terjadi dalam pengambilan sampel. *K-fold cross validation* secara berulang-ulang membagi data menjadi data *training* dan data *testing*, dimana setiap data mendapat kesempatan menjadi data *testing* (Gokgoz & Subasi, 2015). *K* merupakan besar angka partisi data yang digunakan untuk pembagian *training-testing*. Ilustrasi pembagian data menggunakan *K-fold cross validation* terdapat pada Gambar 2.2.



(Sumber: Kurniawan, 2017)

Gambar 2.2 Ilustrasi Pembagian Data

2.8 *Naïve Bayes Classifier* (NBC)

Klasifikasi Bayes adalah klasifikasi statistik yang dapat memprediksi kelas suatu anggota probabilitas. Untuk klasifikasi Bayes sederhana yang lebih dikenal sebagai *Naïve Bayes Classifier* (NBC) dapat diasumsikan bahwa efek dari suatu nilai atribut sebuah kelas yang diberikan adalah bebas dari atribut-atribut lain. Asumsi ini disebut *class conditional independence* yang dibuat untuk memudahkan perhitungan-perhitungan, pengertian ini dianggap '*Naïve'*. Dalam bahasa lebih sederhana, '*Naïve'* itu mengasumsikan bahwa kemunculan suatu term kata dalam suatu

kalimat tidak dipengaruhi kemungkinan kata-kata yang lain dalam kalimat, padahal dalam kenyataannya bahwa kemungkinan kata dalam kalimat sangat dipengaruhi kemungkinan keberadaan kata-kata yang ada dalam kalimat (Durajati & Gumelar, 2012).

Pembelajaran *Bayes* merupakan jenis pembelajaran yang paling praktis untuk banyak permasalahan, dengan cara menghitung nilai probabilitas yang terlihat. Teknik pembelajaran *Bayes* sangat kompetitif jika dibandingkan dengan algoritma pembelajaran lainnya dan dalam banyak kasus nilainya cenderung melebihi yang lain. Algoritma pembelajaran *Bayes* sangat penting, dikarenakan Bayes mampu memberikan perspektif yang unik dalam memahami banyak algoritma yang tidak secara langsung menggunakan probabilitas. Pendekatan ini didasarkan pada teori *Bayes* yang dapat dinyatakan pada persamaan (2.2).

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}, \tag{2.2}$$

dimana,

$P(Y|X)$ adalah probabilitas kategori *Y* berdasarkan kondisi *X* (*posterior probability*).

$P(X|Y)$ adalah probabilitas *X* dengan kemungkinan *Y* (*likelihood*).

$P(Y)$ adalah nilai probabilitas pada data *training* melalui perhitungan kategori *Y* (*prior probability*).

$P(X)$ adalah total probabilitas untuk semua data *training X*.

$P(X|Y)$ merupakan probabilitas mendapatkan *X* sebagai *input* ketika sudah diketahui masuk ke dalam kelas Y_i. Nilai $P(X)$ merupakan konstanta yang bernilai konstant sehingga dapat dihilangkan, karena jika nilai probabilitas diurutkan berdasarkan kelas, maka nilai $P(X)$ akan selalu sama. Maka dapat disederhanakan menjadi persamaan (2.3).

$$P(Y|X) \propto P(X|Y)P(Y) \tag{2.3}$$

Naïve Bayes Classifier (NBC) atau multinomial *Naïve Bayes* merupakan model penyederhanaan dari algoritma Bayes yang cocok dalam pengklasifikasian teks atau dokumen. Algoritma *Naïve Bayes Classifier* merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Feldman & Sanger, 2007). Kelebihan NBC adalah algoritmanya sederhana tetapi memiliki akurasi yang tinggi. Terdapat dua tahap dalam klasifikasi *tweet*. Tahap pertama adalah pelatihan (*training*) terhadap *tweet* yang telah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi (*testing*) *tweet* yang belum diketahui kategorinya (Falahah & Nur, 2015).

Klasifikasi adalah proses pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data dengan tujuan agar model tersebut dapat digunakan untuk memprediksi kelas dari suatu objek yang belum diketahui kelasnya. Selain itu, klasifikasi *Naïve Bayes* terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam basis data dengan jumlah yang besar (Han, Kamber, & Pei, 2012). Salah satu penerapan teori *Bayes* dalam klasifikasi adalah *Naïve Bayes*. Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori yang diujikan (V_{MAP}). Adapun persamaan V_{MAP} terdapat pada persamaan (2.5).

$$V_{MAP} = \arg\max_{V_j \in V} P\left(V_j | w_1, w_2, ..., w_k\right), \tag{2.5}$$

dengan menggunakan pendekatan teori *Naïve Bayes* pada persamaan (2.2), maka persamaan (2.5) dapat ditulis menjadi persamaan (2.6).

$$V_{MAP} = \arg\max_{V_j \in V} \frac{P\left(V_j\right) P\left(w_1, w_2, ..., w_k | V_j\right)}{P\left(w_1, w_2, ..., w_k\right)}, \tag{2.6}$$

karena nilai $P\left(w_1, w_2, ..., w_k\right)$ konstan, maka dapat disederhanakan seperti persamaan (2.3) menjadi persamaan (2.7).

$$V_{MAP} \propto \arg\max_{V_j \in V} P\left(w_1, w_2, ..., w_k | V_j\right) P\left(V_j\right) \tag{2.7}$$

Naïve Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara *conditional* saling bebas jika diberikan nilai *output* atau dapat diberikan persamaan (2.8).

$$P\left(w_1, w_2, \ldots, w_k|V_j\right) = \prod_{k=1}^{n} P\left(w_k|V_j\right) \tag{2.8}$$

Sehingga jika persamaan (2.8) disubstitusikan pada persamaan (2.5), maka akan didapatkan pendekatan yang dipakai dalam *Naïve Bayes Classifier* pada persamaan (2.9).

$$V_{NB} = \arg\max_{V_j \in V} P\left(V_j\right) \prod_{k=1}^{n} P\left(w_k|V_j\right), \tag{2.9}$$

dimana, V_{NB} merupakan nilai *output* hasil klasifikasi *Naïve Bayes*. Nilai $P\left(V_j\right)$ pada persamaan (2.9) dihitung pada saat *training*, didapat dengan persamaan (2.10).

$$P\left(V_j\right) = \frac{\left|docs_j\right|}{\left|training\right|}, \tag{2.10}$$

dengan, $|docs_j|$ merupakan frekuensi *tweet* pada setiap kategori *j* dalam *training*. Sedangkan, |*training*| merupakan banyaknya *tweet* pada data *training*. Untuk setiap probabilitas kemunculan kata w_k pada setiap kategori kelas V_j yaitu $P\left(w_k|V_j\right)$, dihitung pada saat training dengan persamaan (2.11).

$$P\left(w_k \mid V_j\right) = \frac{n_k + 1}{\left|n + \text{kosa kata}\right|}, \tag{2.11}$$

dimana, n_k adalah frekuensi kata w_k dalam *tweet* yang berkategori V_j, sedangkan *n* adalah seluruh kata dalam *tweet* dengan kategori V_j dan |*kosa kata*| adalah banyak kata dalam data *training*.

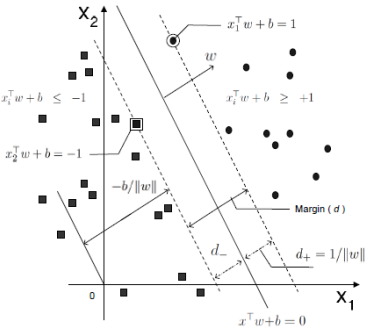
2.9 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah sistem pembelajaran yang menggunakan hipotesis fungsi linear dalam ruang berdimensi tinggi dan dilatih dengan algoritma berdasarkan teori optimasi dengan menerapkan *learning bias* yang berasal dari teori statistik

(Cristianini & Shawe-Taylor, 2000). Tujuan utama dari metode ini adalah membangun pemisah optimum yang disebut OSH (*Optimal Separating Hyperplane*) sehingga dapat digunakan untuk klasifikasi.

2.9.1 SVM pada *Linearly Separable Data*

Linearly separable data merupakan data yang dapat dipisahkan secara linier. Misalkan diberikan data *set* $\mathbf{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ setiap observasi terdiri dari sepasang *p* prediktor $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}\}$, $\mathbf{x}_i \in \Re^p$ untuk *i* = 1, 2, …, *n*, dimana *n* merupakan banyak data dan label kelas dari data $\mathbf{x}_i$ dinotasikan $y_i \in y = \{-1, +1\}$. Jika $\mathbf{x}_i$ adalah anggota kelas (+1) maka $\mathbf{x}_i$ diberi label (target) $y_i = +1$ dan jika tidak maka diberi label (target) $y_i = -1$, sehingga data yang diberikan berupa pasangan $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ merupakan himpunan data *training* dari dua kelas yang akan diklasifikasikan dengan SVM. Pada Gambar 2.3 dapat dilihat berbagai alternatif bidang pemisah yang dapat memisahkan semua data set sesuai dengan kelasnya. Bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin yang paling besar.



(Sumber: Härdle, Prastyo, & Hafner, 2012)

Gambar 2.3 *Linearly Separable Data*

Data yang berada pada bidang pembatas disebut dengan *support vector*. Dalam Gambar 2.3, dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar. Bidang pembatas pertama membatasi kelas pertama, sedangkan bidang pembatas kedua membatasi kelas kedua. Sehingga diperoleh persamaan (2.12).

$$\begin{aligned} \mathbf{x}_i'\mathbf{w} + b \geq +1, y_i = +1 \\ \mathbf{x}_i'\mathbf{w} + b \leq -1, y_i = -1 \end{aligned} \tag{2.12}$$

w adalah normal bidang dan *b* adalah posisi bidang alternatif terhadap pusat koordinat. Nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) adalah $\frac{1-b-(-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$. Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan (2.12). Dengan mengalikan *b* dan **w** dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstanta yang sama (Gunn, 1998). Oleh karena itu, konstrain pada persamaan (2.12) merupakan *scaling constraint* yang dapat dipenuhi dengan *rescaling b* dan **w**. Selain itu karena memaksimalkan $\frac{1}{\|\mathbf{w}\|}$ sama dengan meminimumkan $\|\mathbf{w}\|$ dan jika kedua bidang pembatas pada persamaan (2.12) direpresentasikan dalam pertidaksamaan (2.13).

$$y_i(\mathbf{x}_i'\mathbf{w} + b) - 1 \geq 0 \text{ dengan } i = 1, 2, \ldots, n \tag{2.13}$$

Maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu pada persamaan (2.14).

$$\min_{\mathbf{w},b} \left\{ \frac{\|\mathbf{w}\|^2}{2} \right\} \text{dengan } y_i(\mathbf{x}_i'\mathbf{w} + b) \geq 1 \tag{2.14}$$

Persamaan (2.14) ini akan lebih mudah diselesaikan jika diubah ke dalam formula *Lagrangian* yang menggunakan *Lagrange Multiplier*. Dengan demikian permasalahan optimasi konstrain dapat diubah menjadi persamaan (2.15).

$$\min L_P(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\left(\mathbf{x}_i'\mathbf{w}+b\right)-1\right)\right), \quad (2.15)$$

dengan tambahan konstrain, $\alpha_i \geq 0$ (nilai koefisien *Lagrange*). Dengan meminimumkan L_p terhadap **w** dan *b*, maka dari $\frac{\partial L_P(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \mathbf{w}} = 0$ dan dari $\frac{\partial L_P(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} = 0$, diperoleh persamaan (2.16).

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \text{ dan } \sum_{i=1}^{n}\alpha_i y_i = 0 \quad (2.16)$$

Vektor **w** sering kali bernilai besar (tak terhingga), tetapi nilai α_i terhingga. Untuk itu formula *Lagrangian* L_P (*primal problem*) diubah ke dalam L_D (*dual program*). Dengan mensubstitusikan persamaan (2.16) ke (2.15), diperoleh L_D pada persamaan (2.17).

$$L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i'\mathbf{x}_j \quad (2.17)$$

Sehingga persoalan pencarian bidang pemisah terbaik dapat dirumuskan pada persamaan (2.18).

$$\max_{\alpha} L_D = \max_{\alpha}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i'\mathbf{x}_j\right), \quad (2.18)$$

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

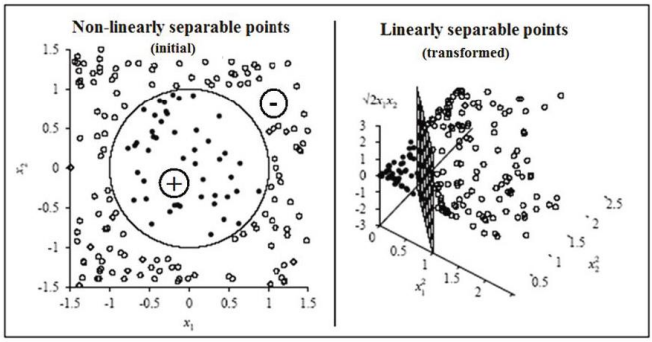
Dengan demikian, dapat diperoleh dari nilai α_i yang nantinya digunakan untuk menemukan **w**. Terdapat nilai α_i untuk setiap data *training* yang memiliki nilai $\alpha_i > 0$ adalah *support vector*, sedangkan sisanya memiliki nilai $\alpha_i = 0$. Dengan demikian, fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*. Formula pencarian bidang pemisah terbaik ini adalah permasalahan *quadratic programming*, sehingga nilai maksimum global dari α selalu dapat ditemukan. Setelah solusi permasalahan *quadratic programming* ditemukan (nilai α_i), maka kelas dari suatu data *x testing* ($\mathbf{x}_{new}$) dapat ditentukan dengan persamaan (2.19).

$$f\left(\mathbf{x}_{new}\right) = sign\left(\mathbf{x}'_{new}\hat{\mathbf{w}} + \hat{b}\right), \tag{2.19}$$

dimana $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$ dan $b = \frac{1}{n_{sv}}\left(\sum_{i=1}^{n_{sv}} \frac{1}{y_i} - \left(\mathbf{x}'_{new}\hat{\mathbf{w}}\right)\right)$ dengan $\mathbf{x}_i$ adalah *support vector*, $\mathbf{x}_{new}$ adalah data yang diklasifikasikan, dan n_{sv} adalah jumlah *support vector*.

2.9.2 SVM pada *Non Linearly Separable Data* dengan Metode Kernel

Klasifikasi data yang tidak dapat dipisahkan secara linier diperlukan formula SVM yang dimodifikasi agar dapat menemukan solusinya. Penggambaran *non linearly separable* data dapat dilihat pada Gambar 2.4.



(Sumber: Gorunescu, 2011)

Gambar 2.4 *Non Linearly Separable Data*

Untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier, formula SVM harus dimodifikasi karena tidak akan ada solusi yang ditemukan. Pencarian *hyperplane* yang optimal akan memperhatikan data-data yang tidak berada pada kelasnya (*misclassification error*) yang dilambangkan dengan ξ. Oleh karena itu, kedua bidang pembatas (persamaan (2.12)) harus diubah sehingga lebih fleksibel dengan penambahan variabel ξ_i (ξ_i

$\geq$ 0, $\forall_i : \xi_i = 0$ jika x_i diklasifikasikan dengan benar) menjadi $y_i\left(\mathbf{x}_i'\mathbf{w}+b\right)\geq 1-\xi_i$ untuk $y_i = 1$ dan $y_i\left(\mathbf{x}_i'\mathbf{w}+b\right)\geq -\left(1-\xi_i\right)$ untuk $y_i = -1$. Pencarian bidang pemisah terbaik dengan penambahan variabel ξ_i sering disebut dengan *soft margin hyperplane*. Dengan demikian, formula pencarian bidang pemisah terbaik berubah menjadi persamaan (2.20).

$$\min_{\mathbf{w},\xi_i} = \left\{ \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n}\xi_i \right\} \qquad (2.20)$$

$$\text{dengan } y_i\left(\mathbf{x}_i'\mathbf{w}+b\right)\geq 1-\xi_i \text{ , } \xi_i \geq 0,$$

dimana, *C* adalah parameter yang menentukan besar biaya akibat kesalahan klasifikasi (*misclassification*) dari data *training* selama proses pembelajaran dan nilainya ditentukan oleh pengguna. Ketika nilai *C* besar, maka *margin* aka menjadi lebih kecil, yang mengindikasikan bahwa tingkat toleransi kesalahan akan menjadi lebih kecil ketika suatu kesalahan tejadi. Sebaliknya, jika nilai *C* kecil, tingkat toleransi kesalahan akan menjadi lebih besar (Huang, Hung, Lee, Li, & Jiang, 2014). Sehingga didapatkan fungsi *Lagrange Multiplier* untuk kasus *linearly nonseparable* pada persamaan (2.21).

$$\begin{aligned}\min L_P\left(\mathbf{w},b,\boldsymbol{\alpha},\mu,\xi\right) = \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right) - \sum_{i=1}^{n}\alpha_i \\ \left(y_i\left(\mathbf{x}_i'\mathbf{w}+b\right)-1+\xi_i\right) + \sum_{i=1}^{n}\mu_i\xi_i \text{ ,}\end{aligned} \qquad (2.21)$$

dimana, $\alpha_i \geq 0$ dan $\mu_i \geq 0$ adalah *Lagrange Multiplier*, $\alpha_i\left(y_i\left(\mathbf{x}_i'\mathbf{w}+b\right)-1+\xi_i\right)=0$ dan $\mu_i\xi_i = 0$. Nilai optimal dari persamaan (2.21) dapat dihitung dengan meminimalkan L_P terhadap **w**, *b*, dan ξ maka dari $\frac{\partial L_P\left(\mathbf{w},b,\boldsymbol{\alpha},\mu,\xi\right)}{\partial \mathbf{w}} = 0$ dan dari

$\frac{\partial L_P(\mathbf{w}, b, \boldsymbol{\alpha}, \mu, \xi)}{\partial b} = 0$, serta $\frac{\partial L_P(\mathbf{w}, b, \boldsymbol{\alpha}, \mu, \xi)}{\partial \xi_i} = 0$ diperoleh persamaan (2.22).

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \ , \ \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ , dan } \alpha_i = C - \mu_i \qquad (2.22)$$

Sehingga persoalan pencarian bidang pemisah terbaik dapat dirumuskan pada persamaan (2.23).

$$\max_{\alpha} L_D = \max_{\alpha} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i' \mathbf{x}_j \right) \qquad (2.23)$$

Constraint yang digunakan untuk memaksimalkan α_i pada persamaan (2.23) adalah $\sum_{i=1}^{n} \alpha_i y_i = 0, \ 0 \le \alpha_i \le C$.

Metode lain untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier adalah dengan metode Kernel. Metode Kernel mentransformasikan data ke dalam dimensi ruang fitur (*feature space*) sehingga dapat dipisahkan secara linier pada *feature space*. Suatu data *x* di *input space* ke *feature space* dengan menggunakan fungsi transformasi $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$. Sehingga nilai $\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \phi(\mathbf{x}_i)$ dan fungsi klasifikasi (*score*) yang dihasilkan ditunjukkan pada persamaan (2.24).

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j) + b \qquad (2.24)$$

Feature space dalam praktiknya biasanya memiliki dimensi yang tinggi dari vektor *input space* (Gunn, 1998). Hal ini mengakibatkan komputasi pada *feature space* mungkin sangat besar, karena ada kemungkinan *feature space* dapat memiliki jumlah *feature* yang tidak terhingga.

Selain itu, sulit mengetahui fungsi transformasi yang tepat. "*Kernel Trick*" digunakan untuk mengatasi masalah ini pada SVM.

Pada persamaan (2.24) dapat dilihat *dot product* $\phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$. Jika terdapat sebuah fungsi Kernel *K* sehingga $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$ maka fungsi transformasi $\phi(\mathbf{x}_i)$ tidak perlu diketahui secara persis. Dengan demikian, fungsi yang dihasilkan dari *training* ditunjukkan pada persamaan (2.25).

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b \tag{2.25}$$

Syarat sebuah fungsi menjadi fungsi Kernel adalah memenuhi teorema Mercer yang menyatakan bahwa matriks Kernel yang dihasilkan harus bersifat *positive semi-definite*. Fungsi Kernel yang umum digunakan pada metode SVM ditunjukkan pada Tabel 2.1.

Tabel 2.1 Fungsi Kernel pada SVM

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

Dari keempat metode Kernel yang ada, fungsi Kernel RBF direkomendasikan untuk diuji. Fungsi Kernel RBF memiliki performansi yang sama dengan Kernel Linier pada parameter tertentu, memiliki perilaku seperti fungsi Kernel Sigmoid dengan parameter tertentu dan rentang nilai kecil [0,1] (Hsu, Chang, & Lin, 2010). Sedangkan dalam kasus data teks mayoritas peneliti menggunakan ketiga jenis Kernel pertama. Diantaranya adalah membandingkan ketepatan klasifikasi dengan Kernel Linier dan Kernel RBF dimana hasil yang didapatkan yaitu Kernel Linier lebih baik (Guduru, 2006) dan membandingkan antara Kernel

Polynomial dan Kernel RBF hasilnya adalah RBF yang lebih baik (Joachims, 1998).

Pembahasan pada SVM ini akan sama dengan pembahasan pada NBC, perbedaannya adalah ditambahkan parameter untuk SVM dengan Kernel RBF yaitu *C* dan γ, sedangkan untuk linier digunakan parameter *C*. Saat menentukan kombinasi parameter *C* dan γ, terlebih dahulu harus dideskripsikan *range* dari kedua parameter untuk dikombinasikan. Parameter *C* yang paling tepat untuk SVM adalah antara 10^{-2} hingga 10^{-4} (Huang, Lee, Lin, & Huang, 2007).

2.10 Ketepatan Klasifikasi

Pengukuran ketepatan klasifikasi dilakukan untuk melihat performa klasifikasi yang telah dilakukan. Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah pada setiap kelas prediksi dan kelas aktual yang terdiri dari *TP* (*True Positive*) yaitu jumlah *tweet* bersentimen positif yang tepat diprediksi dalam kelas positif, *TN* (*True Negative*) yaitu *tweet* yang tepat terprediksi dalam kelas negatif, *FP* (*False Positive*) yaitu *tweet* bersentimen negatif yang terprediksi dalam kelas positif, dan *FN* (*False Negative*) yaitu *tweet* bersentimen positif yang terprediksi dalam kelas negatif. *Confusion matrix* yang memuat keempat nilai tersebut terdapat pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix*

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

Pengukuran yang sering digunakan untuk menghitung ketepatan klasifikasi adalah *accuracy*, *specificity*, dan *sensitivity* (Hotho, Numberger, & Paas, 2005). *Accuracy* merupakan persentase dokumen yang teridentifikasi secara tepat dari total dokumen dalam proses klasifikasi. Akurasi digunakan untuk menghitung ketepatan klasifikasi sebuah dokumen yang

mempunyai data yang *balanced* pada tiap kategorinya. Persamaan untuk menghitung *accuracy*, *sensitivity*, dan *specificity* terdapat pada persamaan (2.26), (2.27), dan (2.28).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.26}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.27}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.28}$$

F-measure merupakan kompromi dari *recall* dan *precision* untuk mengukur kinerja keseluruhan pengklasifikasi. Cara perhitungan *F-measure* ditunjukkan pada persamaan (2.29) (Hotho, Numberger, & Paas, 2005). *Recall* mengindikasikan sebagian kecil dari dokumen yang relevan diambil. Sedangkan *precision* mengkuantitatifkan fraksi dokumen diambil yang sebenarnya relevan, dalam contoh milik kelas sasaran (Bekkar, Djemaa, & Alitouch, 2013). Persamaan untuk *recall* dan *precision* ditunjukkan pada persamaan (2.30) dan (2.31).

