ANALISIS SENTIMEN TERHADAP OPINI PUBLIK TENTANG KESEHATAN MENTAL SELAMA PANDEMI COVID-19 DI MEDIA SOSIAL *TWITTER* MENGGUNAKAN *NAIVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE*

(Studi Kasus : Data Opini Twitter Tentang Kesehatan Mental Selama Pandemi COVID-19 Tahun 2020 di Indonesia)

TUGAS AKHIR

Deinda Afiya Pangestu
15611151

JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2020

HALAMAN PERSETUJUAN PEMBIMBING

TUGAS AKHIR

HALAMAN PERSETUJUAN PEMBIMBING

TUGAS AKHIR

Judul : Analisis Sentimen Opini Publik tentang Kesehatan Mental Selama Pandemi Covid-19 di Media Sosial Twitter Menggunakan Naive Bayes Classifier dan Support Vector Machine
(Studi Kasus : Data Opini Twitter Tentang Kesehatan Mental Selama Pandemi COVID-19 Tahun 2020 di Indonesia)

Nama Mahasiswa : Deinda Afiya Pangestu

Nomor Mahasiswa : 15611151

TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK DIUJIKAN

Yogyakarta,

Pembimbing

(Muhammad Hasan Sidiq K, S.Si., M.Sc.)

HALAMAN PENGESAHAN

TUGAS AKHIR

ANALISIS SENTIMEN OPINI PUBLIK TENTANG DEPRESI DI MEDIA SOSIAL *TWITTER* MENGGUNAKAN *NAIVE BAYES CLASSIFIER* DAN *SUPPORT VECTOR MACHINE*

(Studi Kasus : Data Opini Twitter Tentang Kesehatan Mental Selama Pandemi COVID-19 Tahun 2020 di Indonesia)

Nama Mahasiswa : Deinda Afiya Pangestu

NIM : 15611151

TUGAS AKHIR INI TELAH DIUJIKAN
PADA TANGGAL: Senin, 22 Juni 2020

Nama Penguji:	**Tanda Tangan**
1. (Muhammad Muhajir, S.Si.,M.Sc)	
2. (Rahmadi Yotenka, S.Si.,M.Sc)	.....
3. (Muhammad Hasan Sidiq K, S.Si., M.Sc.)	.....

Mengetahui,
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

(Prof. Riyanto, S.Pd., M.Si., Ph.D.)

KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakaatu

Alhamdulillaahirabbil'aalamiin, Puji syukur atas kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis diberikan keimanan, kekuatan, kesehatan, kesabaran, kelancaran, serta keselamatan selama penyusunan tugas akhir ini hingga dapat terselesaikan. Shalawat serta salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad SAW semoga mendapatkan safaatnya diakhir hayat nanti. Tugas Akhir yang berjudul Analisis Sentimen Opini Publik tentang Kesehatan Mental Selama Pandemi Covid-19 di Media Sosial Twitter Menggunakan Naive Bayes Classifier dan Support Vector Machine (Studi Kasus : Data Opini Twitter Tentang Kesehatan Mental Selama Pandemi COVID-19 Tahun 2020 di Indonesia) disusun sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang strata satu di Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.

Penyelesaian tugas akhir ini tidak terlepas dari dukungan bantuan dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan kali ini penulis bermaksud untuk menyampaikan ucapan terima kasih kepada :

1. Bapak Dr.Edy Widodo,S.Si., M.Si. selaku ketua Jurusan Statistika beserta seluruh jajarannya.
2. Bapak Muhammad Hasan Sidiq K, S.Si., M.Sc. selaku dosen pembimbing Tugas Akhir yang telah membimbing dan memberikan arahan kepada penulis selama penyusunan Tugas Akhir ini.
3. Seluruh Dosen, dan Staff Administrasi yang telah banyak membantu dan memberikan kemudahan, terima kasih atas ilmu yang diberikan dan bimbingannya.
4. Mamas (Alm) Asep Kurnia Ali terima kasih karena telah menjadi kakak yang baik dan sabar walaupun sering dijahili dan berantem terus. Semoga tenang di alam sana.

5. Semua pihak yang tidak tersebut yang turut membantu penulis dalam menyelesaikan tugas akhir ini, penulis mengucapkan terima kasih semoga Allah SWT selalu memberikan rahmat dan anugerah-Nya kepada kita semua.