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{2.29}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.30}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.31}$$

Sedangkan untuk data *imbalanced*, pengukuran ketepatan klasifikasi dapat menggunakan *G-Mean*. *G-Mean* atau *Geometric Mean* merupakan rata-rata geometrik nilai *recall* dari data yang memiliki dua kategori (Sun, Kamel, & Wang, 2006). *G-Mean* cukup penting digunakan untuk pengukuran menghindari *overfitting* ke kelas negatif dan sejauh mana kelas positif dapat terpisahkan. Persamaan untuk mendapatkan nilai *G-Mean* terdapat pada persamaan (2.32).

$$G - Mean = \sqrt{Sensitivity \times Specificity} \tag{2.32}$$

Selain *G-Mean*, digunakan juga nilai *Area Under Curve* (AUC). AUC merupakan indikator performansi kurva ROC (*Receiver Operating Characteristic*) yang dapat meringkas kinerja sebuah *classifier* menjadi satu nilai (Bekkar, Djemaa, & Alitouch, 2013). Kurva ROC merupakan metode yang popular digunakan untuk mengukur kinerja metode klasifikasi yang digunakan jika terdapat dua kelas kategori. Area di bawah kurva ROC (AUC) dapat digunakan sebagai ukuran kinerja metode klasifikasi. Persamaan untuk mendapatkan nilai AUC terdapat pada persamaan (2.33).

$$AUC = \frac{1}{2}\left(Sensitivity + Specivicity\right) \tag{2.33}$$

Nilai AUC adalah peluang metode klasifikasi akan memberikan peringkat *random positive test case* lebih tinggi dari *random negative test*. Nilai AUC terletak pada interval 0 hingga 1, sehingga jika semakin mendekati angka 1 maka nilai AUC semakin bagus (Zaky & Meira, 2014). Namun dalam praktiknya, nilai AUC bervariasi antara 0.5 hingga 1. Pada Tabel 2.3 ditunjukkan skala untuk interpretasi nilai AUC yang didapatkan (Gorunescu, 2011).

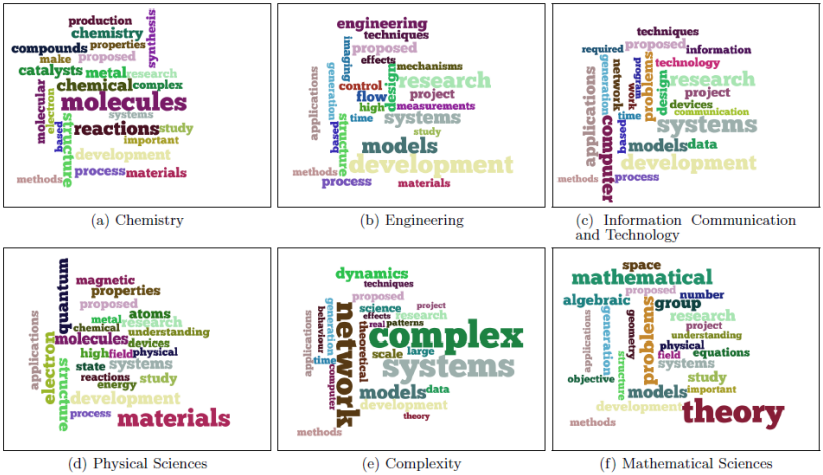
Tabel 2.3 Interpretasi Nilai AUC

Nilai AUC	Interpretasi
> 0.90 – 1.00	*Excellent Classification*
> 0.80 – 0.90	*Good Classification*
> 0.70 – 0.80	*Fair Classification*
> 0.60 – 0.70	*Poor Classification*
≥ 0.50 – 0.60	*Failure*

2.11 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang sering digunakan. *Word cloud* merupakan representasi grafis dari sebuah dokumen yang dilakukan dengan *plotting* kata-kata yang sering muncul pada sebuah dokumen pada ruang dua dimensi. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut. Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut

muncul dalam dokumen. Contoh dari visualisasi dokumen teks dengan *word cloud* ditunjukkan pada Gambar 2.5 (Castella & Sutton, 2014).



(a) Chemistry (b) Engineering (c) Information Communication and Technology

(d) Physical Sciences (e) Complexity (f) Mathematical Sciences

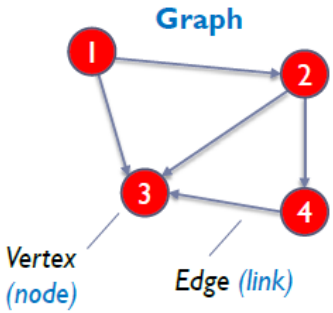
(Sumber: Castella & Sutton, 2014)

Gambar 2.5 Visualisasi Data dengan *Word Cloud*

2.12 *Social Network Analysis* (SNA)

Tsvetovat & Kouznetsov (2011) mengemukakan bahwa *Social Network Analysis* (SNA) dapat dideskripsikan sebagai studi yang mempelajari tentang hubungan manusia dengan memanfaatkan teori *graph*. Namun, Cheliotis (2010) menambahkan bahwa SNA bukan hanya sebuah metodologi, tetapi SNA adalah perspektif yang unik tentang bagaimana fungsi masyarakat. *Social Network Analysis* berfokus pada pola hubungan antar aktor serta menggambarkan hubungan jaringan tersebut. *Social Network Analysis* dapat digunakan untuk mempelajari pola jaringan organisasi, ide-ide, dan orang-orang yang terhubung melalui berbagai cara dalam sebuah lingkaran (Pfeil & Zaphiris, 2009). Menurut Wiki Book (2011), *Social Network Analysis* menganggap hubungan sosial dalam hal teori jaringan yang terdiri dari *nodes* dan *ties* (juga disebut *edges*, *links*, *or connections*).

Contoh visualisasi SNA disebut *sociogram* yang ditampilkan pada Gambar 2.6.



(Sumber: Cheliotis, 2010)

Gambar 2.6 Contoh *Sociogram* Sederhana

Dari pengertian tersebut dapat dikatakan bahwa SNA lebih menekankan pada interaksi antar entitas didalamnya daripada entitas itu sendiri, dengan kata lain SNA lebih banyak membahas hubungan antar aktor daripada atribut aktor tersebut . Pola interaksi antar entitas akan memberikan informasi baru. Namun bukan berarti entitas tidak ada gunanya sama sekali. Atribut pada entitas yang menjadi *node* pada graf memiliki informasi yang dapat membantu untuk membuat hipotesa atas fenomena yang terjadi. Sebagai contoh pada kasus jejaring sosial online di Twitter, yang lebih banyak dianalisis adalah interaksi antar *user* Twitter, dalam hal ini pola interaksi akan dapat menentukan *user* mana yang paling berpengaruh dalam suatu lingkup grup tertentu. SNA dapat memetakan relasi antar orang, organisasi, topik, lokasi, dan intensitas informasi lainnya. Node atau titik di dalam jaringan menggambarkan orang, organisasi, lokasi, atau entitas informasi. Garis sambungan antar titik menggambarkan relasi antar titik.

SNA didalam teori jaringan terdiri dari *node* dan *edge* (juga disebut relasi, link, atau koneksi). *Node* adalah seorang individu dalam jaringan, dan *edge* adalah hubungan antara *node*. Ada 2 macam graf yaitu *Graf Undirected* dan *Graf Directed*. *Graf Undirected* adalah graph yang hubungannya tidak mempunyai

orientasi arah. Pada *graf undirected*, nilai antar node yang dihubungkan oleh edge tidak diperhatikan, yang penting saling berhubungan atau berkoneksi maka memiliki nilai. *Graf directed* adalah graf yang setiap hubungan diberikan orientasi arah, dimana edgenya diperhatikan.

SNA dalam Twitter menampilkan peta relasi antar aktor di sosial media. Retweet menandakan tentang persetujuan (*agreement*), sedangkan mention menandakan pertanyaan atau diskusi (*discussion*). Ukuran (*metric*) yang digunakan dalam penentuan aktor dalam penelitian ini adalah *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigen vector centrality*.

A. *Degree Centrality*

Degree centrality didefinisikan sebagai jumlah *links* atau *ties* pada *node* (Wiki Book, 2011). Kemudian Cheliotis (2010) menjelaskan bahwa *degree centrality* berguna dalam menilai *node* yang *central* untuk menyebarkan informasi dan mempengaruhi orang lain dalam lingkungan mereka. Dalam sebuah jaringan dengan *g nodes*, *degree centrality* dari *node* ini adalah pada persamaan (2.34).

$$C_d(n_i) = d(n_i), \tag{2.34}$$

dimana, $d(n_i)$ adalah banyaknya garis atau *ties* yang ada atau terkait dalam jaringan. Suatu *node* mempunyai nilai *degree centrality* antara 0 hingga *g-1*. Untuk membedakan *centrality* dalam *node* dari jaringan dengan skala yang berbeda, normalisasi *degree centrality* diperlukan sehingga menjadi seperti pada persamaan (2.35).

$$C_d = \frac{d(n_i)}{g-1} \tag{2.35}$$

B. *Closeness Centrality*

Wiki Book (2011) mendeskripsikan *closeness centrality* sebagai *shortest-path length*. Cheliotis (2010) menambahkan bahwa *closeness* adalah rata-rata semua jalur terpendek dari satu *node* untuk semua *node* lain dalam jaringan. Ukuran dari

jangkauan, yaitu berapa lama akan mencapai *nodes* lain dari sebuah *node* awal. *Closeness centrality* berguna dalam kasus dimana kecepatan penyebaran informasi adalah perhatian utama. Dalam sebuah jaringan dengan *g nodes*, *closeness centrality* dari *node* n_i sehingga menjadi persamaan (2.36).

$$C_c(n_i) = \left[\left\{g_j =_1 d(n_i, n_j)\right\}\right]^{-1}, \tag{2.36}$$

dimana, $d(n_i, n_j)$ adalah jumlah *edge* yang terhubung dari n_i dan n_j. Sedangkan untuk membandingkan di jaringan dengan skala yang berbeda, *closeness centrality* di normalisasikan dengan mengalikan *g-1*.

$$C_c(n_i) = (g-1)C_c(n_i) \tag{2.37}$$

C. *Betweenness Centrality*

Cheliotis (2010) menjelaskan bahwa *betweenness centrality* adalah jumlah jalur terpendek yang melewati sebuah *node* dibagi dengan semua jalur terpendek dalam jaringan. *Betweenness centrality* menunjukkan *node* yang lebih cenderung berada dalam jalur komunikasi antara *node* lain. Dalam satu jaringan dengan *g nodes*, *betweenness centrality* untuk *node* didefinisikan sebagai jumlah dari jalur terpendek yang melalui *node*.

$$C_b(x) = \left\{\frac{_{jck} g_{jk}(n_i)}{g_{jk}}\right\}, \tag{2.38}$$

dimana, g_{jk} adalah jumlah jalur terpendek antara 2 *node* dalam jaringan g_{jk}, n_i adalah jumlah jalur terpendek dari *node j* ke *node k* melewati *node i*, dengan normalisasi pada persamaan (2.39).

$$C_B = \frac{C_b}{\left[(g-1)(g-2)/2\right]} \tag{2.39}$$

D. *Eigenvector Centrality*

Wiki Book (2011) menjelaskan bahwa *eigenvector centrality* adalah ukuran pentingnya sebuah *node* dalam jaringan. Dengan kata lain, sebuah *node* dengan *eigenvector centrality* yang tinggi terhubung ke *nodes* lain dengan *eigenvector centrality* yang

tinggi. Hal ini mirip dengan cara *Google* memeringkat halaman *web*. *Eigenvector centrality* berguna dalam menentukan siapa yang terhubung dengan *nodes* yang paling terhubung (Cheliotis, 2010). *Bonacich eigenvector centrality* terdapat pada persamaan (2.40) dan (2.41).

$$C_i(\beta) = \sum_j (\alpha + \beta c_j) A_{ji} \tag{2.40}$$

$$C(\beta) = \alpha(I - \beta A)^{-1} A1, \tag{2.41}$$

dimana, α adalah konstanta normalisasi (skala vektor), β melambangkan seberapa banyak suatu *node* mempunyai bobot *centrality* dalam *node* yang terikat. *A* adalah *adjacency matrix*, *I* adalah *identity matrix* dan *1* adalah *matrix*. Besarnya β adalah *radius power* dari suatu *node*. Jika β positif, maka mempunyai ikatan *centrality centrality* yang tinggi dan terhubung dengan orang-orang yang *central*. Sedangkan jika β negatif, maka mempunyai ikatan *centrality* tinggi namun terhubung dengan orang-orang yang tidak *central*. Jika $\beta = 0$, maka akan didapat *degree centrality*.

2.13 Media Sosial

Pada dasarnya, media sosial merupakan perkembangan mutakhir dari teknologi-teknologi *web* baru berbasis internet, yang memudahkan semua orang untuk berkomunikasi, berpartisipasi, saling berbagi, dan membentuk sebuah jaringan secara *online*. Media sosial mempunyai banyak bentuk, diantaranya yang paling populer yaitu *microblogging* (Twitter). Twitter adalah suatu situs *web* yang merupakan layanan dari *microblog*, yaitu suatu bentuk *blog* yang membatasi ukuran setiap *post*-nya, yang memberikan fasilitas bagi pengguna untuk dapat menuliskan pesan.Twitter merupakan salah satu jejaring sosial yang paling mudah digunakan, karena hanya memerlukan waktu yang singkat tetapi informasi yang disampaikan secara luas (Twitter, 2018).

Twitter pertama kali ada pada tanggal 21 Januari 2000 di San Fransisco, California. *Microblogging* Twitter hanya mampu mengantar pesan pendek, dengan panjang maksimal 140 karakter

untuk setiap pesan. Twitter merupakan jejaring sosial besar yang fokus pada kecepatan komunikasi. Kecepatan dan kemudahan dalam hal publikasi pesan membuat Twitter menjadi media komunikasi yang penting. Kumar, Morstatter, & Liu (2013) mengatakan bahwa terdapat sekitar 140 juta pengguna aktif yang membuat lebih dari 400 juta pesan setiap hari.

Twitter dapat digunakan oleh pengguna untuk mempublikasikan pesan (‘*tweeting*’) dengan sangat cepat dan mudah. Pengguna dapat terhubung dengan pengguna lain melalui fitur ‘*follow*’, sehingga pengguna dapat mengikuti *tweet* terbaru dari pengguna yang di *follow*. Perlu diperhatikan bahwa mekanisme *follow* tidak mewajibkan pengguna lain untuk melakukan *follow* balik. Istilah lain yang cukup populer diantaranya adalah RT atau *retweet*, dengan simbol ‘@’ yang diikuti nama pengguna. *Retweet* merupakan sarana membalas *tweet* dengan menyertakan isi *tweet* sumber, sehingga pengguna yang menerima *retweet* bisa memahami konteks pesan yang diterima. Selain itu, terdapat simbol ‘#’ yang diikuti sebuah kata yang merepresentasikan sebuah *hastag*. *Opsi* ini penting untuk menandai konteks dari sebuah pesan Twitter, namun *hastag* bukanlah syarat publikasi *tweet*.

Terdapat beberapa alasan pesan Twitter digunakan sebagai sumber penelitian, diantaranya adalah frekuensi *posting* pesan yang sangat tinggi, pesan Twittertidak terlalu panjang hanya 140 karakter, sehingga lebih deskriptif dan mudah dimengerti. Selain itu, Twitter menyediakan *semu-structuredmeta-data* (kota, negara, jenis kelamin, dan umur) (Culotta, 2010).

2.14 Transportasi Umum Taksi

Menurut Peraturan Pemerintah Republik Indonesia No 41 Tahun 1993, Taksi adalah kendaraan umum dengan jenis mobil penumpang yang diberi tanda khusus dan dilengkapi dengan argometer. Sistem pelayanan taksi bersifat fleksibel bila dibandingkan moda angkutan lainnya dan memiliki pelayanan dari pintu ke pintu (*door to door service*) dalam wilayah operasi

terbatas. Dengan kata lain, bahwa taksi memiliki kelebihan utama pada pelayanan angkutan umum, bila dilihat dari keleluasaan waktu yang tidak terjadwal, rute pelayanan dan tempat pemberhentiannya yang bebas, serta dilengkapi dengan argometer.

Taksi merupakan salah satu jenis layanan transport yang mempunyai karakteristik pelayanan khusus, yang merupakan perpaduan antara kendaraan pribadi dan angkutan umum (Siswanto, Riyanto, & Sunaryo, 2010). Karena taksi dapat melayani ke semua tempat di daerah urban dan dapat dipanggil melalui telepon serta mampu memberikan pelayanan perjalanan secara pribadi, sehingga taksi cenderung merupakan kendaraan pribadi daripada kendaraan umum. Salah perusahaan yang bisnis utamanya bergerak di bidang jasa transportasi umum atau taksi adalah PT Blue Bird Group.

Dalam perkembangannya, taksi banyak mengalami perubahan dari segi operasional dan investasi. Adapun fenomena baru yang terjadi pada angkutan umum ini yaitu fenomena taksi berbasis *online* atau biasa disebut dengan aplikasi *ride sharing*. Berbeda dengan taksi konvensional biasanya, taksi berbasis aplikasi ini menggunakan mobil berplat warna hitam. Aplikasi ini mempertemukan para calon penumpang dengan supir dengan menggunakan perkembangan teknologi GPS dan menawarkan tarif yang relatif lebih murah dibandingkan dengan taksi konvensional. Adapaun perusahaan pengembang metode aplikasi ini sudah memasuki wilayah Indonesia, yaitu GoCar, GrabCar, dan sebagainya (Ilmawan, 2017).

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan kumpulan *tweet* dari pengguna Twitter di Indonesia dengan *keywords* dari masing-masing taksi konvensional dan taksi berbasis *online*. Data yang digunakan dalam penelitian ini baik untuk tahapan awal adalah data yang terkumpul dari hasil *crawling* 3 *account* Twitter, yaitu BlueBird Taxi (@Bluebirdgroup) sebagai taksi konvensional, serta GoCar (@gojekindonesia) dan GrabCar (@GrabID) sebagai taksi berbasis *online*. Data tersebut diambil dari tanggal 5 Februari 2018 hingga 5 Mei 2018 dengan menggunakan Twitter API (*Application Programming Interface*).

3.2 Struktur Data

Data yang didapatkan pada setiap media dibagi menjadi data *training* dan data *testing* dengan perbandingan 90%:10%. Struktur data yang digunakan dalam penelitian ini setelah dilakukan praproses pada data teks *tweet* terdiri dari variabel prediktor yaitu kata dasar setiap *tweet* dan variabel respon yaitu klasifikasi sentimen *tweet* (positif dan negatif). Struktur data yang diambil dari *website www.twitter.com* dengan bantuan Twitter API *software* R 3.3.3 pada Tabel 3.1.

Tabel 3.1 Struktur Data Penelitian Sebelum *Pre-processing*

No.	*Tweet* (*t*)	*Reply to SN*	*Created*	*Screen Name*	Klasifikasi Sentimen (*y*)
1	t_1	$reply_1$	$date_1$	$account_1$	y_1
2	t_2	$reply_2$	$date_2$	$account_2$	y_2
.	.	.	.	.	.
.	.	.	.	.	.
n	t_n	$reply_n$	$date_n$	$account_n$	y_n

Notasi *n* menunjukkan jumlah data yang digunakan dalam penelitian. Kolom *Reply to SN* menunjukkan kepada siapa *tweet*

tersebut ditujukan. Kolom *Created* menunjukkan waktu *tweet* tersebut diunggah. Kolom *Screen Name* menunjukkan *account* yang mengunggah *tweet*. Kolom Klasifikasi Sentimen merupakan hasil pengkategorian *tweet* yang bersentimen yang dikategorikan positif atau negatif.

Setelah dilakukan *pre-processing* data, struktur data yang terbentuk terdapat pada Tabel 3.2. Pada Tabel 3.2, notasi k menunjukkan banyak kata kunci yang terbentuk, sehingga menunjukkan banyak kata kunci berjalan hingga w_k yaitu kata kunci yang didapatkan dari hasil data *tweet* yang telah ditentukan. Struktur data pada Tabel 3.2 dapat digunakan untuk analisis lebih lanjut, yaitu untuk analisis *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM).

Tabel 3.2 Struktur Data Setelah *Pre-processing*

No.	***Screen Name***	***Tweet* (*t*)**	**Klasifikasi Sentimen (*y*)**	**Kata Kunci (w_1)**	**Kata Kunci (w_2)**	**...**	**Kata Kunci (w_k)**
1	$account_1$	t_1	y_1	$w_{1,1}$	$w_{1,2}$	...	$w_{1,k}$
2	$account_2$	t_2	y_2	$w_{2,1}$	$w_{2,2}$	...	$w_{2,k}$
.	.	.	.	.	.	...	.
.	.	.	.	.	.	...	.
n	$account_n$	t_n	y_n	$w_{n,1}$	$w_{n,2}$	...	$w_{n,k}$

Selain dilakukan analisis dengan menggunakan metode *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM), pada penelitian ini dilakukan pula analisis *Social Network Analysis* (SNA) dengan struktur data yang terdapat pada Tabel 3.3. *Node* adalah seorang individu dalam jaringan, dan *edge* adalah hubungan antara *node*. Pada kolom *Label* menunjukkan daftar-daftar *account* Twitter yang terlibat dalam topik yang diambil oleh peneliti dan kemudian diberi nomor ID pada kolom ID. Sedangkan, kolom *Source* menunjukkan *account* Twitter yang merupakan pengunggah atau penulis *tweet* yang ditujukan pada *account* Twitter pada kolom Target.

Tabel 3.3 Struktur Data *Social Network Analysis*

Nodes		***Edges***	
ID	***Label***	***Source***	**Target**
1	$account_1$	$IDSource_1$	$IDTarget_1$
2	$account_2$	$IDSource_2$	$IDTarget_2$
.	.	.	.
.	.	.	.
n	$account_n$	$IDSource_n$	$IDTarget_n$

3.3 Langkah Analisis

Pada tahap *pre-processing* proses-proses yang dilakukan yaitu tahap klasifikasi sentimen terhadap *tweet* yang sudah diketahui mengandung positif atau negatif. Tujuan dari tahap *pre-processing* adalah untuk mencari *keyword* beserta probabilitasnya yang nantinya akan digunakan pada proses *testing*. Adapun langkah-langkah yang dilakukan adalah sebagai berikut.

1. Mengambil data *tweet* dengan menggunakan Twitter API.
   a. Memasukkan *keyword* yang berhubungan dengan BlueBird Taxi,GoCar, dan GrabCar.
   b. Menyimpan hasil *searching* ke *database*.
2. Melakukan klasifikasi sentimen pada data *tweet* dengan sentimen positif atau negatif dan menghapus *tweet* yang tidak memiliki sentimen atau netral.
3. Menyiapkan data *tweet*, daftar *stopwords*, dan kata dasar.
   a. Daftar *stopwords* didapatkan dari tesis F. Z. Tala yang berjudul "*A Study of Stemming effect on Information Retrieval in Bahasa Indonesia"* (Tala, 2003).
   b. Kata dasar diambil dari Kamus Besar Bahasa Indonesia.
4. Praproses teks.
   a. Melakukan *cleansing*, yaitu proses pembersihan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam *tweet* adalah karakter HTML, *emoticons*, *hastag(#)*, *username (@username)*, simbol *retweet* (*response tweet*) "RT", dan *link* URL.
   b. Melakukan *case folding*, yaitu mengubah semua teks dengan huruf kecil (non kapital) serta menghilangkan tanda baca.

c. Melakukan *stemming* pada kata-kata yang tersisa pada *tweet* untuk mendapatkan kata dasar. Pada tahap ini dilakukan algoritma *confix-stripping stemmer* untuk mendapatkan kata dasar. Untuk membandingkan hasil pemenggalan kata dengan kata dasar maka proses ini membutuhkan kamus kata dasar yang telah disiapkan sebelumnya.
d. Melakukan proses *stopping* berdasarkan *stoplist* yang berisi *stopwords* yang telah ditentukan sebelumnya. Kata-kata yang terdapat pada *tweet* akan dibandingkan dengan daftar *stopwords*, jika terdapat kata-kata yang terdapat pada *stopwords* maka kata tersebut akan dihapus dari *tweet* sehingga ditemukan kata kunci yang identik.
e. Melakukan *tokenizing* untuk memecah *tweet* menjadi kata per kata.

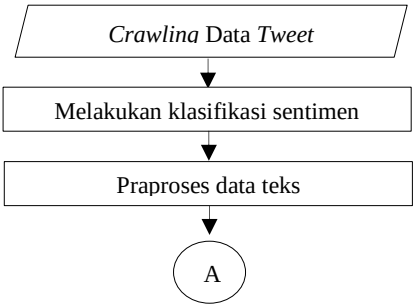
5. Merubah teks menjadi variabel dengan pembobotan kata dengan *tf-idf* menggunakan persamaan (2.1).
6. Data *tweet* dibagi menjadi data *training* dan data *testing* dengan perbandingan *training-testing* sebesar 90%:10% dengan menggunakan *k-fold cross validation*.
7. Klasifikasi data menggunakan *Naïve Bayes Classifier* untuk masing-masing taksi.
   a. Menghitung probabilitas dari V_j pada data *training*, dimana V_j merupakan kategori sentimen, yaitu V_1 = negatif dan V_2 = positif.
   b. Menghitung probabilitas kata w_k pada kategori V_j.
   c. Model probabilitas NBC disimpan dan digunakan untuk tahap data *testing*.
   d. Menghitung probabilitas tertinggi dari kategori sentimen yang diujikan (V_{MAP}) dengan persamaan (2.9).
   e. Mencari nilai V_{MAP} paling maksimum dan memasukkan *tweet* tersebut pada kategori dengan V_{MAP} maksimum.
   g. Menghitung nilai akurasi dari model yang terbentuk.
8. Klasifikasi data menggunakan *Support Vector Machine* untuk masing-masing taksi.

a. Menentukan parameter pada SVM tiap jenis Kernel.
b. Membangun model SVM menggunakan fungsi Kernel Linier dan *Radial Basis Function* (RBF).
c. Menghitung nilai akurasi dari model yang terbentuk.
d. Membandingkan tingkat akurasi dari kedua model.

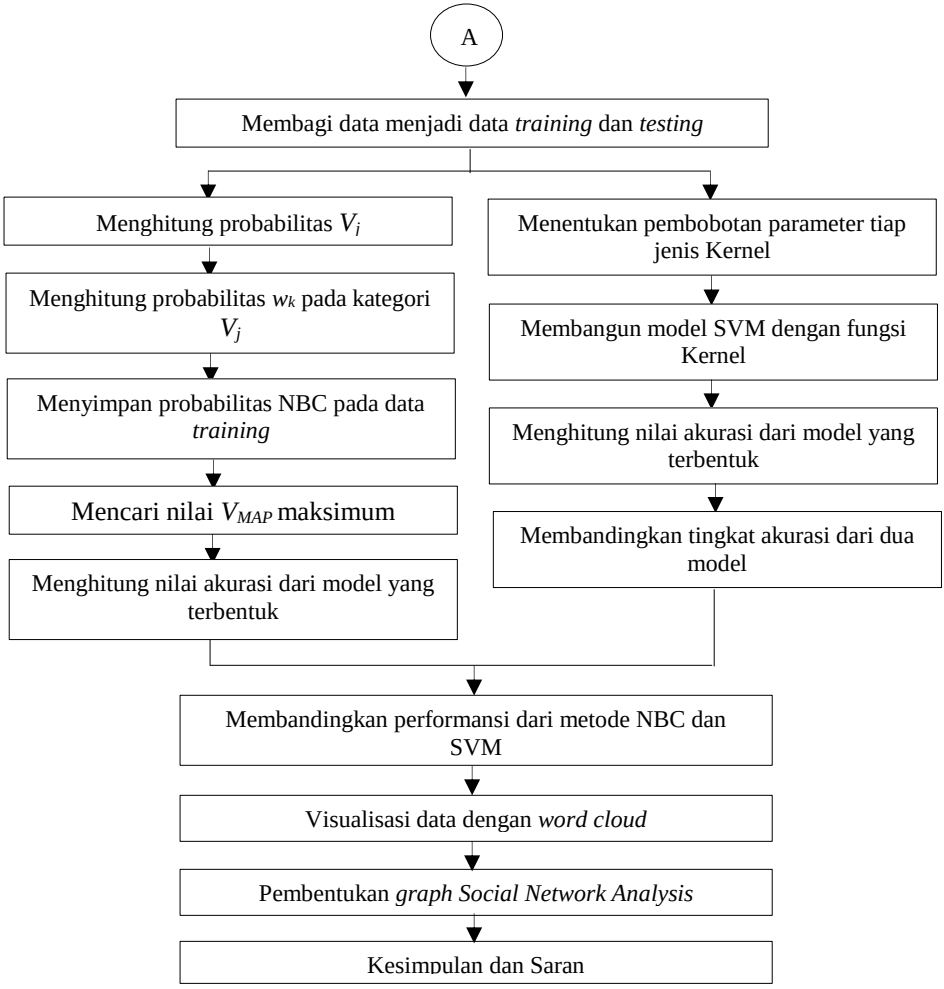
9. Evaluasi hasil klasifikasi
   Menghitung ketepatan klasifikasi dan membandingkan performansi metode NBC dan SVM berdasarkan tingkat *accuracy*, *sensitivity*, dan *specificity* dengan persamaan (2.24), (2.25), dan (2.26) jika data *balanced*, atau berdasarkan nilai *G-mean* dan AUC dengan persamaan (2.30) dan (2.31) jika data *imbalanced*.
10. Melakukan visualisasi *tweet* dengan *word cloud*.
11. Menentukan *Social Network Analysis* untuk menentukan pola jaringan-jaringan antar pengguna pada setiap taksi berdasarkan pengguna yang muncul pada data *tweet*. Didapatkan dengan menggunakan *software* Gephi, yaitu *software* khusus dalam menganalisis *Social Network Analysis*.
12. Melakukan interpretasi dan menarik kesimpulan.

3.4 Diagram Alir

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



Gambar 3.1 Diagram Alir Penelitian



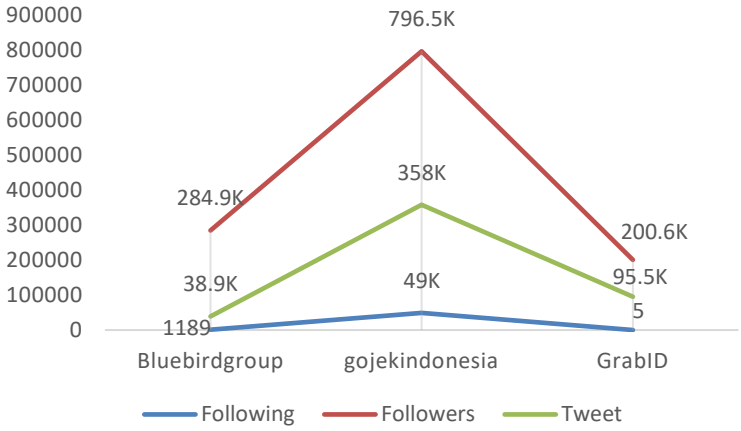
Gambar 3.1 Diagram Alir Penelitian (Lanjutan)

BAB IV
ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas hasil analisis dari pengolahan data yang telah dilakukan. Metode yang digunakan dalam analisis pada penelitian ini adalah klasifikasi dengan metode *Naïve Bayes Classifier* dan *Support Vector Machine* menggunakan data *tweet* yang telah didapatkan dengan Twitter API. Sebelum menganalisis, dilakukan praproses teks. Selain itu, dilakukan pula analisis dengan *Social Network Analysis* (SNA).

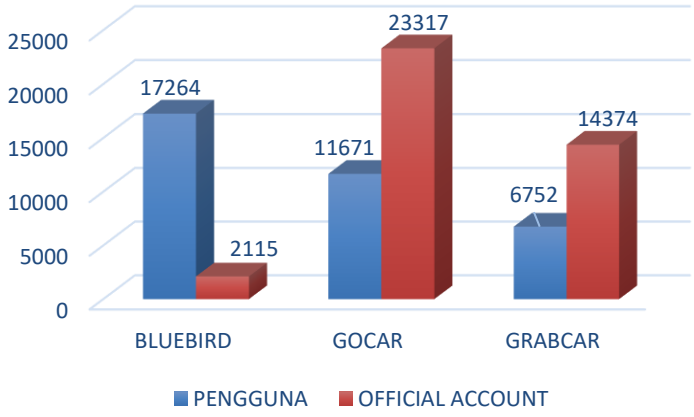
4.1 Karakteristik Data

Sebelum melakukan analisis lebih lanjut, terlebih dahulu dilakukan analisis untuk karakteristik data. Pada penelitian ini, data yang digunakan merupakan data *tweet* yang didapatkan dari hasil *crawling* dengan menggunakan Twitter API. Sebelum beranjak pada analisis karakteristik data yang digunakan, terlebih dahulu akan dilakukan perbandingan sosial media Twitter dari masing-masing penyedia jasa layanan transportasi umum, yaitu @Bluebirdgroup, @gojekindonesia, dan @GrabID.



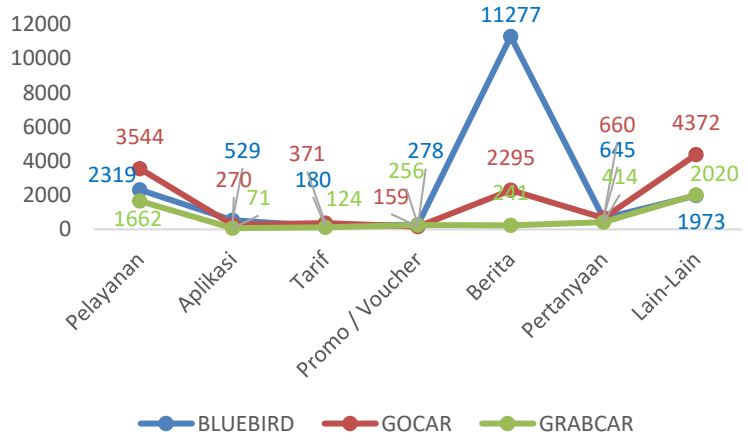
Gambar 4.1 Perbandingan Antar *Official Account* Twitter Tiap Taksi

Berdasarkan pada Gambar 4.1, dari data yang didapatkan per tanggal 5 Mei 2018, didapatkan bahwa *official account* dari @gojekindonesia merupakan *official account* dengan jumlah *followers* dan *tweet* tertinggi jika dibandingkan dengan *official account* @Bluebirdgroup dan @GrabID. Hal tersebut disebabkan karena pelayanan yang diberikan oleh @gojekindonesia bukan hanya sekedar pelayanan pada jasa transportasi umum taksi GoCar saja, namun adapula GoJek (pelayanan jasa transportasi umum motor ojek), GoFood (pelayanan jasa untuk pembelian makanan), GoSend (pelayanan antar barang), dan lain sebagainya. Pada @GrabID juga terdapat pelayanan yang sama, namun keaktifan @GrabID di media sosial Twitter masih jauh dibandingkan dengan @gojekindonesia. Sedangkan untuk @Bluebirdgroup, jumlah *tweet* pada *official account* tergolong paling sedikit jika dibandingkan dengan yang lainnya karena pelayanan yang diberikan oleh @Bluebirdgroup hanya sebatas pelayanan jasa transportasi umum taksi secara *offline* maupun *online*. Berikut merupakan data banyak *tweet* yang didapatkan dari periode 5 Februari 2018 hingga 5 Mei 2018 dan disajikan pada Gambar 4.2.



Gambar 4.2 Perbandingan Banyak Data *Tweet* Dari Tiap Taksi

Dari Gambar 4.2 dapat diketahui bahwa banyak *tweet* yang didapatkan dari periode tanggal 5 Februari 2018 hingga 5 Mei 2018 dengan menggunakan *keyword* BlueBird Taxi yaitu sebanyak 17264 *tweet*, dengan *keyword* GoCar sebanyak 11671 *tweet*, dan dengan *keyword* GrabCar didapatkan sebanyak 6752 *tweet*. Selain itu dapat pula dilihat perbandingan banyaknya *tweet* yang dilakukan oleh masing-masing *official account*. *Tweet* terbanyak dilakukan oleh *official account* dari penyedia jasa layanan GoCar yaitu @gojekindonesia. Namun perlu diketahui, *tweet* yang diunggah oleh @gojekindonesia bukan saja tentang GoCar, akan tetapi bercampur dengan pelayanan yang lain. Sehingga wajar saja jika @gojekindonesia merupakan *official account* dengan jumlah *tweet* terbanyak. Jika dibandingkan dengan *official account* dari penyedia jasa layanan BlueBird Taxi memang sangat berbeda jauh, karena pelayanan yang diberikan oleh pihak @Bluebirdgroup hanya sebatas pelayanan jasa transportasi umum secara *offline* dan *online*. Dari Gambar 4.2 dapat dilakukan analisis lebih detail untuk mengetahui jenis *tweet* apa saja yang didapatkan seperti ditampilkan pada Gambar 4.3.



Gambar 4.3 Kategori *Tweet* Pengguna

Gambar 4.3 menunjukkan beberapa macam *tweet* dari pengguna yang dibagi dalam beberapa kategori yang telah ditentukan, yaitu tentang pelayanan, aplikasi, tarif, promo atau voucher, berita, pertanyaan, dan lain-lain. Dapat diketahui bahwa sebagian besar *tweet* yang didapatkan dengan menggunakan *keyword* BlueBird Taxi adalah pada kategori berita yaitu sebanyak 11277 *tweet*. Sedangkan, untuk *tweet* dari pengguna tentang pelayanan BlueBird Taxi hanya sebanyak 2319 *tweet*. Kemudian pada *keyword* GoCar didapatkan sebagian besar dari pengguna merupakan *tweet* yang tidak termasuk dalam beberapa kategori yang diberikan atau dapat dikatakan termasuk dalam kategori *tweet* dengan kategori lain-lain yaitu sebanyak 4372 *tweet*. Hal tersebut dikarenakan sebagian besar pengguna Twitter menggunakan kata GoCar sebagai bahan bercanda atau hanya sekedar istilah pada *tweet* yang diunggahnya, sehingga tidak termasuk dalam beberapa kategori tentang pelayanan atau hal lain yang berhubungan dengan GoCar. Namun, dari keseluruhan *tweet* didapatkan sebanyak 3544 *tweet* dari *tweet* pengguna yang menggunggah tentang pelayanan dari jasa penyedia transportasi umum taksi GoCar.

Selanjutnya pada *keyword* GrabCar, pengguna Twitter sebagian besar menggunggah *tweet* yang tidak termasuk dalam beberapa kategori yang telah ditentukan, yaitu sebanyak 2020 *tweet*. Penyebab dari munculnya hal tersebut sama seperti pada jenis taksi GoCar, yaitu sebagian besar para pengguna Twitter menggunakan kata GrabCar hanya sebagai kata kiasan ataupun hanya sekedar sebagai bahan bercanda, sehingga *tweet* tersebut tidak dapat dikategorikan ke dalam jenis *tweet* tentang pelayanan, aplikasi, tarif, dan sebagainya. Akan tetapi, pada kategori pelayanan didapatkan jumlah *tweet* sebanyak 1662 *tweet*. Jumlah tersebut lebih besar jika dibandingkan dengan kategori-kategori lainnya Sehingga dapat dikatakan bahwa kategori pelayanan juga mendominasi pada *tweet* yang diunggah oleh pengguna GrabCar. Berikut akan diberikan beberapa contoh *tweet* untuk setiap kategori beserta contoh *tweet* dari *official account* dari masing-masing

penyedia jasa transportasi umum taksi yang disajikan pada Tabel 4.1.

Tabel 4.1 Contoh Isi *Tweet* Tiap Kategori

Kategori	Taksi	*Tweet*
Official Account	BlueBird	@W_Thok Terima kasih Pak Wahyu atas kepercayaannya dan apresiasinya kepada pelayana taksi CCD7431. Semoga pengemudi… https://t.co/8alltuR8uJ
	GoCar	@remahremahastor Hai Dessy, silakan informasikan nomor order yg Anda keluhkan dan detail kronologi Anda melalui DM. Tks ^yun
	GrabCar	@AsmaraAra Hai Kak, saya Gita dari Grab Indonesia. Terima kasih sudah berpartisipasi dan mendukung program… https://t.co/HlPNRm8EYH
Pelayanan	BlueBird	Kami sangat apresiasi sekali dgn pengemudi @Bluebirdgroup ini. Bertanggung jawab & ikhlas memutar ruter menjemput p… https://t.co/dh1UHdWrQO
	GoCar	Gue berterimakasih bgt sama bapak driver gocar yg td plus mamas2 driver gojek yg anterin ke hotel trs ke bandara ju… https://t.co/xgyFd52jUJ
	GrabCar	@GrabID Hai,sy mau komplen kejadian barusan jm 16.12. Psn grabcar utk papa,di cek ud in transit dan… https://t.co/cVJU9dsPrl
Aplikasi	BlueBird	Fitur Easy Ride di @Bluebirdgroup sangat menyenangkan, kalau saya lagi nggak pegang uang cash.
	GoCar	Setelah update @gojekindonesia di appstore tampilan pemesanan berubah. Jujur tampilan baru sangat tdk mendukung use… https://t.co/UItEQWHC6r

Tabel 4.1 Contoh Isi *Tweet* Dari BlueBird Taxi (Lanjutan)

Kategori	Taksi	*Tweet*
Aplikasi	GrabCar	Secara user interface , grab car memiliki desain yang lebih elegan dibandingkan dengan taksi lainnya #katague
Tarif	BlueBird	Hai @bluebirdgroup! Tarifnya skrg semakin mahal. Ada kenaikan charge argo kah?? Biasanya estimasi dr Mampang Depok… https://t.co/AQnY894Dsr
	GoCar	Go ride sekarang mahal yah... Biasa pake gopay cuma 6K jadi 10K. Malah murah naik gocar masaa.. @gojekindonesia https://t.co/I3xc7YONKL
	GrabCar	lama tak naik grab tapi kenapa grabcar mahal gila ah skrg.
Promo atau Voucher	BlueBird	Coba e-voucher @Bluebirdgroup.. Mantap.. Lihat perjalanan saya menggunakan Blue Bird: https://t.co/gYvNdDzBsN
	GoCar	Enakny pakai GO Car pakai voucher. Jarak jauh dekat cuma segitu :) #gocar #gojek #gocar @… https://t.co/v4KvFRwUIt
	GrabCar	Grab ini makin lama makin pelit "Diskon 10% utk 10x naik GrabCar tp maks Rp3ribu"
Berita	BlueBird	Blue Bird Siap Hadapi Transportasi Online https://t.co/HY8qBgwvLm
	GoCar	SOLOPOS HARI INI : Gocar Mogok, Tarif Melonjak 300% https://t.co/70WpNZVloM
	GrabCar	Taksi Online GrabCar Resmi Beroperasi di Bandara Husein Sastranegara Bandung https://t.co/P9ArrLe07g
Pertanyaan	BlueBird	@Bluebirdgroup kalo pesan taxi di area bandara via aplikasi mybluebird apakah kena tambahan service charge?
	GoCar	@gojekindonesia saya mau bergabung menjadi driver go car.tapi mobil saya masih plat putih atau profit.stnk nya belu… https://t.co/GeARu9vdu7

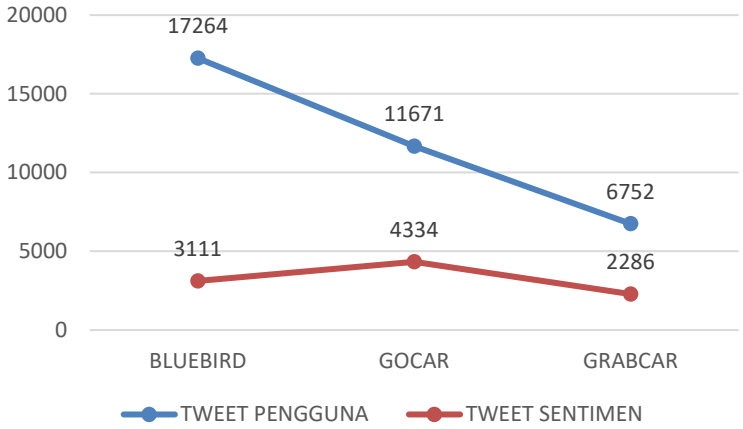
Tabel 4.1 Contoh Isi *Tweet* Dari BlueBird Taxi (Lanjutan)

Kategori	Taksi	*Tweet*
Pertanyaan	GrabCar	@GrabID min sy tadi naik grabcar terus ada barang ketinggalan di mobil sedangkan gaada nomer drivernya. kode booking ADR99616093499 gmn min?
Lain-Lain	BlueBird	@otong_leyong Nasib bluebird (dan taksi non online) semua bergantung permenhub baru mengenai layanan taksi online.… https://t.co/nk4PKFlnc1
	GoCar	RT @nadiarvh: Aku : Pak, saya minta turun di lobby south ya. Driver gocar: Mba saya turunin di west ya. Saya malu nemenin mba, mba cantik,…
	GrabCar	Gue sih #TimDudukBelakang, supaya kalo gue inget mantan terus sedih, ga ketauan sama mitra GrabCar. Kan malu. Wkwk https://t.co/ZSmPNcy9HV

Dari semua *tweet* pengguna jasa transportasi umum taksi yang didapatkan pada Gambar 4.2 tidak semua *tweet* mengandung sentimen, sehingga peneliti melakukan langkah awal dengan melakukan klasifikasi sentimen dengan dua kategori yaitu sentimen positif dan sentimen negatif pada masing-masing *tweet* pengguna yang didapatkan secara manual dan berdasarkan dari persepsi peneliti. Sehingga didapatkan banyak *tweet* yang mengandung sentimen positif atau sentimen negatif yang disajikan pada Gambar 4.4.

Pada Gambar 4.4 dapat diketahui bahwa dari 17264 *tweet* pengguna yang mengunggah *tweet* menggunakan *keyword* BlueBird Taxi didapatkan 3111 *tweet* yang mengandung sentimen. Kemudian dapat diketahui pula dari *tweet* pengguna yang mengunggah *tweet* dengan *keyword* GoCar yaitu sebanyak 11671 *tweet*. Namun setelah dilakukan klasifikasi sentimen, ternyata hanya didapatkan sebanyak 4334 *tweet* pengguna yang mengandung sentimen. Sedangkan pada *tweet* pengguna yang menggunakan *keyword* GrabCar, yaitu sebanyak 6752 *tweet*

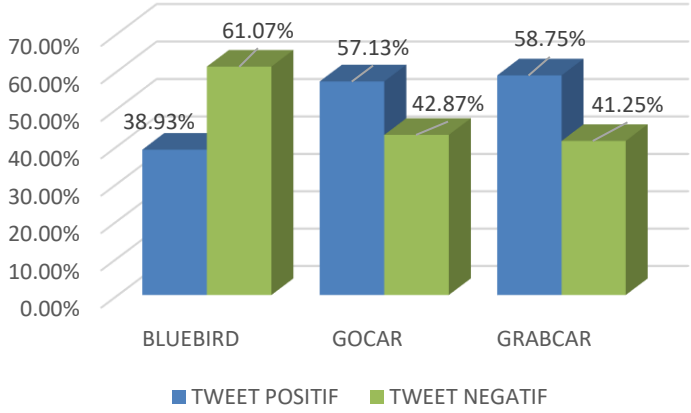
pengguna yang didapatkan, hanya terdapat sebanyak 2286 *tweet* yang mengandung sentimen. Berdasarkan dari Gambar 4.4, maka didapatkan perbandingan antara ketiga jenis penyedia jasa transportasi umum taksi yang disajikan pada Gambar 4.5.



Gambar 4.4 *Tweet* Sentimen

Berdasarkan Gambar 4.4 dapat diketahui bahwa banyak *tweet* yang mengandung sentimen yang didapatkan merupakan *tweet* tentang jasa transportasi umum taksi GoCar dibandingkan dengan BlueBird ataupun GrabCar. *Tweet* yang diunggah oleh para pengguna tentang jenis taksi GoCar jika berdasarkan dari Gambar 4.5 lebih banyak mengarah ke hal yang positif daripada hal negatif. Jika dibandingkan antara jenis taksi BlueBird Taxi dan GrabCar, berdasarkan Gambar 4.4 pengguna lebih sering mengunggah *tweet* yang mengandung sentimen tentang BlueBird Taxi daripada tentang GrabCar. Namun berdasarkan Gambar 4.5, sebagian besar *tweet* yang diunggah oleh pengguna tentang BlueBird Taxi lebih mengarah pada sentimen negatif daripada sentimen positif. Akan tetapi pada jenis taksi GrabCar, walaupun pengguna jarang mengunggah *tweet* tentang GrabCar namun terlihat pada Gambar 4.5 bahwa *tweet* yang mengandung sentimen yang diunggah oleh

pengguna lebih banyak mengarah pada hal positif daripada negatifnya. Berdasarkan Gambar 4.5, dapat diketahui pula bahwa perbandingan *tweet* yang mengandung sentimen positif dan sentimen negatif dari masing-masing penyedia jasa transportasi umum taksi cenderung *balanced* atau seimbang antara *tweet* yang mengandung sentimen positif dan *tweet* yang mengandung sentimen negatif.