Demikianlah yang dapat penulis sampaikan, semoga tugas akhir ini dapat memberikan manfaat baik bagi penulis maupun bagi semua pihak yang membutuhkan. Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan dan jauh dari kesempurnaan karena keterbatasan pengetahuan yang dimiliki penulis. Oleh karena itu, kritik dan saran yang membangun akan sangat penulis harapkan demi kesempurnaan tugas akhir ini. Akhir kata, semoga Allah SWT senantiasa melimpahkan rahmat serta hidayah-Nya kepada kita semua, Aamiin aamiin ya robbal ‘alamiin.

Wassalamu’alaikum Warahmatullaahi Wabarakaatu

Yogyakarta, 22 Juni 2020

Deinda Afiya Pangestu

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam tugas akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk tugas akhir. Tugas akhir ini diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 22 Juni 2020

Deinda Afiya Pangestu

ANALISIS SENTIMEN OPINI PUBLIK TENTANG KESEHATAN MENTAL SELAMA PANDEMI COVID-19 *TWITTER* MENGGUNAKAN *NAIVE BAYES CLASSIFIER*DAN *SUPPORT VECTOR MACHINE*

(Studi Kasus : Data Opini Twitter Tentang Kesehatan Mental Selama Pandemi COVID-19 Tahun 2020 di Indonesia)

Deinda Afiya Pangestu

Program Studi Statistika Fakultas MIPA

Universitas Islam Indonesia

Sejak kasus pertama terdeteksi di Wuhan bulan Desember tahun lalu, pandemi Covid-19 masih belum pasti kapan akan berakhir. Hal tersebut membuat sebagian besar orang menjadi cemas, bukan hanya karena takut terinfeksi virus, melainkan juga karena berbagai pemicu kecemasan lain yang dihadapi dalam waktu yang bersamaan. Pandemi COVID-19 merupakan sebuah krisis global yang bukan hanya mengancam kesehatan masyarakat secara fisik,namun juga secara mental. Beberapa masyarakat berupaya untuk menanggulangi dampak psikis akibat pandemi Covid-19 adalah dengan sekedar menyalurkan keluh kesahnya atau pun saling berbgai informasi melalui media sosial salah satunya Twitter. Berdasarkan cuitan pada media sosial Twitter peneliti ingin mencari bagaimana pendapat msyarakat terutama di Indonesia tentang kesehatan mental selama pandemi. Apa faktor-faktor yang sering dikeluhkan pengguna Twitter mengenai kesehatan mentalnya di Twitter dan apa saja yang dirasakan. Dalam skripsi ini, peneliti fokus pada komentar-komentar terkait kesehatan mental selama pandemi Covid-19. Pada penelitian ini analisis sentimen dilakukan untuk melihat apa saja opini pengguna Twitter mengenai kesehatan mental. Pengklasifikasian data *tweet* menggunakan algoritma *naïve bayes classifier* (NBC) dan *support vector machine* (SVM). Didapatkan hasil akurasi sebesar 80,81% untuk metode SVM dengan kernel Polinomial. Dan untuk hasil akurasi yang lain yaitu metode SVM dengan kernel RBF sebesar 78,79%, SVM dengan kernel Linier sebesar 71,73% dan NBC sebesar 70,71%.

Kata Kunci: Analisis Sentimen, *Text Mining*, Naive Bayes, SVM, Twitter

SENTIMENT ANALYSIS OF PUBLIC OPINION ON MENTAL HEALTH DURING COVID-19 PANDEMIC IN SOCIAL MEDIA TWITTERS USING NAIVE BAYES CLASSIFIER AND SUPPORT VECTOR MACHINE

Deinda Afiya Pangestu

Department of Statistics, Faculty of Mathematics and Natural Science

Universitas Islam Indonesia

Since the first cases were detected in Wuhan in December last year, the Covid-19 pandemic is still uncertain when it will end. This makes most people anxious, not only because they are afraid of being infected with the virus, but also because of the various other anxiety triggers they are facing at the same time. The COVID-19 pandemic is a global crisis that not only threatens public health physically, but also mentally. Some people are trying to overcome the psychological impact of the Covid-19 pandemic by simply channeling their complaints or sharing information through social media, one of which is Twitter. Based on a tweet on social media Twitter, the researcher wanted to find out what the public, especially in Indonesia, thought about mental health during the pandemic. What are the factors that Twitter users often complain about their mental health on Twitter and how they feel. In this thesis, researchers focus on comments related to mental health during the Covid-19 pandemic. In this study, sentiment analysis was carried out to see what Twitter user opinions about mental health. The classification of tweet data uses the naïve Bayes classifier (NBC) and support vector machine (SVM) algorithm. The results obtained are 80.81% accuracy for the SVM method with the Polynomial kernel. And for other accuracy results, namely SVM with an RBF kernel of 78.79%, SVM with a Linear kernel of 71.73% and NBC of 70.71%.