Gambar 4.5 Perbandingan *Tweet* Sentimen Antar Jenis Taksi

Berdasarkan Gambar 4.4 dan Gambar 4.5 didapatkan bahwa dari *tweet* sentimen yang diperoleh dari topik tentang taksi konvensional dan taksi berbasis *online* didapatkan bahwa pengguna lebih banyak mengunggah *tweet* bersentimen positif pada taksi berbasis *online* baik itu GoCar maupun GrabCar jika dibandingkan dengan taksi konvensional BlueBird Taxi. Hal tersebut dapat menunjukkan bahwa pengguna sebagian besar telah merasa terpuaskan dengan jasa yang diberikan sehingga memberikan respon yang positif terhadap jasa yang telah diberikan dengan mengunggah *tweet* bersentimen positif pada taksi berbasis *online* GoCar dan GrabCar. *Tweet* yang diunggah dapat berupa jasa dari segi pelayanan secara langsung dari sopir atau *customer*

service, jasa dari aplikasi yang telah disediakan, promo atau voucher yang diberikan, dan lain-lain. Sedangkan, jasa dari taksi konvensional BlueBird Taxi sebagian besar belum dapat memuaskan pelanggan dari segi pelayanan dari sopir atau *customer service*, jasa dari aplikasi yang disediakan, promo atau voucher yang diberikan, dan lain-lain sehingga pengguna taksi konvensional BlueBird Taxi lebih cenderung memberikan respon yang negatif dengan mengunggah *tweet* bersentimen negatif. Dalam hal ini, kepuasan pengguna terhadap jasa yang diberikan oleh taksi berbasis *online* lebih baik jika dibandingkan dengan taksi konvensional.

Setelah didapatkan *tweet* yang mengandung sentimen dari masing-masing jenis taksi, maka dapat dilakukan analisis selanjutnya yaitu klasifikasi dengan menggunakan metode *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) pada data *tweet* yang telah diklasifikasikan mengandung sentimen positif ataupun negatif. Namun sebelumnya akan dilakukan praproses teks.

4.2 Praproses Teks

Data *tweet* mengenai taksi konvensional BlueBird Taxi dan taksi berbasis *online* GoCar dan GrabCar yang telah terkumpul dilakukan praproses teks yang meliputi *cleansing*, *case folding*, *stemming*, *stopword*, dan *tokenizing*. Penjelasan mengenai hasil praproses teks dari setiap tahapan akan dijabarkan pada simulasi praproses teks pada sebuah data *tweet*. *Tweet* yang akan digunakan sebagai contoh adalah dua contoh *tweet* dari BlueBird Taxi. Tahapan dan hasil praproses menggunakan salah satu *tweet* tersebut ditunjukkan pada Tabel 4.2. Pada Tabel 4.2 menunjukkan contoh *tweet* yang melalui tahapan praproses teks hingga didapatkan hasil akhir pada proses *stopwords* dan *tokenizing*. Praproses teks yang telah dilakukan bertujuan untuk meningkatkan ketepatan klasifikasi dan mengurangi kesalahan klasifikasi data.

Tabel 4.2 Simulasi Praproses Teks

Tahapan	Hasil Praproses
Contoh *tweet* 1	@Bluebirdgroup Itu yang saya males.. Ribet pake telp... Akun saya juga gak bisa login..... Nunggu yang lewat aja lah.....
Contoh *tweet* 2	@Bluebirdgroup luar biasa inisiatif driver Bapak Rudyanto. Membuat penumpang nyaman dan Tissue, bolpen,… https://t.co/x4fo6xFyp
Cleansing tweet (menghapus simbol RT, *username*, dan *link* URL)	Itu yang saya males.. Ribet pake telp... Akun saya juga gak bisa login..... Nunggu yang lewat aja lah.....
	luar biasa inisiatif driver Bapak Rudyanto. Membuat penumpang nyaman dan Tissue, bolpen,
Case folding	itu yang saya males ribet pake telp akun saya juga gak bisa login nunggu yang lewat aja lah
	luar biasa inisiatif driver bdu2412 bapak rudyanto membuat penumpang nyaman dan tissue bolpen
Stemming	itu yang saya males ribet pake telp akun saya juga gak bisa login nunggu yang lewat aja lah
	luar biasa inisiatif driver bapak rudyanto buat tumpang nyaman dan tissue bolpen
Stopwords dan *Tokenizing*	['males', 'ribet', 'pake', 'telp', 'akun', 'login', 'nunggu']
	['inisiatif', 'driver', 'rudyanto', 'tumpang', 'nyaman', 'tissue', 'bolpen']

Dari kedua contoh *tweet* dari *tweet* tentang BlueBird Taxi yang telah melalui praproses teks tersebut, maka didapatkan struktur data setelah praproses teks ditunjukkan pada Tabel 4.3. Tabel 4.3 menunjukkan pembentukan struktur data setelah dilakukan praproses teks dengan menjadikan setiap kata menjadi variabel prediktor dan meletakkannya pada satu baris. Jika terdapat tambahan kata (variabel prediktor) dari *tweet* baru, maka kata tersebut diletakkan pada baris yang sama dan di kolom berikutnya. Namun jika terdapat kata yang sama atau kata yang telah ada pada

struktur data, maka kata tersebut tidak dimasukkan lagi pada struktur data. Sehingga tidak terdapat kata atau variabel prediktor yang sama dalam struktur data. Nilai dari setiap kata tersebut merupakan frekuensi kemunculan kata dalam *tweet* ke-*i* seperti yang terdapat pada Tabel 4.3.

Tabel 4.3 Struktur Data Setelah Praproses Teks

Tweet	**Variabel Prediktor**								
ke-	males	ribet	pake	telp	akun	login	…	tissue	bolpen
1	1	1	1	1	1	1	…	0	0
2	0	0	0	0	0	0	…	1	1
.	.	.	.	.	.	.	…	.	.
.	.	.	.	.	.	.	…	.	.
3111	0	0	0	0	0	0	…	0	0

Data yang telah terbentuk *document term matrix* seperti pada Tabel 4.3 dapat dilakukan perhitungan banyak kata yang selanjutnya akan menjadi banyak variabel dari setiap jenis taksi. Data *tweet* BlueBird Taxi memiliki kata kunci sebanyak 3737 kata dan data *tweet* GoCar memiliki kata kunci sebanyak 5184 kata, sedangkan data *tweet* GrabCar memiliki kata kunci sebanyak 3147 kata. Setelah terbentuk struktur data yang diinginkan, maka dapat dilakukan perhitungan frekuensi kemunculan kata pada setiap taksi. Berikut merupakan 10 kata kunci dengan frekuensi tertinggi untuk tiap penyedia jasa layanan taksi yang ditampilkan pada Tabel 4.4.

Pada Tabel 4.4 dapat disimpulkan bahwa hal yang paling diperhatikan atau disampaikan pendapatnya oleh para pengguna taksi konvensional dan taksi berbasis *online* sama, yaitu tentang ‘driver’. Hal tersebut terbukti karena pada kata kunci BlueBird Taxi dan GoCar, kata kunci ‘driver’ merupakan kata kunci terbanyak pertama, dan pada GrabCar kata kunci ‘driver’ berada pada urutan kelima. Hal ini tentu saja dapat menjadi pertimbangan untuk perusahaan penyedia jasa taksi, baik itu taksi konvensional maupun taksi berbasis *online* untuk lebih memperhatikan drivernya agar lebih dapat memberikan pelayanan yang lebih baik kepada

para penggunanya, karena driver merupakan salah satu hal yang penting yang sangat diperhatikan oleh para pengguna jasa transportasi umum taksi. Dari daftar 10 kata kunci dengan frekuensi kemunculan tertinggi pada setiap penyedia jasa layanan transportasi umum taksi pada Tabel 4.4 menunjukkan bahwa kata-kata tersebut merupakan kata-kata yang mempunyai pengaruh yang signifikan dalam pembangunan model klasifikasi.

Tabel 4.4 Frekuensi Kata Kunci Untuk Masing-Masing Taksi

BlueBird		**GoCar**		**GrabCar**	
Kata Kunci	**Frekuensi**	**Kata Kunci**	**Frekuensi**	**Kata Kunci**	**Frekuensi**
Driver	463	Driver	1392	Mitra	456
App	457	Uber	916	Ngobrol	450
Tumpang	327	Selamat	772	Timdudukdepan	438
Supir	319	Rating	757	Inspirasi	429
Bandara	263	Bahas	756	Driver	393
Tarif	249	Aneh	753	Pake	194
Gojek	248	Simpel	753	Katague	144
Org	210	Gojek	389	Grabbike	142
Nyegat	202	Order	346	Mobil	140
Gada	200	Pake	330	Order	139

4.3 Klasifikasi Menggunakan *Naïve Bayes Classifier* (NBC)

Metode *Naïve Bayes Classifier* merupakan klasifikasi statistik yang dapat memprediksi kelas suatu anggota probabilitas. Kelebihan dalam NBC adalah algoritmanya sederhana tetapi memiliki akurasi yang tinggi. Dalam analisis dengan menggunakan NBC, data akan dibagi menjadi data *training* dan data *testing*. Proses ini akan dilakukan pada masing-masing jenis taksi dengan menggunakan seluruh data *tweet* yang telah dilakukan pengkategorian sentimen sebelumnya. Klasifikasi dengan menggunakan metode *Naïve Bayes Classifier* menghasilkan probabilitas yang digunakan untuk menentukan apakah *tweet* masuk ke dalam kategori sentimen positif atau negatif. Model klasifikasi yang didapatkan dari data *training* untuk menghitung

nilai probabilitas untuk tiap kategori sentimen ditampilkan dalam Tabel 4.5.

Tabel 4.5 Model Klasifikasi NBC

Taksi	Sentimen	Model
BlueBird	Positif	$0.389\left(0.000188^{(f_{b1})} \times 0.000377^{(f_{b2})} \times \ldots \times 0.000188^{(f_{b3734})}\right)$
	Negatif	$0.611\left(0.000312^{(f_{b1})} \times 0.000156^{(f_{b2})} \times \ldots \times 0.000312^{(f_{b3734})}\right)$
GoCar	Positif	$0.572\left(0.000126^{(f_{go1})} \times 0.000252^{(f_{go2})} \times \ldots \times 0.000252^{(f_{go5181})}\right)$
	Negatif	$0.428\left(0.00024^{(f_{go1})} \times 0.00012^{(f_{go2})} \times \ldots \times 0.00012^{(f_{go5181})}\right)$
GrabCar	Positif	$0.583\left(0.000427^{(f_{gr1})} \times 0.00705^{(f_{gr2})} \times \ldots \times 0.000214^{(f_{go3144})}\right)$
	Negatif	$0.417\left(0.000197^{(f_{gr1})} \times 0.004133^{(f_{gr2})} \times \ldots \times 0.000197^{(f_{go3144})}\right)$

Pada Tabel 4.5, menunjukkan f_{b1} merupakan frekuensi kata pertama pada BlueBird Taxi, f_{b2} merupakan frekuensi kata kedua pada BlueBird Taxi, hingga f_{b3734} yaitu banyaknya kata kunci yang didapatkan pada BlueBird Taxi. Kemudian f_{go1} merupakan frekuensi kata pertama pada GoCar, f_{go2} merupakan frekuensi kata kedua pada GoCar, hingga f_{go5181} yaitu banyaknya kata kunci yang didapatkan pada GoCar. Sedangkan, f_{gr1} merupakan frekuensi kata pertama pada GrabCar, f_{gr2} merupakan frekuensi kata kedua pada GrabCar, hingga f_{gr3144} yaitu banyaknya kata kunci yang didapatkan pada GrabCar. Frekuensi setiap kata kunci yang didapatkan untuk masing-masing sentimen tidak sama, tergantung berapa banyak kata kunci tersebut muncul pada sentimen positif atau sentimen negatif. Berdasarkan model yang telah didapatkan, maka untuk menentukan kategori sentimen pada data *tweet* dapat dicari probabilitas terbesar dari hasil yang telah didapatkan dari masing-masing kategori sentimen. Berikut adalah hasil *accuracy* pada data *testing* dengan menggunakan metode *k-fold cross*

validation dengan menggunakan 10-*fold* untuk tiap jenis taksi yang ditunjukkan pada Tabel 4.6.

Tabel 4.6 Nilai *Accuracy* Metode NBC dengan 10-*fold CV*

fold ke-	**BlueBird**	**GoCar**	**GrabCar**
1	0.750	0.647	0.778
2	0.701	0.802	0.896
3	0.791	0.906	0.848
4	0.871	0.880	0.873
5	0.775	0.832	0.732
6	0.711	0.694	0.662
7	0.932	0.617	0.680
8	0.920	0.624	0.899
9	0.897	0.729	0.759
10	0.746	0.685	0.697
Rata-Rata	0.809	0.741	0.782

Berdasarkan Tabel 4.6 didapatkan rata-rata nilai *accuracy* pada masing-masing penyedia jasa transportasi umum taksi dengan menggunakan 10-*fold cross validation*. Pada BlueBird Taxi, didapatkan rata-rata nilai *accuracy* sebesar 80.9% dengan nilai *accuracy* tertinggi pada *fold* ke 7. Kemudian, untuk GoCar didapatkan rata-rata nilai *accuracy* sebesar 74.1% dengan nilai *accuracy* tertinggi pada *fold* ke 3. Sedangkan, pada GrabCar didapatkan rata-rata nilai *accuracy* sebesar 78.2% dengan nilai *accuracy* tertinggi pada *fold* ke 8. Selanjutnya akan dilakukan pengukuran ketepatan klasifikasi dengan membentuk *confusion matrix* berdasarkan *fold* dengan nilai *accuracy* tertinggi dari masing-masing jenis taksi. Berikut merupakan hasil *confusion matrix* untuk tiap jenis taksi yang disajikan pada Tabel 4.7.

Tabel 4.7 *Confusion Matrix* dengan Metode NBC

Data	**Jenis Taksi**	**Jumlah *Tweet***	***True Positive***	***False Positive***	***True Negative***	***False Negative***
Training	BlueBird	2800	939	33	1677	151
	GoCar	3900	2012	147	1525	216
	GrabCar	2058	1165	83	766	44

Tabel 4.7 *Confusion Matrix* dengan Metode NBC (Lanjutan)

Data	Jenis Taksi	Jumlah *Tweet*	*True Positive*	*False Positive*	*True Negative*	*False Negative*
Testing	BlueBird	311	120	20	170	1
	GoCar	434	247	40	146	1
	GrabCar	228	129	18	76	5

Setelah terbentuk *confusion matrix* pada Tabel 4.7, langkah selanjutnya yaitu melakukan perhitungan ketepatan klasifikasi. Hasil perhitungan ketepatan klasifikasi untuk setiap jenis taksi dirangkum menjadi satu tabel berdasarkan data *training* dan data *testing*. Berikut merupakan hasil pengukuran ketepatan klasifikasi dari setiap penyedia jasa transportasi umum taksi dengan menggunakan pengukuran nilai *accuracy*, *sensitivity*, *specificity*, *G-Mean*, dan AUC yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 Ketepatan Klasifikasi Metode NBC

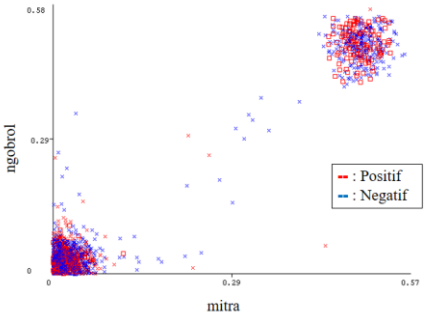
Data	Taksi	*Accuracy*	*Sensitivity*	*Specificity*	*G-Mean*	AUC
Training	BlueBird	0.934	0.861	0.981	0.919	0.921
	GoCar	0.907	0.903	0.912	0.908	0.908
	GrabCar	0.938	0.964	0.902	0.932	0.933
Testing	BlueBird	0.932	0.992	0.895	0.942	0.943
	GoCar	0.906	0.996	0.785	0.884	0.890
	GrabCar	0.899	0.963	0.809	0.882	0.886

Berdasarkan Tabel 4.8 dapat diketahui bahwa hasil klasifikasi pada data *training* untuk ketiga jenis taksi didapatkan nilai *accuracy* yang cukup tinggi yaitu diatas 90%. Kemudian pada nilai AUC, didapatkan hasil nilai AUC untuk ketiga jenis taksi berada diatas 0.9. Jika mengacu pada Tabel 2.3, maka dapat diartikan ketepatan klasifikasi pada data *training* untuk ketiga jenis taksi adalah baik sekali. Karena data yang digunakan cenderung *balanced*, maka selanjutnya perbandingan dapat dilakukan dengan menggunakan nilai *accuracy*. Pada data *testing*, didapatkan nilai *accuracy* untuk penyedia jasa transportasi umum BlueBird Taxi, yaitu sebesar 93.2%. Sedangkan untuk penyedia jasa transportasi

umum GoCar dan GrabCar, didapatkan nilai *accuracy* yaitu sebesar 90.6% untuk GoCar dan 89.9% untuk GrabCar.

4.4 Klasifikasi Menggunakan *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan sistem pembelajaran yang menggunakan hipotesis fungsi linear dalam ruang berdimensi tinggi dan dilatih dengan algoritma berdasarkan teori optimasi dengan menerapkan *learning bias* yang berasal dari teori statistik. Tujuan utama dari metode ini adalah membangun pemisah optimum yang disebut OSH (*Optimal Separating Hyperplane*) sehingga dapat digunakan untuk klasifikasi. Pembahasan pada SVM ini akan sama dengan pembahasan pada NBC. Perbedaannya adalah ditambahkan parameter untuk SVM dengan Kernel Linear yaitu parameter *C*, dan parameter untuk SVM dengan Kernel RBF (*Radial Basis Function*) yaitu parameter *C* dan parameter *gamma* (γ). Data yang akan dianalisis menggunakan metode SVM merupakan data yang telah dilakukan pembobotan dengan menggunakan *term frequency-inverse document frequency* (TF-IDF) terlebih dahulu pada setiap kata. Pembobotan tersebut menyebabkan adanya sebaran data. Berikut ini merupakan salah satu contoh sebaran data pada variabel kata 'driver' dan 'app' yang merupakan kata kunci yang didapatkan dari BlueBird Taxi pada kategori sentimen positif dan sentimen negative yang disajikan pada Gambar 4.6.



Gambar 4.6 *Scatterplot* Variabel Ngobrol dan Variabel Mitra

Pada Gambar 4.6 menunjukkan bahwa *scatterplot* antara variabel ‘mitra’ dan variabel ‘ngobrol’ terdapat kemungkinan data dapat terpisah secara linear, sehingga pada pembahasan klasifikasi menggunakan metode SVM akan dilakukan dengan menggunakan Kernel Linear dan Kernel RBF. Pembahasan pertama akan dilakukan dengan menggunakan Kernel Linear dan kemudian dilanjutkan dengan menggunakan Kernel RBF.

4.4.1 SVM Menggunakan Kernel Linear

Pada mulanya data *tweet* dibagi menjadi dua klasifikasi sentimen sama seperti pembahasan pada NBC. Parameter *C* yang digunakan pada Kernel Linear akan dicoba dari 10^{-2} hingga 10^{4} (Huang, Lee, Lin, & Huang, 2007). Dalam analisis dengan menggunakan SVM Kernel Linear, data akan dibagi menjadi data *training* dan data *testing*. Berikut merupakan hasil ketepatan klasifikasi dari metode SVM dengan menggunakan Kernel Linear untuk menentukan parameter *C* yang memberikan hasil ketepatan klasifikasi optimum untuk masing-masing penyedia jasa transportasi umum taksi yang disajikan pada Tabel 4.9.

Tabel 4.9 Penentuan Parameter SVM Kernel Linear

C	**Ketepatan Klasifikasi**		
	BlueBird Taxi	**GoCar**	**GrabCar**
0.01	0.606	0.577	0.590
0.1	0.738	0.698	0.706
1	0.851	0.808	0.806
10	0.839	0.791	0.796
100	0.832	0.767	0.792
1000	0.824	0.766	0.792
10000	0.824	0.766	0.792

Dari Tabel 4.9 didapatkan ketepatan klasifikasi optimum yang didapat menggunakan SVM Kernel Linear pada masing-masing jenis taksi adalah dengan menggunakan parameter *C* sebesar 1. Selanjutnya adalah dilakukan pengukuran ketepatan klasifikasi dengan membentuk *confusion matrix* dengan

menggunakan parameter *C* sebesar 1. Selanjutnya, dilakukan pemecahan data *tweet* menjadi data *training* dan data *testing* berdasarkan metode *k-fold cross validation* dengan menggunakan 10-*fold*. Berikut adalah nilai *accuracy* pada data *testing* dengan menggunakan 10-*fold cross validation* pada setiap penyedia jasa transportasi umum taksi yang disajikan pada Tabel 4.10.

Dari Tabel 4.10 menunjukkan rata-rata nilai *accuracy* pada BlueBird Taxi sebesar 81.3% dengan nilai *accuracy* tertinggi pada *fold* ke 7. Kemudian pada GoCar, didapatkan rata-rata nilai *accuracy* dengan menggunakan 10-*fold cross validation* adalah sebesar 86.6% dengan nilai *accuracy* tertinggi pada *fold* ke 3. Sedangkan untuk GrabCar, didapatkan rata-rata nilai *accuracy* adalah sebesar 75.8% dengan nilai *accuracy* tertinggi didapatkan pada *fold* ke 8.

Tabel 4.10 Nilai *Accuracy* Metode SVM Kernel Linear dengan 10-*fold CV*

fold ke-	**BlueBird**	**GoCar**	**GrabCar**
1	0.731	0.733	0.791
2	0.768	0.802	0.887
3	0.817	0.866	0.830
4	0.900	0.843	0.855
5	0.711	0.841	0.724
6	0.727	0.765	0.671
7	0.932	0.663	0.583
8	0.920	0.762	0.904
9	0.875	0.769	0.684
10	0.752	0.731	0.649
Rata-Rata	0.813	0.778	0.758

Langkah selanjutnya adalah melakukan pengukuran ketepatan klasifikasi dengan membentuk *confusion matrix* berdasarkan *fold* dengan nilai *accuracy* tertinggi yang terlah didapatkan pada Tabel 4.10. Berikut adalah hasil *confusion matrix* pada data *training* dan *testing* dari setiap jenis taksi yang disajikan pada Tabel 4.11.

Tabel 4.11 *Confusion Matrix* dengan Metode SVM Kernel Linear

Data	Jenis Taksi	Jumlah *Tweet*	*True Positive*	*False Positive*	*True Negative*	*False Negative*
Training	BlueBird	2800	1010	28	1682	80
	GoCar	3900	2129	88	1584	99
	GrabCar	2058	1177	18	831	32
Testing	BlueBird	311	118	18	172	3
	GoCar	434	247	57	129	1
	GrabCar	228	129	17	77	5

Tabel 4.11 menunjukkan hasil *confusion matrix* yang didapatkan dengan menggunakan SVM Kernel Linear dengan parameter *C* sebesar 1. Setelah didapatkan *confusion matrix* tersebut, maka langkah selanjutnya adalah melakukan perhitungan ketepatan klasifikasi untuk setiap penyedia jasa transportasi umum taksi yang akan dirangkum menjadi satu tabel berdasarkan data *training* dan data *testing*. Berikut ini merupakan hasil pengukuran ketepatan klasifikasi untuk setiap jenis taksi yang disajikan pada Tabel 4.12.

Tabel 4.12 Ketepatan Klasifikasi Metode SVM Kernel Linear

Data	Taksi	*Accuracy*	*Sensitivity*	*Specificity*	*G-Mean*	*AUC*
Training	BlueBird	0.961	0.927	0.984	0.955	0.955
	GoCar	0.952	0.956	0.947	0.951	0.951
	GrabCar	0.976	0.974	0.979	0.976	0.976
Testing	BlueBird	0.932	0.975	0.905	0.940	0.940
	GoCar	0.866	0.996	0.694	0.831	0.845
	GrabCar	0.904	0.963	0.819	0.888	0.891

Dari Tabel 4.12 menunjukkan hasil ketepatan klasifikasi untuk ketiga jenis taksi yang mendapatkan nilai *accuracy* cukup tinggi untuk data *training* yaitu berada diatas 90%. Kemudian untuk nilai AUC, pada ketiga jenis taksi didapatkan nilai AUC yang berada dikisaran angka 0.9 hingga 1. Jika mengacu pada Tabel 2.3, maka dapat disimpulkan bahwa hasil ketepatan klasifikasi yang didapatkan dapat dikatakan baik sekali jika ditinjau dari nilai AUC. Karena data yang digunakan cenderung

balanced, maka selanjutnya perbandingan dapat dilakukan dengan menggunakan nilai *accuracy*. Kemudian pada data *testing*, nilai *accuracy* yang didapatkan oleh BlueBird Taxi merupakan nilai *accuracy* yang tertinggi jika dibandingkan dengan taksi yang lainnya, yaitu sebesar 93.2%. Sedangkan untuk jenis taksi GoCar dan GrabCar mendapatkan nilai *accuracy* sebesar 86.6% dan 90.4%.

4.4.2 SVM Menggunakan Kernel *Radial Basis Function* (RBF)

Pada SVM dengan menggunakan Kernel RBF, parameter yang digunakan ada dua, yaitu *C* dan γ. Parameter *C* yang digunakan pada Kernel Linear akan dicoba dari 10^{-2} hingga 10^{4} (Huang, Lee, Lin, & Huang, 2007). Sedangkan untuk *gamma* (γ) akan didapatkan melalui hasil percobaan untuk mendapatkan hasil yang paling baik. Dalam analisis dengan menggunakan SVM Kernel RBF, data akan dibagi menjadi data *training* dan data *testing*. Berikut merupakan hasil ketepatan klasifikasi dari metode SVM dengan Kernel RBF yang disajikan pada Tabel 4.13.

Tabel 4.13 Penentuan Parameter SVM Kernel RBF

Jenis Taksi	*C*	*Gamma* (γ)						
		1000	100	10	1	0.1	0.01	0.001
BlueBird	0.01	0.606	0.606	0.606	0.606	0.606	0.606	0.606
	0.1	0.699	0.699	0.702	0.719	0.640	0.606	0.606
	1	0.777	0.777	0.779	0.845	0.769	0.646	0.606
	10	0.777	0.777	0.781	0.859	0.857	0.779	0.646
	100	0.777	0.777	0.781	0.859	0.841	0.855	0.781
	1000	0.777	0.777	0.781	0.859	0.841	0.833	0.855
	10000	0.777	0.777	0.781	0.859	0.841	0.833	0.834
GoCar	0.01	0.571	0.571	0.571	0.570	0.570	0.570	0.570
	0.1	0.669	0.669	0.669	0.675	0.617	0.570	0.570
	1	0.701	0.701	0.703	0.796	0.741	0.608	0.570
	10	0.701	0.701	0.704	0.808	0.802	0.742	0.608
	100	0.701	0.701	0.704	0.808	0.792	0.798	0.742
	1000	0.701	0.701	0.704	0.808	0.789	0.787	0.798
	10000	0.701	0.701	0.704	0.808	0.789	0.774	0.784

Tabel 4.13 Penentuan Parameter SVM Kernel RBF (Lanjutan)

Jenis Taksi	*C*	*Gamma* (*γ*)						
		1000	100	10	1	0.1	0.01	0.001
GrabCar	0.01	0.590	0.590	0.590	0.590	0.590	0.590	0.590
	0.1	0.684	0.684	0.684	0.694	0.617	0.590	0.590
	1	0.717	0.717	0.724	0.814	0.737	0.622	0.590
	10	0.717	0.717	0.726	0.817	0.814	0.750	0.627
	100	0.717	0.717	0.726	0.817	0.807	0.812	0.750
	1000	0.717	0.717	0.726	0.817	0.808	0.796	0.813
	10000	0.717	0.717	0.726	0.817	0.808	0.795	0.794

Dari Tabel 4.13 didapatkan ketepatan klasifikasi terbaik yang didapat menggunakan SVM Kernel RBF pada masing-masing penyedia jasa transportasi umum taksi dengan menggunakan parameter *C* sebesar 10 dan parameter *gamma* (*γ*) sebesar 1. Setelah didapatkan parameter yang sesuai digunakan dengan untuk metode SVM Kernel RBF, akan dilakukan pemecahan data *tweet* menjadi data *training* dan data *testing* dengan menggunakan metode *k-fold cross validation*. Pada proses ini akan dilakukan dengan menggunakan 10-*fold*. Tabel 4.14 menunjukkan hasil *accuracy* pada data *testing* dengan menggunakan 10-*fold cross validation* pada setiap penyedia jasa transportasi umum taksi.

Tabel 4.14 Nilai *Accuracy* Metode SVM Kernel RBF dengan 10-*fold CV*

fold ke-	**BlueBird**	**GoCar**	**GrabCar**
1	0.718	0.724	0.800
2	0.717	0.809	0.896
3	0.814	0.876	0.852
4	0.875	0.882	0.886
5	0.730	0.848	0.715
6	0.788	0.698	0.675
7	0.961	0.649	0.649
8	0.920	0.776	0.895
9	0.894	0.782	0.789
10	0.746	0.713	0.693
Rata-Rata	0.816	0.776	0.785

Pada Tabel 4.14 menunjukkan hasil nilai *accuracy* pada masing-masing jenis taksi dengan menggunakan 10-*fold cross validation*. Nilai *accuracy* rata-rata yang didapatkan untuk BlueBird Taxi adalah sebesar 96.1% dengan nilai *accuracy* tertinggi pada *fold* ke 7. Kemudian untuk GoCar, didapatkan rata-rata nilai *accuracy* sebesar 88.2% dengan nilai *accuracy* tertinggi pada *fold* ke 4. Sedangkan, nilai rata-rata *accuracy* yang didapatkan pada GrabCar adalah sebesar 78.5% dengan nilai *accuracy* yang tertinggi diperoleh pada *fold* ke 2. Setelah didapatkan *fold* dengan nilai *accuracy* tertinggi pada masing-masing jenis taksi, maka selanjutnya akan dilakukan pengukuran ketepatan klasifikasi dengan membentuk *confusion matrix*. Berikut adalah hasil *confusion matrix* pada data *training* dan data *testing* dari setiap taksi yang ditunjukkan pada Tabel 4.15.

Tabel 4.15 *Confusion Matrix* dengan Metode SVM Kernel RBF

Data	**Jenis Taksi**	**Jumlah *Tweet***	***True Positive***	***False Positive***	***True Negative***	***False Negative***
Training	BlueBird	2800	1080	4	1706	10
	GoCar	3900	2225	7	1665	3
	GrabCar	2056	1207	4	844	1
Testing	BlueBird	311	119	10	180	2
	GoCar	434	241	44	142	7
	GrabCar	230	133	22	73	2

Pada Tabel 4.15 didapatkan hasil *confusion matrix* dengan menggunakan SVM Kernel RBF dengan parameter *C* sebesar 10 dan parameter *gamma* (γ) sebesar 1. Kemudian akan dilakukan perhitungan ketepatan klasifikasi untuk setiap penyedia jasa transportasi umum taksi yang akan dirangkum dalam satu tabel berdasarkan data *training* dan data *testing*. Berikut adalah hasil pengukuran ketepatan klasifikasi yang disajikan pada Tabel 4.16.

Tabel 4.16 Ketepatan Klasifikasi Metode SVM Kernel RBF

Data	Taksi	*Accuracy*	*Sensitivity*	*Specificity*	*G-Mean*	*AUC*
Training	BlueBird	0.995	0.991	0.998	0.994	0.994
	GoCar	0.997	0.999	0.996	0.997	0.997
	GrabCar	0.998	0.999	0.995	0.997	0.997
Testing	BlueBird	0.961	0.983	0.947	0.965	0.965
	GoCar	0.882	0.972	0.763	0.861	0.868
	GrabCar	0.896	0.985	0.768	0.870	0.877

Dari Tabel 4.16 didapatkan hasil ketepatan klasifikasi untuk ketiga jenis taksi menunjukkan nilai *accuracy* untuk data *training* cukup tinggi yaitu berada diatas 90%. Kemudian untuk nilai AUC, didapatkan hasil nilai AUC untuk ketiga jenis taksi berada diatas 0.9. Jika mengacu pada Tabel 2.3, maka dapat diartikan ketepatan klasifikasi pada data *training* untuk ketiga jenis taksi adalah baik sekali. Karena data yang digunakan cenderung *balanced*, maka selanjutnya perbandingan dapat dilakukan dengan menggunakan nilai *accuracy*. Selanjutnya untuk data *testing*, didapatkan nilai *accuracy* untuk penyedia jasa transportasi umum BlueBird Taxi merupakan nilai *accuracy* tertinggi jika dibandingkan dengan jenis taksi yang lain, yaitu sebesar 96.1%. Sedangkan untuk penyedia jasa transportasi umum taksi GoCar dan GrabCar, didapatkan nilai *accuracy* sebesar 88.2% dan 89.6%.

4.4.3 Pengukuran Performa SVM

Dari hasil analisis SVM dengan menggunakan Kernel Linear dan Kernel RBF didapatkan perbedaan parameter dan hasil ketepatan klasifikasi pada masing-masing Kernel. Sehingga akan dilakukan perbandingan ketepatan klasifikasi antara metode SVM dengan menggunakan Kernel Linear dan Kernel RBF dengan menggunakan rata-rata nilai *accuracy* pada data *testing* dengan 10-*fold cross validation* yang akan disajikan pada Tabel 4.17.

Tabel 4.17 Perbandingan Ketepatan Klasifikasi Antara SVM Kernel Linear dan Kernel RBF

Jenis Taksi	Kernel Linear		Kernel RBF	
	Parameter	***Accuracy***	**Parameter**	***Accuracy***
BlueBird	$C = 1$	0.813	$C = 10$ $\gamma = 1$	0.816
GoCar		0.778		0.776
GrabCar		0.758		0.785

Berdasarkan Tabel 4.17 dapat disimpulkan bahwa dari perbandingan metode SVM dengan menggunakan Kernel Linear dan Kernel RBF, didapatkan metode SVM yang memiliki ketepatan klasifikasi terbaik atau yang tinggi adalah dengan menggunakan metode SVM Kernel RBF dengan parameter yang digunakan yaitu parameter *C* sebesar 10 dan parameter *gamma* (γ) sebesar 1 pada jenis taksi BlueBird Taxi dan GrabCar. Namun, pada jenis taksi GoCar ketepatan klasifikasi yang tinggi didapatkan dengan menggunakan metode SVM Kernel Linear dengan menggunakan parameter C sebesar 1. Akan tetapi, nilai yang didapatkan dengan menggunakan metode SVM Kernel Linear dan SVM Kernel RBF tidak berbeda jauh, sehingga metode SVM yang terpilih adalah metode SVM Kernel RBF dengan parameter yang digunakan yaitu parameter *C* sebesar 10 dan parameter *gamma* (γ) sebesar 1.

Hasil ketepatan klasifikasi terbaik dari ketiga penyedia jasa transportasi umum taksi pada metode SVM Kernel RBF menggunakan nilai parameter yang sama yaitu menggunakan parameter *C* sebesar 10 dan parameter *gamma* (γ) sebesar 1. Kemudian parameter *gamma* tersebut disubstitusikan pada persamaan Kernel RBF yang terdapat pada Tabel 2.1, sehingga fungsi Kernel RBF pada ketiga taksi yang terbentuk sama yaitu sebagai berikut.

$$K\left(\mathbf{x}_i, \mathbf{x}\right) = \exp\left(-1\left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right)$$

Fungsi *hyperplane* didapatkan dengan mensubstitusikan fungsi Kernel pada persamaan (2.25). Sehingga didapatkan fungsi

hyperplane dari data *training* untuk masing-masing penyedia jasa transportasi umum taksi yang disajikan pada Tabel 4.18.

Tabel 4.18 Persamaan *Hyperplane* pada Setiap Taksi

Jenis Taksi	Persamaan *Hyperplane*
BlueBird	$f_b(x) = \sum_{i=1}^{2766} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + 0.2219$
GoCar	$f_{go}(x) = \sum_{i=1}^{3883} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - 0.1029$
GrabCar	$f_{gr}(x) = \sum_{i=1}^{2044} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - 0.1097$

Pada persamaan *hyperplane* pada Tabel 4.18, α_i merupakan vektor koefisien atau *lagrange multiplier* dari *support vector* yang berukuran (2766,1) untuk BlueBird Taxi (nilainya terlampir pada Lampiran 18), dan berukuran (3883,1) untuk GoCar (nilainya terlampir pada Lampiran 19), serta berukuran (2044,1) untuk GrabCar (nilainya terlampir pada Lampiran 20). Kemudian y_i merupakan label kelas yang memiliki dua nilai yaitu +1 dan -1, serta *x* merupakan nilai *input* yang akan diklasifikasikan. Nilai 0.2119, -0.1029, dan -0.1097 merupakan nilai bias untuk masing-masing persamaan *hyperplane* jenis taksi. $f_b(x)$ merupakan persamaan *hyperplane* untuk BlueBird Taxi, $f_{go}(x)$ merupakan persamaan *hyperplane* untuk GoCar, dan $f_{gr}(x)$ merupakan persamaan *hyperplane* untuk GrabCar.

4.5 Perbandingan Hasil Klasifikasi Antara Metode NBC dan SVM

Setelah mengetahui hasil dari masing-masing ketepatan klasifikasi pada kedua metode, maka langkah selanjutnya adalah membandingkan. Berikut merupakan perbandingan antara kedua metode berdasarkan rata-rata nilai *accuracy* dari data *testing* dengan 10-*fold cross validation* yang ditunjukkan pada Tabel 4.19.

Berdasarkan perbandingan metode NBC dan SVM untuk masing-masing jenis taksi pada Tabel 4.19, jika dilihat dari rata-rata nilai *accuracy* pada data *testing*, maka dapat disimpulkan bahwa metode SVM dengan Kernel RBF lebih baik jika dibandingkan dengan metode NBC. Sehingga metode yang cocok digunakan untuk pengklasifikasian pada penelitian ini adalah metode SVM dengan Kernel RBF dengan menggunakan parameter *C* sebesar 10 dan *gamma* (γ) sebesar 1. Ketepatan klasifikasi yang didapatkan untuk masing-masing jenis taksi dengan metode SVM Kernel RBF didapatkan rata-rata nilai *accuracy* pada data *testing* sebesar 81.6% untuk BlueBird Taxi, kemudian untuk GoCar sebesar 77.6%, sedangkan untuk GrabCar sebesar 78.5%.

Tabel 4.19 Perbandingan Metode NBC dan SVM

Metode	Taksi	*Accuracy*
NBC	BlueBird	0.809
	GoCar	0.741
	GrabCar	0.782
SVM Kernel RBF	BlueBird	0.816
	GoCar	0.776
	GrabCar	0.785

4.6 Visualisasi *Word Cloud*

Visualisasi data teks dengan menggunakan *word cloud* bertujuan untuk mengetahui kata-kata yang paling sering muncul pada data. Pada penelitian ini, visualisasi *word cloud* digunakan untuk visualisasi *tweet* berdasarkan kategori sentimennya, sehingga dapat diketahui kata-kata yang sering muncul pada setiap sentimen tentang masing-masing jenis taksi. Ukuran *font* pada *word cloud* menunjukkan frekuensi kemunculan kata. Jadi, semakin besar ukuran *font* berarti semakin besar frekuensi kemunculan kata tersebut.

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

pengguna menilai masing-masing pelayanan dari penyedia jasa taksi berdasarkan kategori sentimen. Hasil visualisasi dengan *word cloud* untuk masing-masing kategori sentimen pada BlueBird Taxi ditunjukkan pada Gambar 4.7.

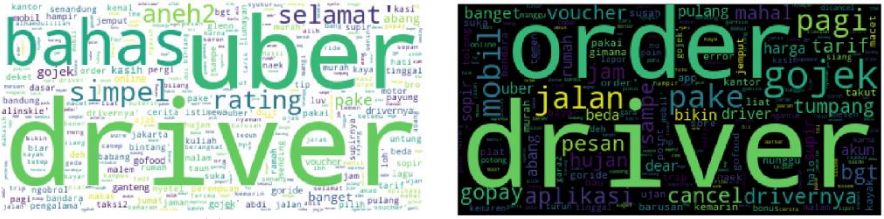
(a) (b)

Gambar 4.7 *Word Cloud* BlueBird Taxi Sentimen Positif (a) dan Sentimen Negatif (b)

Pada Gambar 4.7 menunjukkan kata kunci yang mengarah pada BlueBird Taxi, terlihat bahwa kata kunci terbanyak yang digunakan untuk respon pengguna pada sentimen positif terbesar yaitu kata ‘tarif’. Hal tersebut disebabkan karena para pengguna sering membandingkan tarif antara taksi konvensional dengan tarif taksi berbasis *online* yang kenyataannya tarif dari BlueBird Taxi memang lebih murah dibandingkan taksi berbasis *online* menurut pengguna. Sedangkan, pada sentimen negatif kata kunci yang terbesar adalah ‘app’, dikarenakan memang aplikasi dari BlueBird Taxi masih kalah jika dibandingkan dengan aplikasi dari taksi berbasis *online*, sehingga banyak dikeluhkan oleh para pengguna. Kemudian, didapatkan pula kata kunci ‘driver’ dan ‘supir’ yang mendapat banyak perhatian dari para pengguna, sehingga banyak pengguna yang berpendapat positif atau negatif terhadap driver atau supir dari BlueBird Taxi. Kata-kata lain pada sentimen positif dan sentimen negatif mempunyai frekuensi yang jauh kecil yang ditandai dengan ukuran *font* yang berbeda. Selanjutnya, dilakukan visualisasi *word cloud* untuk masing-masing kategori sentimen pada GoCar yang disajikan pada Gambar 4.8.

Berdasarkan Gambar 4.8 didapatkan bahwa kata kunci terbesar untuk GoCar dari sentimen positif dan negatif sama, yaitu

kata ‘driver’. Hal tersebut memang sangat dirasakan oleh para pengguna GoCar, karena terkadang pengguna GoCar mendapatkan driver yang memang baik dan memberikan pelayanan yang memuaskan, sehingga pengguna memberikan pendapatnya tentang pelayanan yang diperoleh dari driver. Terkadang pula pengguna mendapatkan driver yang foto dan plat nomor kendaraannya tidak sesuai dengan aplikasi, sehingga hal tersebut juga sering dikeluhkan oleh pengguna. Terlebih lagi, banyak juga beredar pemberitaan tentang kasus-kasus yang dilakukan oleh driver GoCar.



(a) (b)

Gambar 4.8 *Word Cloud* GoCar Sentimen Positif (a) dan Sentimen Negatif (b)

Selain itu, pada kategori sentimen positif didapatkan pula bahwa pengguna jasa layanan taksi GoCar juga sering menggunakan kata ‘uber’ untuk menyampaikan pendapatnya tentang pelayanan dari GoCar, yang berarti banyak pengguna yang membandingkan pelayanan dari penyedia jasa taksi GoCar dengan penyedia jasa taksi lain yaitu Uber. Selanjutnya, dilakukan visualisasi *word cloud* pada masing-masing kategori sentimen dari penyedia jasa transportasi umum GrabCar yang disajikan pada Gambar 4.9.

Dari Gambar 4.9 didapatkan kata kunci dari sentimen positif untuk GrabCar adalah kata ‘ngobrol’. Hal tersebut karena banyak pengguna yang suka dengan driver GrabCar yang suka mengajak berinteraksi selama diperjalanan, sehingga hal tersebut menjadi nilai tambah untuk citra dari GrabCar. Namun pada sentimen negatif, didapatkan kata kunci yang paling banyak digunakan adalah ‘driver’. Permasalahan driver ini memang banyak

dikeluhkan oleh pengguna GrabCar, lantaran terkadang pengguna GrabCar sulit mendapatkan driver jika sedang melakukan pemesanan, atau terkadang driver tidak datang untuk memenuhi pesanan dan tiba-tiba membatalkan pesanan yang dilakukan pelanggan. Sehingga hal tersebut menjadi sebuah permasalahan yang seharusnya diperhatikan oleh pihak GrabCar. Selain itu, sama seperti halnya GoCar, pengguna jasa transportasi umum taksi juga sering membandingkan GrabCar dengan jasa transportasi umum taksi berbasis *online* yang lain yaitu Uber. Hal tersebut ditunjukkan dari adanya kata 'uber' yang cukup dominan pada *word cloud*. Namun berbeda dengan GoCar, sebagian besar pengguna jasa transportasi umum taksi GrabCar lebih banyak mengunggah *tweet* dengan berisi sentimen negatif pada saat membandingkan antara penyedia jasa transportasi umum taksi berbasis *online* Uber dengan GrabCar.

(a) (b)

Gambar 4.9 *Word Cloud* GrabCar Sentimen Positif (a) dan Sentimen Negatif (b)

4.7 *Social Network Analysis* (SNA)

Setelah dilakukan analisis klasifikasi dengan menggunakan metode *Naïve Bayes Classifier* dan *Support Vector Machine*, maka selanjutnya dilakukan anailisis dengan menggunakan *Social Network Analysis* (SNA) untuk mengetahui pengguna Twitter yang berpengaruh berdasarkan topik tentang taksi konvensional dan taksi berbasis *online* dengan menganalisa hasil representasi *graph* yang terbentuk. *Graph type* dari *network* yang digunakan adalah *directed*, yaitu jenis *graph* yang mempertimbangkan arah interaksi.

Berikut adalah hasil analisis *Social Network Analysis* untuk masing-masing jenis taksi.

4.7.1 *Social Network Analysis* BlueBird Taxi

Berikut ini adalah hasil observasi SNA pada *keyword* BlueBird Taxi dalam media sosial Twitter. Sebelumnya, dilakukan analisis dari data perbandingan jaringan dari BlueBird Taxi yang ditunjukkan pada Tabel 4.20.

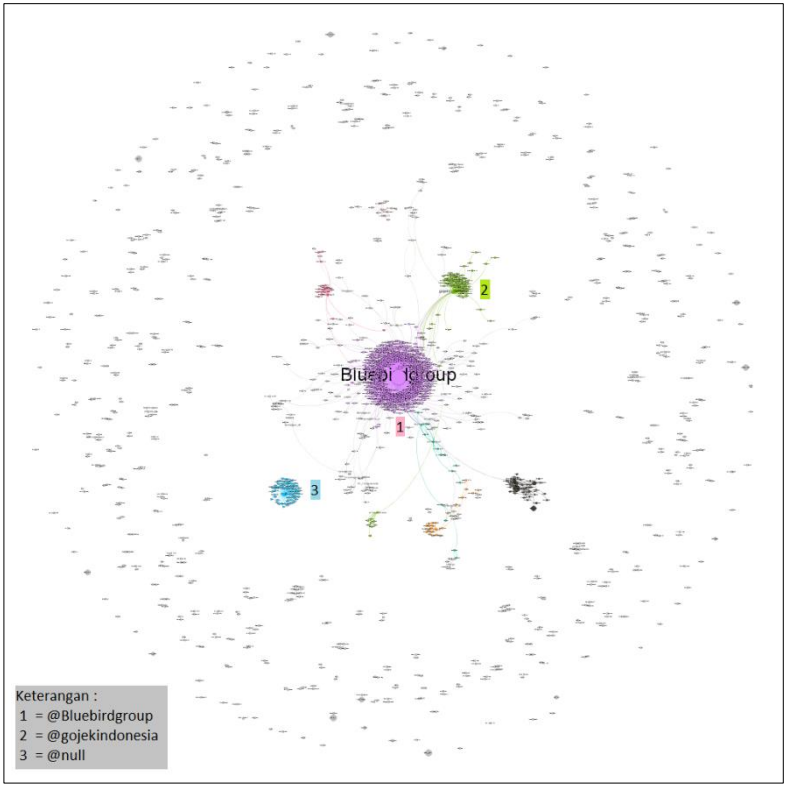
Tabel 4.20 Data Perbandingan Jaringan BlueBird Taxi

Degree Range	*Nodes*	*Edges*
0	4623	1651
1	1721	1651
2	417	523
3	150	213

Dari Tabel 4.20 didapatkan jumlah *node* dan *edge* dari jaringan BlueBird Taxi. Dapat diketahui bahwa jika tanpa menggunakan *filter degree range*, jumlah *nodes* dan *edges* yang didapatkan adalah sebanyak 4623 *nodes* dan 1651 *edges*. Sedangkan, jika menggunakan *filter degree range*, jumlah *nodes* yang didapatkan menjadi 1721 dan jumlah *edges* tetap. Hal tersebut karena jika menggunakan *filter degree range* minimal satu, maka *nodes* yang tidak memiliki interaksi *edges* minimal satu akan dihilangkan, sehingga dapat mengurangi *noise* pada *graph*. Berikut adalah hasil visualisasi *network* yang terbentuk untuk jaringan BlueBird Taxi dengan menggunakan *filter degree range* satu yang ditunjukkan pada Gambar 4.10.