Keyword: Sentiment Analysis, Text Mining, Naive Bayes, SVM, Twitter

BAB I
PENDAHULUAN

1.1. Latar Belakang

Sejak kasus pertama terdeteksi di kota Wuhan, China bulan Desember tahun lalu pada tahun 2019 pandemi Covid-19 masih belum diketahui kapan akan berakhir. Data dari John Hopkins University menyebutkan bahwa hingga kini virus tersebut telah menginfeksi hampir 3,5 juta manusia dengan 240 ribu lebih diantaranya meninggal dunia. (Johns Hopkins University, 2020).

Hal tersebut membuat sebagian besar orang menjadi cemas, bukan hanya karena takut terinfeksi virus, melainkan juga karena berbagai pemicu kecemasan lain yang dihadapi dalam waktu yang bersamaan seperti panic buying di beberapa darerah yang sempat terjadi pada masa awal pandemi memasuki Indoesia, harga kebutuhan pokok yang meninggi, PHK pada sejumlah tepat kerja, PSBB yang sempat dilaksanakan di beberapa daerah, dan lainnya. Pandemi COVID-19 merupakan sebuah krisis global yang bukan hanya mengancam kesehatan masyarakat secara fisik,namun juga secara mental. Permasalahan psikologis tersebut dapat menimpa siapa saja, baik petugas medis, korban yang terinfeksi, keluarga korban, bahkan masyarakat secara umum (Kang, dkk, 2020). Keadaan tersebut dapat diperparah dengan pemberitaan yang kerap muncul baik di media elektronik maupun daring. Berita palsu (hoaks) yang sering muncul di media sosial dapat menimbulkan kecemasan, kebencian dan bahkan rasisme (Kadam & Atre, 2020). Beberapa masyarakat berupaya untuk menanggulangi dampak psikis akibat pandemi Covid-19 adalah dengan sekedar menyalurkan keluh kesahnya atau pun saling berbgai informasi melalui media sosial salah satunya Twitter.

Twitter merupakan jejaring sosial dan layanan komunikasi *real-time* yang dirilis pertama kali pada tahun 2006 dan hingga kini telah digunakan oleh jutaan orang dan organisasi sebagai media untuk mengakses dan membagikan informasi

secara real-time. Survei yang dilakukan oleh Asosiasi Penyelenggara Jaringan Internet Indonesia (APJII) sepanjang tahun 2018 mengungkapkan bahwa Twitterdi Indonesia menempati urutan keempat sebagai media sosial yang paling banyak memiliki pengguna aktif yaitu sebanyak 6,43 juta pengguna aktif. Pada kuartal pertama tahun 2018, pengguna Twittertumbuh 11 persen. Pada kuartal kedua, pengguna Twitter Indonesia tumbuh 31 persen. Pada kuartal ketiga, pengguna Twitter Indonesia tumbuh 33 persen. Puncaknya pada kuartal keempat, TwitterIndonesia mencatat pertumbuhan pengguna aktif harian sebesar 41 persen.

Inti dari Twitter adalah *tweet.Tweet* adalah tulisan yang panjangnya 140 karakter pada awal berdirinya Twitter. Tahun 2017 Twittermenambah jumlah karakter menjadi 280 karakter dalam satu*tweet.* Kata *tweet*merujuk sebagai kata benda, misalnya dalam kalimat "Apakah kamu sudah membaca*tweet* ini?" dan juga merujuk sebagai kata kerja, seperti dalam kalimat "Silahkan *tweet* ini". Pada awalnya Twitterdimaksudkan sebagai fasilitas untuk menjawab pertanyaan, walaupun seiring berjalannya waktu sebagian orang meng-*update*tentang kegiatan yang sedang dilakukan, berita baru ataupun menjawab pertanyaan-pertanyaan dari para pengguna Twitter lainnya. (Zarella, 2011)