Gambar 4.10 merupakan visualisasi *network* untuk BlueBird Taxi dengan menggunakan tipe jaringan *Frunchterman Reingold* dan *ForceAtlas 2*. Tipe tersebut merupakan tipe yang umum digunakan dalam pembuatan *graph* untuk SNA. Dari Gambar 4.10 didapatkan nilai *density* sebesar 0.001. *Density* adalah kepadatan *graph* suatu *network* yang menunjukkan jumlah hubungan yang hadir dalam suatu kelompok. Ketika menghitung *density*, SNA melihat bagaimana erat hubungan seseorang yang satu dengan

yang lain. Nilai *density* sebesar 0.001 menunjukkan bahwa *network* BlueBird Taxi memiliki kepadatan yang kecil atau renggang. *Density* yang kecil atau renggang ini terjadi karena rendahnya interaksi antar *account*, baik berupa *mention*, *quote retweet*, atau *reply* yang dilakukan antar *node* dalam jaringan. Berikut adalah hasil analisis *centrality* dari Gambar 4.10 yang ditunjukkan pada Tabel 4.21.



Gambar 4.10 Visualisasi *Network* BlueBird Taxi

Analisis *centrality* pada Tabel 4.21 bertujuan untuk menemukan *account* yang paling berperan dalam sebuah *network*. *Metric* yang digunakan dalam penentuan *centrality* ini adalah

degree centrality, *closeness centrality*, *betweenness centrality*, dan *eigenvector centrality*. Analisis *degree centrality* menentukan *account* yang paling berperan berdasarkan banyaknya *edge* atau hubungan yang terjadi antara sebuah *node* dengan *node* yang lainnya. *Account* @Bluebirdgroup memiliki nilai *degree centrality* tertinggi yaitu 635 (pada Gambar 4.10 ditunjukkan dengan angka 1). Hal ini dapat diartikan bahwa @Bluebirdgroup memiliki hubungan atau interaksi yang berupa *mention*, *quote retweet*, atau *reply* antar *node* lain sebanyak 635 kali. Sedangkan, yang kedua adalah *account* @gojekindonesia dengan nilai *degree centrality* sebesar 82 (pada Gambar 4.10 ditunjukkan dengan angka 2), dan yang ketiga adalah *account* @null dengan nilai *degree centrality* sebesar 67 (pada Gambar 4.10 ditunjukkan dengan angka 3).

Tabel 4.21 Analisis *Centrality* dari Jaringan BlueBird Taxi

Centrality	***Username***	***Score***
Degree Centrality	@Bluebirdgroup	635
	@gojekindonesia	82
	@null	67
Closeness Centrality	@PT_TransJakarta	0.888889
	@ancoltmnimpian	0.857143
	@ekyology	0.833333
Betweenness Centrality	@gojekindonesia	990.0
	@WayToLombok	189.0
	@Lostpacker	146.0
Eigenvector Centrality	@Bluebirdgroup	1.0
	@gojekindonesia	0.116251
	@null	0.102723

Analisis *closeness centrality* digunakan untuk melihat *node-node* yang dapat menjangkau *node* lainnya dengan jalur yang lebih pendek. Semakin mendekati 1 atau sama dengan 1, maka semakin dekat *node* tersebut dengan *node* lain. Agar lebih mudah dipahami dan mengurangi bias, maka semua *node* yang memiliki nilai *closeness centrality* sebesar 1 tidak diperhatikan. Terdapat tiga *account* dengan nilai *closeness centrality* tertinggi, yaitu

@PT_TransJakarta, @ancoltmnimpian, dan @ekyology. Nilai *closeness centrality* untuk peringkat pertama yaitu 0.888889 dan peringkat kedua sebesar 0.857143. selisih nilai antara peringkat pertama dan kedua yaitu sebesar 0.031746. Hal ini mengakibatkan *node-node* dengan nilai *closeness centrality* 0.888889 lebih cepat dan lebih mudah dalam berkomunikasi dengan *node* lain ketika melakukan *mention*, *quote retweet*, atau *reply* tanpa melalui banyak perantara yang dilalui. Akan tetapi, *node-node* dengan nilai *closeness centrality* 0.857143 sama baiknya dalam hal kecepatan dan kemudahan berkomunikasi dengan *node* lain tanpa melalui banyak perantara yang dilalui.

Analisis *betweenness centrality* bertujuan untuk mengetahui posisi *node* dalam *network*, dimana *node* tersebut tidak boleh hilang. Jika *node* tersebut hilang maka akan terjadi gangguan komunikasi dalam *network*. *Node* dengan *username* @gojekindonesia memiliki nilai *betweenness centrality* tertinggi dalam *network*. Artinya, @gojekindonesia adalah *account* penghubung atau jembatan dari seluruh aliran informasi dalam percakapan mengenai BlueBird Taxi. Kemudian analisis *eigenvector centrality* bertujuan untuk menemukan *account* yang memiliki performa paling baik dalam *network*. Nilai *eigenvector centrality* paling besar dimiliki oleh *username* @Bluebirdgroup sebesar 1. Selanjutnya diikuti oleh *username* @gojekindonesia dan @null, yang artinya *username* @Bluebirdgroup memiliki interaksi baik berupa *mention*, *quote retweet*, dan *reply* dengan *node-node* tertentu seperti @gojekindonesia dan @null yang mempunyai interaksi tinggi atau menjadi sumber informasi.

4.7.2 *Social Network Analysis* GoCar

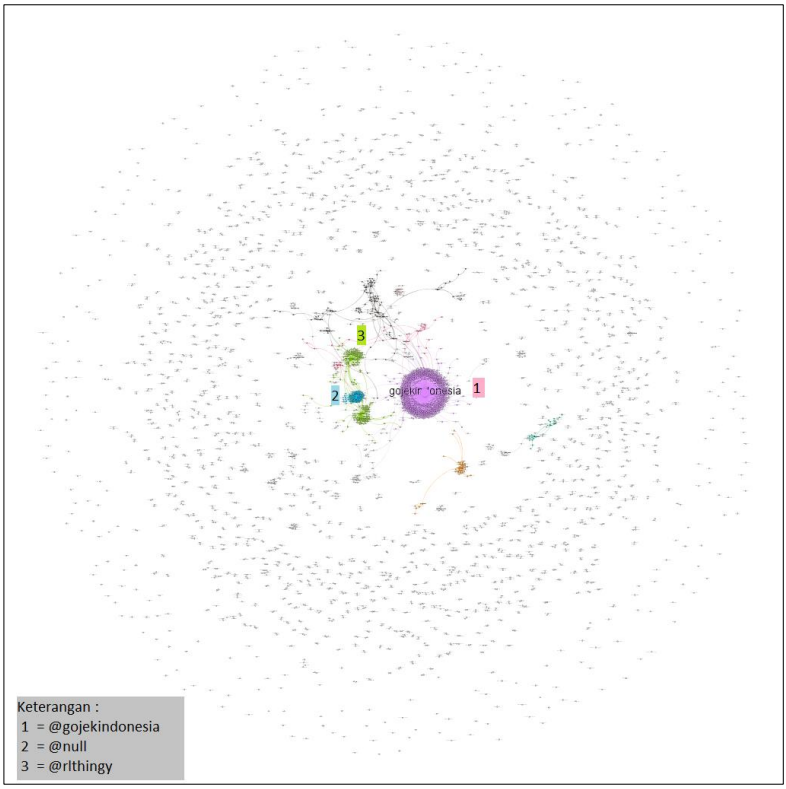
Berikut ini adalah hasil obsevasi SNA pada *keyword* GoCar dalam media sosial Twitter. Sebelumnya, dilakukan analisis dari data perbandingan jaringan yang ditunjukkan pada Tabel 4.22.

Tabel 4.22 Data Perbandingan Jaringan GoCar

Degree Range	*Nodes*	*Edges*
0	13831	3142
1	4027	3142
2	823	819
3	144	150

Pada Tabel 4.22 didapatkan jumlah *node* dan *edge* dari jaringan GoCar yaitu sebanyak 13831 *nodes* dan 3142 *edges*. Kemudian dilakukan proses untuk mengurangi *noise* dari *node* yang tidak memiliki interaksi dengan menggunakan *filter degree range*. Nilai minimal yang digunakan dalam *filter degree range* adalah satu, sehingga *node* yang muncul dalam *graph* adalah *node* yang memiliki minimal satu interaksi atau satu *edge*. Setelah dilakukan *filter degree range*, maka didapatkan jumlah *nodes* sebanyak 4027 *nodes* dan jumlah *edge* tetap. Berikut adalah hasil visualisasi *network* yang terbentuk untuk jaringan GoCar dengan *filter degree range* yang disajikan pada Gambar 4.11.

Gambar 4.11 merupakan visualisasi *network* untuk jaringan GoCar dengan menggunakan tipe jaringan *Frunchterman Reingold* dan *ForceAtlas 2*. Kedua tipe tersebut dipilih karena merupakan tipe yang umum digunakan dalam pembuatan *graph* untuk SNA. Nilai *density* yang didapatkan untuk jaringan GoCar sebesar 0. Nilai *density* sebesar 0 menandakan bahwa jaringan GoCar memiliki kepadatan yang kecil atau renggang. Nilai *density* yang kecil atau renggang ini terjadi karena rendahnya interaksi antar *account*, baik berupa *mention*, *quote retweet*, atau *reply* yang dilakukan antar *node* dalam jaringan. Berikut ini adalah hasil analisis *centrality* yang bertujuan untuk menemukan *account* yang paling berperan dalam sebuah *network* yang ditunjukkan pada Tabel 4.23.



Gambar 4.11 Visualisasi *Network* GoCar

Pada hasil analisis *degree centrality* pada Tabel 4.23 didapatkan bahwa *account* @gojekindonesia memiliki nilai *degree centrality* tertinggi yaitu 735 (pada Gambar 4.11 ditunjukkan dengan angka 1), yang artinya *account* @gojekindonesia memiliki hubungan atau interaksi yang berupa *mention*, *quote retweet*, atau *reply* antar *node* lain sebanyak 735 kali. Kemudian yang kedua adalah *account* @null dengan nilai *degree centrality* sebesar 96 (pada Gambar 4.11 ditunjukkan dengan angka 2), dan yang ketiga adalah *account* @rlthingy dengan nilai *degree centrality* sebesar 87 (pada Gambar 4.11 ditunjukkan dengan angka 3). Selanjutnya

dari nilai *closeness centrality* didapatkan tiga *account* dengan nilai tertinggi yaitu @gojekindonesia, @ffarliani, dan @qaqqah. Nilai *closeness centrality* pada peringkat pertama sebesar 0.983871 dan peringkat kedua sebesar 0.75 dengan selisih antara peringkat pertama dan kedua sebesar 0.233871 mengakibatkan *node-node* dengan nilai *closeness centrality* sebesar 0.984871 lebih cepat dan lebih mudah dalam berkomunikasi dengan *node* lain ketika melakukan *mention*, *quote retweet*, atau *reply* tanpa melalui banyak perantara. Sedangkan, *node-node* dengan nilai *closeness centrality* 0.75 perlu melalui banyak perantara untuk melakukan *mention*, *quote retweet*, dan *reply*.

Tabel 4.23 Analisis *Centrality* dari Jaringan GoCar

Centrality	***Username***	***Score***
Degree Centrality	@gojekindonesia	735
	@null	96
	@rlthingy	87
Closeness Centrality	@gojekindonesia	0.983871
	@ffarliani	0.75
	@qaqqah	0.75
Betweenness Centrality	@gojekindonesia	45095.0
	@JennyJusuf	1815.0
	@kejO_Online	1778.666667
Eigenvector Centrality	@gojekindonesia	1.0
	@Ismailm62599808	0.110057
	@ari_tengol	0.109411

Analisis *betweenness centrality* digunakan untuk mengetahui posisi *node* dalam *network*. Pada Tabel 4.23 didapatkan nilai *betweenness centrality* tertinggi pada *account* @gojekindonesia. Artinya, *account* @gojekindonesia merupakan *account* penghubung atau jembatan dari seluruh aliran informasi dalam percakapan mengenai GoCar. Hal ini mengakibatkan banyak *node* lain yang bergantung pada *tweet* yang di-*posting* oleh @gojekindonesia, karena *node* tersebut adalah sebuah jembatan bagi *node* lain sebagai sumber informasil terkait percakapan

mengenai GoCar. Selanjutnya, dari analisis *eigenvector centrality*, nilai paling besar dimiliki oleh *account* @gojekindonesia sebesar 1. Hal tersebut menunjukkan bahwa *account* @gojekindonesia memiliki interaksi baik berupa *mention, quote retweet*, dan *reply* dengan *node-node* tertentu.

4.7.3 *Social Network Analysis* GrabCar

Berikut ini adalah hasil observasi SNA pada *keyword* GrabCar dalam media sosial Twitter. Sebelum dilakukan analisis lebih lanjut, maka terlebih dahulu dilakukan perbandingan jaringan dari GrabCar yang ditunjukkan pada Tabel 4.24.

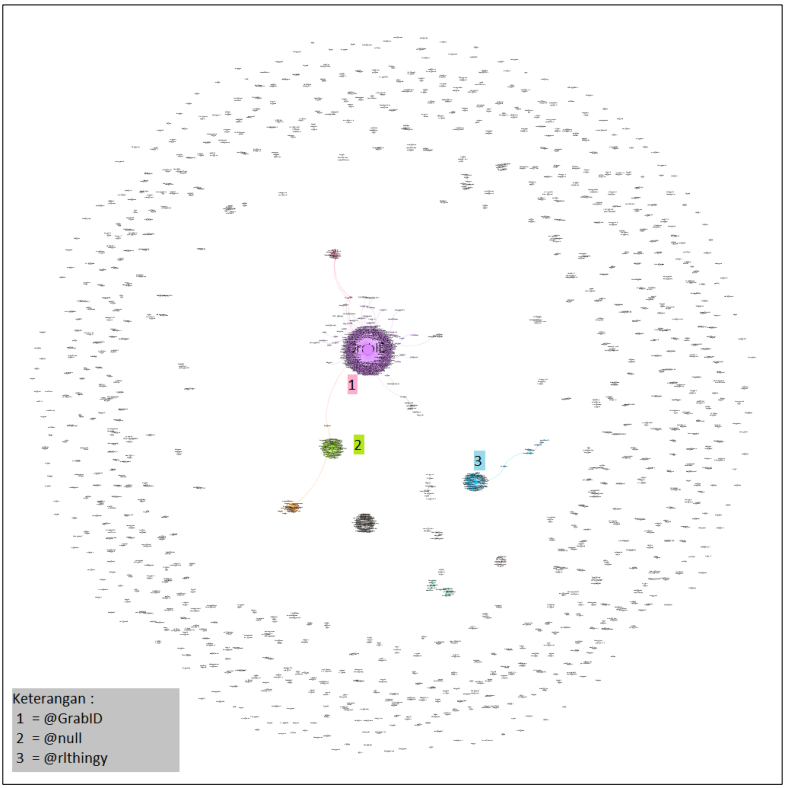
Tabel 4.24 Data Perbandingan Jaringan GrabCar

Degree Range	*Nodes*	*Edges*
0	13674	1765
1	2301	1765
2	406	371
3	67	47

Berdasarkan Tabel 4.24 didapatkan jumlah *node* dan *edge* dalam jaringan GrabCar sebanyak 13674 *nodes* dan 1765 *edges*. Jumlah *node* sebanyak 13674 *nodes* tersebut masih mengandung *node* yang tidak memiliki interaksi atau *edge*, sehingga dilakukan proses *filtering* untuk mengurangi *noise* pada *graph* dengan menggunakan *filter degree range*. Nilai f*ilter degree range* yang digunakan adalah minimal satu, yaitu hanya memunculkan *node* yang memiliki interaksi sebanyak satu. Sehingga setelah dilakukan proses *filtering*, didapatkan jumlah *node* sebanyak 2301 *nodes* dengan jumlah *edge* tetap. Berikut adalah hasil visualisasi *network* yang terbentuk untuk jaringan GrabCar dengan menggunakan *filter degree range* minimal satu yang ditunjukkan pada Gambar 4.12.

Visualisasi *network* untuk GrabCar pada Gambar 4.12 menggunakan tipe jaringan *Frunchterman Reingold* dan *ForceAtlas 2*. Kedua tipe tersebut digunakan karena merupakan tipe yang umum untuk pembuatan *graph* SNA. Dari *graph* jaringan pada Gambar 4.12 didapatkan nilai *density* sebesar 0, yang artinya

jaringan GrabCar memiliki kepadatan yang kecil atau renggang. Nilai *density* yang kecil atau renggang terjadi diakibatkan oleh rendahnya interaksi antar *account*, baik berupa *mention*, *quote retweet*, atau *reply* yang dilakukan antar *node* dalam jaringan tersebut. Berikut adalah hasil analisis *centrality* dari Gambar 4.12 yang ditunjukkan pada Tabel 4.25.



Gambar 4.12 Visualisasi *Network* GrabCar

Analisis *centrality* pada Tabel 4.25, *metric* yang digunakan dalam penentuan *centrality* adalah *degree centrality*, *closeness centrality*, *betweenness centrality*, dan *eigenvector centrality*. Dari

Tabel 4.21 didapatkan nilai *degree centrality* yang terbesar adalah pada *account* @GrabID yaitu sebesar 482 (pada Gambar 4.12 ditunjukkan dengan angka 1). Nilai *degree centrality* sebesar 482 menunjukkan bahwa *account* @GrabID memiliki hubungan atau interaksi yang berupa *mention*, *quote retweet*, atau *reply* antar *node* lain sebanyak 482 kali. Kemudian pada urutan kedua adalah *account* @null yaitu sebesar 73 (pada Gambar 4.12 ditunjukkan dengan angka 2), dan yang ketiga adalah *account* @rlthingy dengan nilai *degree centrality* sebesar 61 (pada Gambar 4.12 ditunjukkan dengan angka 3). Kemudian untuk analisis *closeness centrality* didapatkan tiga *account* dengan nilai tertinggi, yaitu @GrabID, @be_em_we323i, dan @nengbiker. Pada *account* @GrabID didapatkan nilai *closeness centrality* sebesar 0.948052 dan memiliki selisih sebesar 0.114719 dengan *account* @be_em_we323i. Hal ini mengakibatkan *node-node* dengan nilai *closeness centrality* sebesar 0.948052 lebih cepat dan lebih mudah dalam berkomunikasi dengan *node* lain ketika melakukan *mention*, *quote retweet*, atau *reply* tanpa melalui banyak perantara yang dilalui. Sedangkan, *node-node* dengan nilai *closeness centrality* sebesar 0.833333 perlu melalui banyak perantara.

Tabel 4.25 Analisis *Centrality* dari Jaringan GrabCar

Centrality	***Username***	***Score***
Degree Centrality	@GrabID	482
	@null	73
	@rlthingy	61
Closeness Centrality	@GrabID	0.948052
	@be_em_we323i	0.833333
	@nengbiker	0.8
Betweenness Centrality	@GrabID	30279.0
	@Heruu96	830
	@yonpurba	415.0
Eigenvector Centrality	@GrabID	1.0
	@rik_agus	0.159444
	@Heruu96	0.157865

Selanjutnya dilakukan analisis *betweenness centrality* yang bertujuan untuk mengetahui posisi *node* dalam *network*. Pada Tabel 4.25 didapatkan *node* dengan *username* @GrabID memiliki nilai yang paling tinggi yaitu sebesar 30279. Hal tersebut menunjukkan bahwa @GrabID merupakan *account* penghubung atau jembatan dari seluruh aliran informasi di dalam percakapan mengenai GrabCar. Begitu pula pada analisis *eigenvector centrality*, didapatkan bahwa *account* @GrabID merupakan *acoount* yang memiliki nilai *eigenvector centrality* tertinggi, kemudian diikuti oleh *account* @arik_agus dan @Heruu96. Sehingga dapat disimpulkan bahwa dari analisis *eigenvector centrality* diketahui bahwa *account* @GrabID memiliki interaksi baik berupa *mention*, *quote retweet*, dan *reply* dengan *node-node* tertentu, seperti @arik_agus dan @Heruu96 yang mempunyai interaksi tinggi atau menjadi sumber informasi.

4.7.4 *Social Network Analysis* Taksi Konvensional dan Taksi Berbasis *Online*

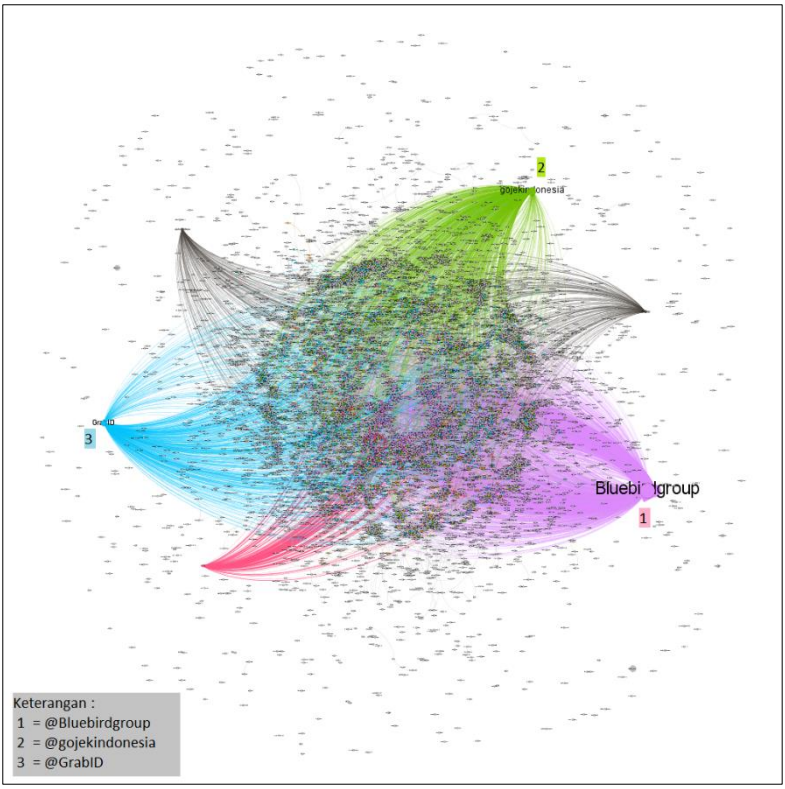
Setelah dilakukan *Social Network Analysis* pada masing-masing penyedia jasa transportasi umum taksi, selanjutnya akan dilakukan *Social Network Analysis* pada topik tentang taksi konvensional BlueBird Taxi dan taksi berbasis *online* GoCar dan GrabCar. Berikut ini adalah hasil observasi SNA pada *keyword* tentang taksi konvensional dan taksi berbasis *online* dalam media sosial Twitter. Sebelum dilakukan analisis lebih lanjut, maka terlebih dahulu dilakukan perbandingan jaringan yang ditunjukkan pada Tabel 4.26.

Tabel 4.26 Data Perbandingan Jaringan

Degree Range	*Nodes*	*Edges*
0	18201	7167
1	7525	7167
2	2186	3077
3	487	779

Dari Tabel 4.26 didapatkan jumlah *node* dan *edge* dari jaringan tentang taksi konvensional dan taksi berbasis *online*. Dapat diketahui bahwa jika tanpa menggunakan *filter degree range*, jumlah *nodes* dan *edges* yang didapatkan adalah sebanyak 18201 *nodes* dan 7167 *edges*. Sedangkan, jika menggunakan *filter degree range* dengan minimal adanya interaksi pada *edge* adalah satu, jumlah *nodes* yang didapatkan menjadi 7525 dan jumlah *edges* tetap. Hal tersebut karena jika menggunakan *filter degree range* minimal satu, maka *nodes* yang tidak memiliki interaksi *edges* minimal satu akan dihilangkan, sehingga dapat mengurangi *noise* pada *graph*. Berikut adalah hasil visualisasi *network* yang terbentuk untuk jaringan tentang taksi konvensional dan taksi berbasis *online* dengan menggunakan *filter degree range* satu yang ditunjukkan pada Gambar 4.13.

Gambar 4.13 merupakan visualisasi *network* untuk topik tentang taksi konvensional dan taksi berbasis *online* dengan menggunakan tipe jaringan *Frunchterman Reingold* dan *ForceAtlas 2*. Tipe tersebut merupakan tipe yang umum digunakan dalam pembuatan *graph* untuk SNA. Dari Gambar 4.13 didapatkan nilai *density* sebesar 0.000. Nilai *density* sebesar 0.000 menunjukkan bahwa *network* dari topik tentang taksi konvensional dan taksi berbasis *online* memiliki kepadatan yang kecil atau renggang. *Density* yang kecil atau renggang ini terjadi karena rendahnya interaksi antar *account*, baik berupa *mention*, *quote retweet*, atau *reply* yang dilakukan antar *node* dalam jaringan. Berikut adalah hasil analisis *centrality* dari Gambar 4.13 yang ditunjukkan pada Tabel 4.27.



Gambar 4.13 Visualisasi *Network*

Pada hasil analisis *degree centrality* pada Tabel 4.27 didapatkan bahwa *account* @Bluebirgroup memiliki nilai *degree centrality* tertinggi yaitu sebanyak 1652 (pada Gambar 4.13 ditunjukkan dengan angka 1), yang artinya *account* @Bluebirdgroup memiliki hubungan atau interaksi yang berupa *mention*, *quote retweet*, atau *reply* antar *node* lain sebanyak 1652 kali. Kemudian yang kedua adalah *account* @gojekindonesia dengan nilai *degree centrality* sebesar 798 (pada Gambar 4.13 ditunjukkan dengan angka 2), dan yang ketiga adalah *account* @GrabID dengan nilai *degree centrality* sebesar 492 (pada

Gambar 4.13 ditunjukkan dengan angka 3). Dari analisis *degree centrality* dapat disimpulkan bahwa *official account* penyedia jasa transportasi umum BlueBird Taxi merupakan *official account* yang paling banyak memiliki interaksi dibandingkan dengan *official account* dari taksi berbasis *online* GoCar dan GrabCar. Hal tersebut dikarenakan adanya pembatasan topik pada taksi berbasis *online* yang diambil yaitu hanya tentang pelayanan taksi, sedangkan untuk jenis pelayanan yang lain tidak diikutkan.

Tabel 4.27 Analisis *Centrality*

Centrality	***Username***	***Score***
Degree Centrality	@Bluebirdgroup	1652
	@gojekindonesia	798
	@GrabID	492
Closeness Centrality	@wintermeshach	0.8
	@nengbiker	0.8
	@bestyoland	0.8
Betweenness Centrality	@Bluebirdgroup	1023661.605813
	@gojekindonesia	208158.941392
	@GrabID	173641.886128
Eigenvector Centrality	@Bluebirdgroup	1.0
	@gojekindonesia	0.355657
	@GrabID	0.201661

Selanjutnya dari nilai *closeness centrality* didapatkan tiga *account* dengan nilai tertinggi yaitu @wintermeshach, @nengbiker, dan @bestyoland. Nilai *closeness centrality* yang didapatkan dari ketiga *account* tersebut sama, yaitu sebesar 0.8. Sehingga *node-node* dengan nilai *closeness centrality* sebesar 0.8 lebih cepat dan lebih mudah dalam berkomunikasi dengan *node* lain ketika melakukan *mention*, *quote retweet*, atau *reply* tanpa melalui banyak perantara. Sedangkan, *node-node* dengan nilai *closeness centrality* dibawah 0.8 perlu melalui banyak perantara untuk melakukan *mention*, *quote retweet*, dan *reply*.

Analisis *betweenness centrality* digunakan untuk mengetahui posisi *node* dalam *network*. Pada Tabel 4.27

didapatkan nilai *betweenness centrality* tertinggi pada *account* @Bluebirdgroup. Artinya, *account* @Bluebirdgroup merupakan *account* penghubung atau jembatan dari seluruh aliran informasi dalam percakapan mengenai taksi konvensional maupun taksi berbasis *online*. Hal ini mengakibatkan banyak *node* lain yang bergantung pada *tweet* yang di-*posting* oleh @Bluebirdgroup, karena *node* tersebut adalah sebuah jembatan bagi *node* lain sebagai sumber informasi terkait percakapan mengenai taksi konvensional dan taksi berbasis *online*. Selanjutnya, dari analisis *eigenvector centrality*, nilai paling besar dimiliki oleh *account* @Bluebirdgroup sebesar 1. Hal tersebut menunjukkan bahwa *account* @Bluebirdgroup memiliki interaksi baik berupa *mention*, *quote retweet*, dan *reply* dengan *node-node* tertentu.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan terhadap data *tweet* tentang penyedia jasa layanan transportasi umum taksi konvensional BlueBird Taxi dan penyedia jasa layanan transportasi umum taksi berbasis *online* GoCar dan GrabCar, maka dapat disimpulkan sebagai berikut.

1. Dari *tweet* sentimen yang diperoleh didapatkan bahwa pengguna lebih banyak mengunggah *tweet* bersentimen positif pada taksi berbasis *online* dibandingkan dengan taksi konvensional. Sebagian besar pengguna telah merasa terpuaskan dengan jasa yang diberikan oleh taksi berbasis *online* sehingga memberikan respon yang positif dengan mengunggah *tweet* bersentimen positif, sedangkan pengguna taksi konvensional sebagian besar masih belum merasa terpuaskan dengan jasa yang diberikan oleh taksi konvensional sehingga memberikan respon negatif terhadap jasa yang diberikan dengan mengunggah *tweet* bersentimen negatif. Dalam hal ini, kepuasan pengguna terhadap jasa yang diberikan oleh taksi berbasis *online* lebih baik jika dibandingkan dengan taksi konvensional.
2. Berdasarkan perbandingan metode NBC dan SVM untuk masing-masing jenis taksi, didapatkan keputusan bahwa dari pengukuran performa dengan rata-rata nilai *accuracy* didapatkan bahwa metode SVM dengan Kernel RBF lebih baik jika dibandingkan dengan NBC. Sehingga metode yang cocok digunakan untuk pengklasifikasian pada penelitian ini adalah metode SVM dengan Kernel RBF dengan menggunakan parameter *C* sebesar 10 dan *gamma* (γ) sebesar 1. Rata-rata nilai *accuracy* pada data *testing* yang didapatkan untuk masing-masing jenis taksi dengan metode SVM Kernel RBF yaitu sebesar 81.6% untuk BlueBird Taxi,

kemudian untuk GoCar sebesar 77.6%, sedangkan untuk GrabCar sebesar 78.5%.

3. Pada visualisasi *word cloud* didapatkan kata terbanyak yang muncul pada sentimen positif untuk BlueBird Taxi adalah 'tarif', sedangkan pada sentimen negatif adalah kata 'app'. Kemudian pada jenis taxi GoCar didapatkan kata yang banyak muncul pada *tweet* bersentimen positif dan sentimen negatif sama, yaitu 'driver'. Sedangkan pada jenis taksi GrabCar didapatkan kata yang paling banyak muncul pada *tweet* bersentimen positif adalah kata 'ngobrol' dan pada *tweet* bersentimen negatif adalah kata 'driver'.
4. Hasil analisis dengan menggunakan *Social Network Analysis* pada jaringan dengan topik pembahasan tentang taksi konvensional dan taksi berbasis *online* didapatkan jumlah *node* dan *edge* yang terbentuk adalah sebanyak 18201 *nodes* dan 7167 *edges* dengan *account* yang memiliki hubungan atau interaksi yang berupa *mention*, *quote retweet*, atau *reply* tertinggi antar *node* lain adalah @Bluebirdgroup, kemudian @gojekindonesia, dan @GrabID. Selain itu, dari analisis *betweenness centrality* didapatkan bahwa *account* @Bluebirdgroup merupakan *account* penghubung atau jembatan dari seluruh aliran informasi dalam percakapan mengenai taksi konvensional maupun taksi berbasis *online*.

5.2 Saran

Saran yang dapat diberikan oleh peneliti setelah melakukan penelitian mengenai analisis kompetitif sosial media dengan menggunakan metode klasifikasi *Naïve Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM), serta *Social Network Analysis* (SNA) pada industri transportasi umum taksi konvensional dan taksi berbasis *online* adalah sebagai berikut.

a. Pada penelitian tentang pengklasifikasian *tweet* ini sangat membutuhkan klasifikasi sentimen atau *tweet* yang mengandung sentiment lebih banyak atau cukup besar agar hasil ketepatan klasifikasi yang digunakan dapat lebih baik.

Disarankan pula data yang digunakan memiliki proporsi yang seimbang untuk tiap kategorinya dengan tujuan untuk menambah tingkat akurasi.

b. Pada praproses teks diharapkan pada penelitian selanjutnya dapat menambahkan atau mengatasi kata-kata khusus untuk kata-kata *slang* atau bahasa sehari-hari pada proses *stemming* dan *stopwords* dan dilengkapi dengan daftar kata singkatan, karena di Twitter bahasa yang sering digunakan adalah bahasa sehari-hari serta singkatan-singkatan yang tidak baku karena adanya keterbatasan jumlah karakter huruf pada Twitter.

c. Peneliti berharap pada penelitian selanjutnya tentang *Social Network Analysis* (SNA) dapat dilakukan lebih detail lagi dan membuat inovasi pada bentuk-bentuk *graph* untuk merepresantikan *graph* yang lebih baik, lebih bagus, dan lebih menarik.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

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

Ariadi, D. & Fithriasari, K. (2015). Klasifikasi Berita Indonesia Menggunakan Metode Naïve Bayes Classification dan Support Vector Machine dengan Confix Stripping Stemmer. *Jurnal Sains dan Seni ITS, 4(2)*, 2337-3520.

Asiyah, S. N., & Fithriasari, K. (2016). Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K-Nearest Neighbor. *Jurnal Sains dan Seni ITS, 5(2)*, 2337-3520.

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

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

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

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

Cheliotis, G. (2010). *Social Network Analysis (SNA) Including a Tutorial on Concepts and Methods*. Singapore: Communications and New Media, National University of Singapore.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machine*. Cambridge: Cambrige University Press.

Culotta, A. (2010). Detecting Influenza Outbreaks by Analyzing Twitter Messages. *Journal of Department of Computer Science Southeastern Louisiana University*, 1 – 11.

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

Durajati, C., & Gumelar, A. B. (2012). Pemanfaatan Teknik Supervised Untuk Klasifikasi Teks Bahasa Indonesia. *Jurnal Link Vol 16/No. 1*, 1-8. *ISSN 1858 – 4667*.

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

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

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

Guduru, N. (2006). *Text Mining With Support Vector Machines and Non-Negative Matrix Factorization Algorithms*. University of Rhode Island.

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

Hamzah, A. (2012). Klasifikasi Teks dengan Naïve Bayes Classifier (NBC) untuk Pengelompokkan Teks Berita dan Abstract Akademis. *In Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III, p. B269-B277*. Yogyakarta.

Han, J. Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques, Third Edition*. USA: Elsevier.

Härdle, W. K., Prastyo, D. D., & Hafner, C. (2012). Support Vector Machines with Evolutionary Feature Selesction for Default Prediction. *SFB 649 Discussion Paper 2012-030*.

Hearst, M. A. (1997). Text Data Mining: Issues, Techniques, and The Relationship to Information Acces. *Presentation Notes for UW/MS Workshop on Data Mining*.

Hendriyani. (2013). *Penggunaan Jejaring Sosial Akun Twitter @STANDUPINDOBDG dalam Penyampaian Informasi*

Seputar Kegiatan Standup Comedy kepada Followers-nya di Bandung. Bandung: Jurusan Ilmu Politik & Sosial, Fakultas Ilmu Komunikasi, Universitas Komputer Indonesia.

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

Hsu, C. W., Chang, C. C., & Lin, C. J. (2010). *A Practical Guide to Support Vector Classification*. Taiwan: Department of Computer Science National Taiwan University.

Huang, C. M., Lee, Y. J., Lin, D. K., & Huang, S. Y. (2007). Model Selection For Support Vector Machines Via Uniform Design. *Computational Statistics & Data Analysis*, 335 – 346.

Huang, M. L., Hung, Y. H., Lee, W. M., Li, R. K., & Jiang, B. R. (2014). SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier. *The Scientific World Journal, 1-10*.

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text Classification Using Machine Learning Techniques. *WSEAS Transactions on Computers Issue, 8(4)*, 966 – 974.

Ilmawan, N. (2016). *Analisis Dampak Perkembangan Taksi Berbasis Online Terhadap Taksi Konvensional Serta Pengaruhnya di Sektor Ekonomi dan Lingkungan (Studi Kasus: Provinsi DKI Jakarta)*. Surabaya: Jurusan Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Joachims, T. (1998). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Germany: Universitat Dortmund.

Kementrian Komunikasi dan Informatika Republik Indonesia. (2017). *Laporan Tahunan 2016*. Jakarta: Menteri Komunikasi dan Informatika.

Kumar, S., Morstatter, F., & Liu, H. (2013). *Twitter Data Analysis*. New York: Springer.

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

Ma'ady, M. N. P. (2014). *Penerapan Klasifikasi Teks Untuk Pemetaan Data Twitter Pelanggan PT. Telekomunikasi Seluler Dalam Bentuk Heat Map*. Surabaya: Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Nuansa, E. P. (2017). *Analisis Sentimen Pengguna Twitter Terhadap Pemilihan Gubernur DKI Jakarta dengan Metode Naïve Bayesian Classification dan Support Vector Machine*. Surabaya: Departemen Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember.

Oktora, R., & Alamsyah, A. (2014). Pola Interaksi dan Aktor yang Paling Beperan Pada Event JGTC 2013 Melalui Media Sosial Twitter (Studi Menggunakan Metode Social Network Analysis). *Jurnal Manajemen Indonesia, 14(3),* 201 – 209.

Pfeil, U., &Zaphiris, P. (2009).Investigating Social Network Patterns Within an Empathic Online Community for Older People. *Computers in Human Behavior, 25*, 1139 - 1155.

Pratama, B. Y. (2015). *Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naïve Bayes, KNN, dan SVM*. Surabaya: Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Rish, I. (2001). An Empirical Study of The Naive Bayes Classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 3(22),* 41-46.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys, 34(1),* 1-47.

Siswanto, J., Riyanto, B., & Sunaryo, B. (2010). Analisis Pelayanan Taksi di Kota Semarang Berdasarkan Pendekatan Keseimbangan Permintaan dan Pendapatan Operator. *Simposium XIII FSTPT, Universitas Katolik Soegijapranata Semarang*.

Sucipto, R. H., & Alamsyah, A. (2016). Analisis Jaringan Teks Berdasarkan Social Network Analysis dan Text Mining

Untuk Mengetahui Persepsi Kualitas Merek Pada Konten Percakapan di Media Sosial Twitter (Studi Pada PT Indosat Tbk. Dan PT Telkomsel. *e-Proceeding of Management*, *3(1)*, 93 – 100. *ISSN: 2355 – 9357*.

Sun, Y., Kamel, M. S., & Wang, Y. (2006). Boosting for Learning Multiple Classes with Im-balanced Class Distribution. *Sixth International Conference on Data Mining (ICDM'06)*, 421 – 431.

Sunni, I., & Widyantoro, H. D. (2012). Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik. *Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika, 1(2),* 200-206.

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

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

Tsvetovat, M., & Kouznetsov, A. (2011). *Social Network Analysis for Startups*. USA: O'reilly Media.

Twitter. (2018). *Twitter Support*. [Online]. Tersedia: http://support.twitter.com/. [08 Februari 2018].

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

Widhianingsih, T. D. A.,& Fithriasari, K. (2016). Aplikasi Text Mining untuk Automasi Klasifikasi Artikel dalam Majalan Online Wanita Menggunakan Naïve Bayes Classifier (NBC) dan Artificial Neural Network (ANN). *Jurnal Sains dan Seni ITS, 5(1)*.

Wiki Book. (2011). *Social Network Analysis: Theory and Applications*. [Online]. Tersedia: https://www.politaktiv.org /documents/10157/29141/SocNet_TheoryApp.pdf. [08 Februari 2018].

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques, Third Edition*. USA: Elsevier.

Zaky, M. J., & Meira, Jr. W. (2014). *Data Mining and Analysis: Foundations and Algorithms*. New York: Cambridge University Press.

LAMPIRAN

Lampiran 1. *Syntax Crawling* Data Menggunakan RStudio

```
library(twitteR)

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

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

#Searching Tweet
gocar <- searchTwitter('gocar', lang="id",
n=15000, resultType="recent")
write.csv(twListToDF(gocar), file="gocar.csv")

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

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

Lampiran 2. Data *Tweet* BlueBird Taxi dari Twitter API

***Tweet* Ke-**	**Class**	***Tweet***	**ReplyToSN**	***ScreenName***
1	0	Nawar bajay dr Tanah Abang ke Gambir. Si abang buka harga 35rb (kaget).. Tawar 20, dia turunin cuma sampe 25. Dalam… https://t.co/w6AJjyXooY	NA	firman23
2	1	antrian taksi non bluebird 40sekian, taksi bluebird 80 sekian. semua taksi online ga bisa diakses saking padatnya permintaan. Jakarta ;(	NA	let666be
3	1	Berkat bluebird gak ada yg pickup, dan gocar harganya jadi 2x lipat. Suda tida paham lagi.	NA	riniiik
4	1	Tp nggak apa, ngirit juga naik brt. @Bluebirdgroup easyride lagi error ya.. beberapa kali saya naik bluebird dan booking pembayaran melalui kartu tidak bisa	Bluebirdgroup	teguhekas
.	.	.	.	.
.	.	.	.	.
3109	1	@Bluebirdgroup taxi blue bird no pintu RD 3699 Limo parkir dari shubuh tadi didepan rumah tetangga saya. Supir ga a… https://t.co/zITx2x3udy	Bluebirdgroup	Wina_aswir
3110	1	@Bluebirdgroup sampai saat ini tidak ada follow up dr blue bird untuk saya sebagai customer	wishnuturangan	wishnuturangan
3111	1	@Bluebirdgroup Saya pelanggan setia blue bird tetapi hari ini saya order jam 5.30 sampai saat email ini di kirim sa… https://t.co/hiAM5Ocm91	Bluebirdgroup	wishnuturangan

Keterangan : 0 = Positif
1 = Negatif

Lampiran 3. Data *Tweet* GoCar dari Twitter API

***Tweet* Ke-**	**Class**	***Tweet***	**ReplyToSN**	***ScreenName***
1	0	terima kasih go car.....sudah antar jemput.....murah sekali https://t.co/PR9BpCFWOd	NA	Solichinzm
2	1	Halo @gojekindonesia saya naik gocar jarak pendek 8k #gopay tapi driver ga mau alasannya posisinya jauh. Dia minta… https://t.co/77ZCuDtsmY	NA	bundaPandu2013
3	1	@gojekindonesia selamat sore min, mau lapor dong. td pagi saya pesen go car dengan pembayaran gopay. saldonya udh kepotong. tp… #Jakarta	gojekindonesia	BeritaIndonesa
4	1	@gojekindonesia selamat sore min, mau lapor dong. td pagi saya pesen go car dengan pembayaran gopay. saldonya udh k… https://t.co/8Oa7Pk7bLy	gojekindonesia	juliolim13
.	.	.	.	.
.	.	.	.	.
4332	1	@gojekindonesia min, saya order gocar, saya blm naik tp udh on the way with you statusnya https://t.co/1Sb6D5dqja	gojekindonesia	rereniren
4333	0	@askmenfess Gatau seenaknya aja. Kalo malem2 naik motor gapapa. Cuma kalo siang2 mending naik gocar bersama drpd na… https://t.co/Tj6Hq5zA9t	askmenfess	95pals
4334	1	@gojekindonesiaa Aplikasi gojek lagi error kah? Tadi saya 3x coba order gocar, dapat driver posisi di ancol, sem… https://t.co/IYHgUN9HH9	gojekindonesia	tubiwityu

Keterangan : 0 = Positif
1 = Negatif

Lampiran 4. Data *Tweet* GrabCar dari Twitter API

Tweet Ke-	Class	*Tweet*	ReplyToSN	*ScreenName*
1	1	Udah gupuh banget sampek cancel grab car, ngebut naik grabbike, eeeh kretanya telat. Harusnya jam 21.00. Sekarang 2… https://t.co/Ed4dDsa3xQ	NA	lalanuradhilla
2	1	Selalu ada pengalaman buruk naik grab car. Driver yang kasar, tak berbudi bahasa, suruh cancel trip. Lepas ni kalau… https://t.co/JCzhRfRy9d	NA	ainunlmuaiyanah
3	1	@GrabID latest update apps grab ada bugs gak bisa pesan grab car. Dari grab gak ada solusi dan gak niat benerin. Sa… https://t.co/ZWPXghETjP	GrabID	JF897
4	1	Pesen grabcar dari apartemen buat balik kos. Sekalian bawa balik koper + tentengan duty free. Drivernya bantuin ang… https://t.co/edvxz4Nq9G	NA	Rerefransiska
.	.	.	.	.
.	.	.	.	.
2284	1	kecurigaanku terjawab. beberapa kali order grabcar dan selalu beda mobil & platnya sama di app. ternyata doi beli akun bodong yawlaa	NA	sofiarinda
2285	0	Setelah sekian lama gak naik grabcar akhirnya hari ini naik grabcar karena lebih murah dr gocar. Dan yah dapet lagi… https://t.co/misQ3EEgXg	NA	Aderinaa
2286	1	@GrabID sy order grabcar, driver udh blg bntar lg dia jemput sy eh tiba2 muncul rating, kaya sy udh diantar. Gmn tuh? Pdhl ga ketemu sm skli	GrabID	dllazz

Keterangan : 0 = Positif
1 = Negatif

Lampiran 5. *Syntax Input* dan Praproses Data Menggunakan Python 3.6

```
import pandas as pd
import pandas as dataframe
import string
import nltk
from nltk.tokenize import word_tokenize
import re
import sys
import os
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory
from IPython.display import display

#Input Data
bluebird = pd.read_csv('D:/bluebird.csv', engine =
'python')
print(bluebird)
bluebirdtweet = bluebird['text']
print (bluebirdtweet)

#Cleansing
#Menghapus Link
bluebirdclearlink = []
for line in bluebirdtweet:
    result = re.sub(r"http\S+","",line)
    bluebirdclearlink.append(result)

#Menghapus Simbol RT
bluebirdclearrt = []
for line in bluebirdclearlink:
    result = re.sub(r"RT","",line)
    bluebirdclearrt.append(result)

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

#Menghapus Hastag
bluebirdclearhastag = []
for line in bluebirdclearusername:
    result = re.sub(r"3(\w+)","",line)
    bluebirdclearhastag.append(result)
```

Lampiran 5. *Syntax Input* dan Praproses Data Menggunakan Python 3.6 (Lanjutan)

```
#Menghapus Nama
bluebirdclearname = []
for line in bluebirdclearhastag:
    result = re.sub(r"bluebird","",line)
    bluebirdclearname.append(result)

#Clear Space Enter
bluebirdclear = []
for line in bluebirdclearname:
    result=re.sub(r"…"," ",line)
    bluebirdclear.append(result)

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

#Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
bluebirdstemmed = map(lambda x: stemmer.stem(x),
bluebirdlower)
bluebird_no_punc = map(lambda x:
x.translate(str.maketrans('','', string.punctuation)),
bluebirdstemmed)
bluebird_no_punc = list(bluebird_no_punc)

#Stopwords dan Tokenizing
stopwords = open('D:/stoplist.txt', 'r').read()
bluebirdtrain = []
bluebirdfinal = []
df = []
for line in bluebird_no_punc:
    word_token = word_tokenize(line)
    word_token = [word for word in word_token if not
   word in stopwords and not word[0].isdigit()]
    bluebirdfinal.append(word_token)
    df.append(" ".join(word_token))
```

Lampiran 5. *Syntax Input* dan Praproses Data Menggunakan Python 3.6 (Lanjutan)

```
for l in bluebirdfinal:
    bluebirdtrain+= l
final_bluebird={v: bluebirdtrain.count(v) for v in
set(bluebirdtrain)}

import csv
with open ('D:/final_bluebird.csv','w',newline="") as
csv_file:
    writer = csv.writer(csv_file)
    for key, value in final_bluebird.items():
       writer.writerow([key, value])
```

Lampiran 6. Frekuensi Kata Kunci

BlueBird Taxi		**GoCar**		**GrabCar**	
Kata Kunci	**Frek**	**Kata Kunci**	**Frek**	**Kata Kunci**	**Frek**
Driver	463	Driver	1392	Mitra	456
App	457	Uber	916	Ngobrol	450
Tumpang	327	Selamat	772	Timdudukdepan	438
Supir	319	Rating	757	Inspirasi	429
Bandara	263	Bahas	756	Driver	393
Tarif	249	Aneh	753	Pake	194
Gojek	248	Simpel	753	Katague	144
Org	210	Gojek	389	Grabbike	142
Nyegat	202	Order	346	Mobil	140
Gada	200	Pake	330	Order	139
sumber	195	mobil	201	uber	133
jalan	194	banget	179	pergi	129
aplikasi	172	drivernya	175	aplikasi	123
online	172	jalan	173	bike	111
pake	165	abang	146	pilih	106
manado	156	pagi	144	tinggal	105
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
genang	1	bapaknyah	1	palsu	1
soeta3	1	setdaahhhh	1	reflek	1
kpan	1	mangkal	1	ehhhh	1

Lampiran 7. *Syntax Naïve Bayes Classifier* (NBC) Menggunakan Python 3.6

```
#Count Vectorizer
from pandas import DataFrame
from sklearn.feature_extraction.text import
CountVectorizer
vect = CountVectorizer(min_df=0., max_df=1.0)
X = vect.fit_transform(df)
cv = DataFrame(X.A, columns=vect.get_feature_names())
print (cv)

#tf-idf
from sklearn.feature_extraction.text import
TfidfTransformer
tfidf =
TfidfTransformer(use_idf=True).fit_transform(cv)
tfidf_train = (tfidf.toarray())
print (tfidf_train)
print (tfidf_train.shape)
tf = DataFrame(tfidf.A,
columns=vect.get_feature_names())
print (tf)

#Analisis Naïve Bayes Classifier
data_sentimen = grabcar['class']
sentiment = pd.DataFrame(data_sentimen)

from sklearn.model_selection import
train_test_split,cross_val_score,StratifiedKFold,KFold
,ShuffleSplit,StratifiedShuffleSplit
from sklearn.feature_selection import
SelectPercentile, f_classif
from sklearn.model_selection import StratifiedKFold
from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection import GridSearchCV
import numpy as np

X_train,X_test,Y_train,Y_test=train_test_split(x_new,s
entiment,test_size=0.1,random_state=43)

probasbaru = [BernoulliNB().fit(X_train,
Y_train).predict_proba(X_train)]
results.append(probasbaru)
```

Lampiran 7. *Syntax Naïve Bayes Classifier* (NBC) Menggunakan Python 3.6 (Lanjutan)

```
probas = BernoulliNB()
probas.fit(X_train, Y_train)
y_pred = probas.predict(X_test)
print(confusion_matrix(Y_test,y_pred))
print(classification_report(Y_test,y_pred))

# get class probabilities for the first sample in the
dataset
class1_1 = [pr[0, 0] for pr in probasbaru]
class2_1 = [pr[0, 1] for pr in probasbaru]
print ('-----------------------------------------------
----------------------')
print (probasbaru)
print ('hasil klasifikasi positif', class1_1)
print ('hasil klasifikasi negatif', class2_1)

YB = DataFrame.as_matrix(sentiment)
print (YB)

X_baru, Y = tfidf_train, YB
kf = StratifiedKFold (n_splits=10)
kf.get_n_splits(X_baru)
cl = BernoulliNB()

for train, test in kf.split(X_baru, YB):
    (X_baru[train],X_baru[test])
    (Y[train], Y[test])
    a = cl.fit(X_baru[train], Y[train])
    b = cl.predict(X_baru[test])
    fpr, tpr, _ = metrics.roc_curve(Y[test], b)
    auc = metrics.auc(fpr,tpr)
    print (b)
    print(confusion_matrix(Y[test], b))
    print(classification_report(Y[test],b))
    print("Area Under Curve ROC = {:.2f}%
".format(auc*100))
```

Lampiran 8. *Confusion Matrix* Metode NBC pada BlueBird Taxi

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	933	39	1671	156	2799
	2	926	34	1676	164	2800
	3	932	33	1677	158	2800
	4	920	36	1674	170	2800
	5	924	31	1679	166	2800
	6	928	29	1681	162	2800
	7	939	33	1677	151	2800
	8	909	36	1674	181	2800
	9	934	41	1669	156	2800
	10	926	34	1676	164	2800
Testing	1	58	14	176	64	312
	2	57	29	161	64	311
	3	76	20	170	45	311
	4	99	18	172	22	311
	5	73	22	168	48	311
	6	76	45	145	45	311
	7	120	20	170	1	311
	8	97	1	189	24	311
	9	102	13	177	19	311
	10	55	13	177	66	311

Lampiran 9. *Confusion Matrix* Metode NBC pada GoCar

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	1907	77	1595	321	3900
	2	1977	126	1546	251	3900
	3	2012	147	1525	216	3900
	4	2011	150	1522	217	3900
	5	1948	111	1561	280	3900
	6	1916	92	1580	312	3900
	7	1904	84	1588	325	3901
	8	1874	72	1600	355	3901
	9	1904	79	1594	325	3902
	10	1901	75	1598	328	3902

Lampiran 9. *Confusion Matrix* Metode NBC pada GoCar (Lanjutan)

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Testing	1	109	14	172	139	434
	2	194	32	154	54	434
	3	247	40	146	1	434
	4	232	36	150	16	434
	5	199	24	162	49	434
	6	137	22	164	111	434
	7	106	25	161	141	433
	8	104	20	166	143	433
	9	154	24	161	93	432
	10	130	19	166	117	432

Lampiran 10. *Confusion Matrix* Metode NBC pada GrabCar

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	1163	88	760	45	2056
	2	1167	93	755	41	2056
	3	1165	98	750	43	2056
	4	1165	96	753	44	2058
	5	1147	80	769	62	2058
	6	1152	78	771	57	2058
	7	1152	76	773	57	2058
	8	1165	83	766	44	2058
	9	1155	85	764	54	2058
	10	1154	84	765	55	2058
Testing	1	108	24	71	27	230
	2	132	21	74	3	230
	3	128	28	67	7	230
	4	129	24	70	5	228
	5	94	21	73	40	228
	6	71	14	80	63	228
	7	80	19	75	54	228
	8	129	18	76	5	228
	9	104	25	69	30	228
	10	85	20	74	49	228

Lampiran 11. *Syntax Support Vector Machine* (SVM) Menggunakan Python 3.6

```
data_sentimen = grabcar['class']
sentiment = pd.DataFrame(data_sentimen)

from __future__ import print_function
from sklearn.model_selection import
train_test_split,cross_val_score,StratifiedKFold,KFold
,ShuffleSplit,StratifiedShuffleSplit
from sklearn.feature_selection import
SelectPercentile, f_classif
from sklearn.model_selection import StratifiedKFold
from sklearn.svm import LinearSVC, SVC
from sklearn import svm, datasets
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.utils import shuffle
import numpy as np

X_train,X_test,Y_train,Y_test=train_test_split(tfidf_t
rain,sentiment,test_size=0.1,random_state=43)

# Spot Check Algorithms
tuned_parameters = [{'kernel':['linear'], 'C':[0.01,
0.1, 1, 10, 100, 1000, 10000]}, {'kernel':['rbf'],
'C':[0.01, 0.1, 1, 10, 100, 1000, 10000],
'gamma':[1000, 100, 10, 1, 0.1, 0.01, 0.001]}]
scores = ['precision', 'recall']
for score in scores:
    print ("# Turning hyper-parameters for %s" %
score)
    print ()

    clf = GridSearchCV(SVC(), tuned_parameters,
cv=KFold(n_splits=10, random_state=0))
    clf.fit(X_train, Y_train)

    print("Best parameters set found on development
set:")
    print()
    print(clf.best_params_)
    print()
    print("Grid scores on training set:")
    print()
```

Lampiran 11. *Syntax Support Vector Machine* (SVM) Menggunakan Python 3.6 (Lanjutan)

```
means = clf.cv_results_['mean_test_score']
    stds = clf.cv_results_['std_test_score']
    for mean, std, params in zip(means, stds,
clf.cv_results_['params']):
        print("%0.3f (+/-%0.3f) for %r"
              % (mean, std * 2, params))
    print()

    print("Detailed classification report:")
    print()
    print("The model is trained on the full
development set.")
    print("The scores are computed on the full
evaluation set.")
    print()
    y_true, y_pred = Y_test, clf.predict(X_test)
    print(confusion_matrix(y_true, y_pred))
    print(classification_report(y_true, y_pred))
    print()

YB = DataFrame.as_matrix(sentiment)
print (YB)

X_baru, Y = tfidf_train, YB
kf = StratifiedKFold (n_splits=10)
kf.get_n_splits(X_baru)
cl = SVC(kernel='linear', C=1)
print (kf)

for train, test in kf.split(X_baru, YB):
    (X_baru[train],X_baru[test])
    (Y[train], Y[test])
    a = cl.fit(X_baru[train], Y[train])
    b = cl.predict(X_baru[test])
    fpr, tpr, _ = metrics.roc_curve(Y[test], b)
    auc = metrics.auc(fpr,tpr)
    print (b)
    print(confusion_matrix(Y[test], b))
    print(classification_report(Y[test],b))
    print("Area Under Curve ROC = {:.2f}%
".format(auc*100))
```

Lampiran 11. *Syntax Support Vector Machine* (SVM) Menggunakan Python 3.6 (Lanjutan)

```
for train, test in kf.split(X_baru, YB):
    (X_baru[train],X_baru[test])
    (Y[train], Y[test])
    a = cl.fit(X_baru[train], Y[train])
    b = cl.predict(X_baru[train])
    fpr, tpr, _ = metrics.roc_curve(Y[train], b)
    auc = metrics.auc(fpr,tpr)
    print (b)
    print(confusion_matrix(Y[train], b))
    print(classification_report(Y[train],b))
    print("Area Under Curve ROC = {:.2f}%
".format(auc*100))
```

Lampiran 12. *Confusion Matrix* Metode SVM Kernel Linear pada BlueBird Taxi

Data	***Fold* ke-**	***True Positive***	***False Positive***	***True Negative***	***False Negative***	**Jumlah *Tweet***
Training	1	1019	27	1683	70	2799
	2	1023	24	1686	67	2800
	3	1015	22	1688	75	2800
	4	1017	22	1688	73	2800
	5	1020	21	1689	70	2800
	6	1015	25	1685	75	2800
	7	1018	28	1682	72	2800
	8	1010	28	1682	80	2800
	9	1018	28	1682	72	2800
	10	1023	21	1689	67	2800
Testing	1	46	8	182	76	312
	2	67	18	172	54	311
	3	79	15	175	42	311
	4	105	15	175	16	311
	5	59	28	162	62	311
	6	71	35	155	50	311
	7	118	18	172	3	311
	8	96	0	190	25	311
	9	99	17	173	22	311
	10	61	17	173	60	311

Lampiran 13. *Confusion Matrix* Metode SVM Kernel Linear pada GoCar

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	2125	68	1604	103	3900
	2	2124	93	1579	104	3900
	3	2129	88	1584	99	3900
	4	2131	98	1574	97	3900
	5	2135	90	1582	93	3900
	6	2139	81	1591	89	3900
	7	2125	70	1602	104	3901
	8	2129	72	1600	100	3901
	9	2128	75	1598	101	3902
	10	2137	70	1603	92	3902
Testing	1	175	43	143	73	434
	2	213	51	135	35	434
	3	247	57	129	1	434
	4	240	60	126	8	434
	5	218	39	147	30	434
	6	185	39	147	63	434
	7	146	45	141	101	433
	8	179	35	151	68	433
	9	198	51	134	49	432
	10	165	34	151	82	432

Lampiran 14. *Confusion Matrix* Metode SVM Kernel Linear pada GrabCar

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	1176	19	829	32	2056
	2	1179	25	823	29	2056
	3	1173	22	826	35	2056
	4	1177	21	828	32	2058
	5	1167	16	833	42	2058
	6	1159	14	835	50	2058
	7	1138	9	840	71	2058
	8	1177	18	831	32	2058
	9	1176	16	833	33	2058
	10	1149	11	838	60	2058

Lampiran 14. *Confusion Matrix* Metode SVM Kernel Linear pada GrabCar (Lanjutan)

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Testing	1	110	23	72	25	230
	2	133	24	71	2	230
	3	128	32	63	7	230
	4	128	27	67	6	228
	5	89	18	76	45	228
	6	67	8	86	67	228
	7	52	13	81	82	228
	8	129	17	77	5	228
	9	85	23	71	49	228
	10	66	12	82	68	228

Lampiran 15. *Confusion Matrix* Metode SVM Kernel RBF pada BlueBird Taxi

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	1086	11	1699	3	2799
	2	1082	4	1706	8	2800
	3	1088	12	1698	2	2800
	4	1079	3	1707	11	2800
	5	1080	3	1707	10	2800
	6	1087	9	1701	3	2800
	7	1080	4	1706	10	2800
	8	1080	5	1705	10	2800
	9	1081	5	1705	9	2800
	10	1083	5	1705	7	2800
Testing	1	40	6	184	82	312
	2	51	18	172	70	311
	3	74	11	179	47	311
	4	95	13	177	26	311
	5	58	21	169	63	311
	6	65	10	180	56	311
	7	119	10	180	2	311
	8	96	0	190	25	311
	96	8	182	25	311	96
	53	11	179	68	311	53

Lampiran 16. *Confusion Matrix* Metode SVM Kernel RBF pada GoCar

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	2225	7	1665	3	3900
	2	2224	6	1666	4	3900
	3	2224	7	1665	4	3900
	4	2225	7	1665	3	3900
	5	2225	7	1665	3	3900
	6	2224	7	1665	4	3900
	7	2225	6	1666	4	3901
	8	2226	8	1664	3	3901
	9	2225	7	1666	4	3902
	10	2227	7	1666	2	3902
Testing	1	167	39	147	81	434
	2	211	46	140	37	434
	3	247	53	133	1	434
	4	241	44	142	7	434
	5	218	36	150	30	434
	6	151	34	152	97	434
	7	135	40	146	112	433
	8	179	29	157	68	433
	9	191	38	147	56	432
	10	157	34	151	90	432

Lampiran 17. *Confusion Matrix* Metode SVM Kernel RBF pada GrabCar

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Training	1	1207	3	845	1	2056
	2	1207	4	844	1	2056
	3	1208	4	844	0	2056
	4	1208	4	845	1	2058
	5	1208	3	846	1	2058
	6	1209	3	846	0	2058
	7	1208	4	845	1	2058
	8	1208	4	845	1	2058
	9	1208	3	846	1	2058
	10	1208	4	845	1	2058

Lampiran 17. *Confusion Matrix* Metode SVM Kernel RBF pada GrabCar (Lanjutan)

Data	*Fold* ke-	*True Positive*	*False Positive*	*True Negative*	*False Negative*	Jumlah *Tweet*
Testing	1	105	16	79	30	230
	2	133	22	73	2	230
	3	128	27	68	7	230
	4	129	21	73	5	228
	5	84	15	79	50	228
	6	70	10	84	64	228
	7	65	11	83	69	228
	8	129	19	75	5	228
	9	103	17	77	31	228
	10	73	9	85	61	228

Lampiran 18. *Output* Persamaan *Hyperplane* BlueBird Taxi

```
Kernel used:
  RBF Kernel: K(x,y) = exp(-1.0*(x-y)^2)

Classifier for classes: 0, 1

BinarySMO

-  1.2321 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  1.3167 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  10     * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  0.525  * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  1.2677 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  10     * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  10     * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
.
.
.
-  10     * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  0.7883 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  0.5974 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  0.2219

Number of support vectors: 2766
```

Lampiran 19. *Output* Persamaan *Hyperplane* GoCar

```
Kernel used:
  RBF Kernel: K(x,y) = exp(-1.0*(x-y)^2)

Classifier for classes: 0, 1

BinarySMO

-  0.7201 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.9657 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  1.1104 * < 0 0 0 0 0 0 0 0 0.166626 0 ... 0 0 0> * X]
-  0.8803 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.655  * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  3.0518 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  1.6109 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
.
.
.
-  0.8129 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.9078 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.8828 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.1029

Number of support vectors: 3883
```

Lampiran 20. *Output* Persamaan *Hyperplane* GrabCar

```
Kernel used:
  RBF Kernel: K(x,y) = exp(-1.0*(x-y)^2)

Classifier for classes: 0, 1

BinarySMO

   1.1245 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.8394 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  1.1816 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  9.7768 * < 0 0 0 0 0 0 0 0 0 0 ...0 0 0> * X]
-  0.8789 * < 0 0.459174 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  1.1012 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.8851 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
.
.
.
```

Lampiran 20. *Output* Persamaan *Hyperplane* GrabCar (Lanjutan)

```
-  0.8646 * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
+  10     * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  10     * < 0 0 0 0 0 0 0 0 0 0 ... 0 0 0> * X]
-  0.1097

Number of support vectors: 2044
```

Lampiran 21. *Syntax Word Cloud* Menggunakan Python 3.6

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS

#mpl.rcParams['figure.figsize']=(8.0,6.0)  #(6.0,4.0)
mpl.rcParams['font.size']=12               #10
mpl.rcParams['savefig.dpi']=100            #72
mpl.rcParams['figure.subplot.bottom']=.1
wordcloud = WordCloud(collocations = False,
                          background_color='white',
                          stopwords=stopwords,
                          max_words=1000,
                          max_font_size=200,
                          random_state=43
                          ).generate(str(df))

print(wordcloud)
fig = plt.figure(1)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
fig.savefig("D:/bluebird.png", dpi=900)
```

Lampiran 22. *Output Social Network Analysis* BlueBird Taxi

ID	*Username*	*Degree Centrality*	*Closeness Centrality*	*Betweenness Centrality*	*Eigenvector Centrality*
1	_____ems	1	1	0	0
2	__aih	0	0	0	0
3	__hyee13	0	0	0	0
4	__lattepapi	1	0	0	0.001533
5	__lisna	0	0	0	0
.	.	.	.	.	.
.	.	.	.	.	.
4619	ZoneWarPuji	4	0.8	4	0.001533
4620	Zsinarf	1	1	0	0
4621	Zuhaald	0	0	0	0
4622	Zulhidayat_AY	0	0	0	0
4623	zzy_my	0	0	0	0

Lampiran 23. *Output Social Network Analysis* GoCar

ID	*Username*	*Degree Centrality*	*Closeness Centrality*	*Betweenness Centrality*	*Eigenvector Centrality*
1	___anka	0	0	0	0
2	___rahayu	1	1	0	0
3	__aih	1	0	0	0.001079
4	__danar	0	0	0	0
5	__gpp	1	0	0	0.001079
.	.	.	.	.	.
.	.	.	.	.	.
9640	zvlfq	0	0	0	0
9641	zwolf_me	1	1	0	0
9642	zwolfuhr	0	0	0	0
9643	zxxxxaad	0	0	0	0
9644	ZyaUlhuda	1	0	0	0.001079

Lampiran 24. *Output Social Network Analysis* GrabCar

ID	*Username*	*Degree Centrality*	*Closeness Centrality*	*Betweenness Centrality*	*Eigenvector Centrality*
1	_____honeybuhni	0	0	0	0
2	___Maaryam	0	0	0	0
3	___rahayu	1	1	0	0
4	___ridwnn	1	0	0	0.001579
5	__aih	1	0.666667	0	0
.	.	.	.	.	.
.	.	.	.	.	.
5824	zyd_zbr	1	1	0	0
5825	zyjmeanie	1	0.490066	0	0
5826	zz_aa_uu	0	0	0	0
5827	zzellano	0	0	0	0
5828	zzfrx	0	0	0	0

Lampiran 25. *Output Social Network Analysis* Taksi Konvensional dan Taksi Berbasis *Online*

ID	*Username*	*Degree Centrality*	*Closeness Centrality*	*Betweenness Centrality*	*Eigenvector Centrality*
1	_____ems	2	0.409174	0	0.038638
2	_____honeybuhni	0	0	0	0
3	___anka	0	0	0	0
4	___Maaryam	0	0	0	0
5	___rahayu	1	1	0	0
.	.	.	.	.	.
.	.	.	.	.	.
18197	zyjmeanie	1	0.309942	0	0
18198	zz_aa_uu	0	0	0	0
18199	zzellano	0	0	0	0
18200	zzfrx	0	0	0	0
18201	zzy_my	0	0	0	0

Lampiran 26. Surat Pernyataan Data

SURAT PERNYATAAN

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

Nama : Putri Ayu Sekar Karimah

NRP : 062116 4500 0009

menyatakan bahwa data yang digunakan dalam Tugas Akhir / ~~Thesis~~ 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 *keywords* "BlueBird Taxi", "GoCar", dan "GrabCar"

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

Mengetahui
Pembimbing Tugas Akhir

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

Surabaya, 23 Juli 2018

Putri Ayu Sekar Karimah
NRP. 062116 4500 0009

*(coret yang tidak perlu)

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Penulis bernama lengkap Putri Ayu Sekar Karimah dilahirkan di Ngawi, 17 Februari 1995 sebagai anak ketiga dari tiga bersaudara. Penulis bertempat tinggal di Ngawi dan telah menempuh pendidikan formal dimulai dari TK Dharma Wanita dan TK Bhayangkari, SDN Margomulyo 3 Ngawi, SMP Negeri 2 Ngawi, dan SMA Negeri 2 Ngawi.Setelah lulus SMA, penulis melanjutkan studi dengan menempuh pendidikan di Institut Teknologi Sepuluh Nopember (ITS) Surabaya di Program Studi Diploma III dan melanjutkan ke Jenjang Sarjana di Departemen Statistika ITS melalui jalur regular. Selama perkuliahan penulis berperan aktif dalam mengikuti organisasi dan kegiatan-kegiatan kemahasiswaan yang ada di ITS untuk menunjang *softskill* diluar kemampuan akademik. Tahun pertama penulis mendapatkan keluarga baru yaitu ∑24 dengan *tagline* "LEGENDARY" dengan nomor sigma ∑24.151. Selain itu, penulis bergabung dengan dua unit kegiatan mahasiswa yaitu ITS Badminton Community (IBC) dan UKAFO (Fotografi). Tahun kedua, pernah bergabung dalam organisasi kemahasiswaan, yaitu sebagai staff departemen Hubungan Luar HIMADATA-ITS periode 2014/2015 dan staff departmen Hubungan Luar UKM-IBC periode 2014/2015. Tahun ketiga, penulis bergabung dalam HIMADATA-ITS sebagai Ketua Biro Media Informasi HUBLU periode 2015/2016. Tahun keempat dan kelima penulis memfokuskan diri untuk meningkatkan kemampuan pada bidang akademik. Selama masa perkuliahan, penulis mendapatkan kesempatan Kerja Praktek di Badan Pusat Statistik (BPS) Kabupaten Ngawi dan PT Nutrifood Indonesia *Plant* Ciawi. Akhir kata, apabila ada kritik dan saran atau diskusi yang berhubungan mengenai Tugas Akhir dapat dikirim melalui email penulis putriayusekarkarimah@gmail.com atau dapat menghubungi melalui nomor +62 813 5763 8550.

(Halaman ini sengaja dikosongkan)