Kelebihan Twitter dibanding media sosial lainnya yaitu membantu penyebaran informasi secara lebih cepat yang kemudian akan menjadi sebuah topik yang dibahas oleh para penggunanya. Hal ini memudahkan masyarakat mendapatkan informasi secara *real time* dan *up to date* karena berita yang di-*update* setiap saat oleh. Twitter juga memudahkan menghubungkan dengan pengguna lain tanpa harus mengikutisatu sama lain ketika ingin mengetahui nama akun yang di-*mention.*

Berdasarkan cuitan pada media sosial Twitter peneliti ingin mencari bagaimana pendapat msyarakat terutama di Indonesia tentang kesehatan mental selama pandemi. Apa faktor-faktor yang sering dikeluhkan pengguna Twitter mengenai kesehatan mentalnya di Twitter dan apa saja yang dirasakan. Dalam skripsi ini, peneliti fokus pada komentar-komentar terkait kesehatan mental selama pandemi Covid-19. Analisis sentimen atau *opinion mining* merupakan proses

memahami, mengekstrak dan mengolah data tekstual secara otomatis utnuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Pada penelitian ini analisis sentimen dilakukan untuk melihat apa saja opini pengguna Twittermengenai kesehatan mental. Pengklasifikasian data *tweet* menggunakan algoritma *naïve bayes classifier* (NBC) dan *support vector machine* (SVM).

1.2. Rumusan Masalah

Berdasarkan latar belakang, maka permasalahan yang akan dikaji dalam penelitian ini adalah sebagai berikut :

1. Bagaimana hasil penerapan metode sentimen analisis terkait *tweet*mengenai kesehatan mental selama pandemi Covid-19.
2. Berapa tingkat akurasi yang didapatkan dari hasil klasifikasi menggunakan algoritma *Naive Bayes Classifier* dan *Support Vector Machine?*

1.3. Batasan Masalah

Untuk menjaga fokus dalam penelitian maka beberapa batasan yang diberikan dalam penelitian adalah :

1. Penelitian ini menggukan media sosial Twitter dengan *library API Developer* untuk mengekstrak *Log* Twitter.
2. Penelitian hanya berfokus pada komentar berbahasa Indonesia.
3. Penelitian ini hanya berfokus pada komentar atau *text* yang berhubungan tentang kesehatan mental selama pandemi Covid-19 pada pengguna Twitter di Indonesia.
4. Metode yang digunakan untuk mengklasifikasikan adalah *Naive Bayes Classifier* dan *Support Vector Machine.*

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Mengetahui hasil penerapan metode sentimen analisis terkait *tweet*mengenai kesehatan mental selama pandemi Covid-19.
2. Mengetahui nilai akurasi yang didapatkan dari hasil klasifikasi menggunakan *Naive Bayes Classifier* dan *Support Vector Machine.*

1.5. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat kepada pihak-pihak yang terkait. Adapun manfaat yang diharapkan antara lain :

1. Dengan diketahuinya perbandingan kinerja metode *Naive Bayes Classifier* dan *Support Vector Machine (SVM)* dalam melakukan klasifikasi maka dapat diketahui metode mana yang terbaik untuk melakukan klasifikasi berdasarkan data media sosial Twitter.
2. Mengetahui gambaran masyarakat mengenai kesehatan mental selama pandemi Covid-19.
3. Menjadi referensi bagi penelitian berikutnya yang relevan mengenai media sosial Twitter.

BAB II
TINJAUAN PUSTAKA

Bab ini memuat penelitian terdahulu yang berguna sebagai bahan acuan penulis untuk melakukan penelitian ini. Beberapa penelitian yang pernah dilakukan sebelumnya terkait studi kasus yang penulis gunakan antara lain:

Penelitian menggunakan metode *Support Vector Machine* pernah dilakukan oleh Anita Novantirani,dkk pada tahun 2015 dengan judul "*Analisis Sentimen pada* Twitter *untuk Mengenai Pengguna Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine"* yangmembahas upaya mengurangi kemacetan pada transportasi umum darat dalam kota. Dibutuhkan opini dari pengguna Twitter untuk mengetahui penilaian pelayanan transportasi umum darat dalam kota apakah positif atau negatif, serta mengetahui faktor opini apa yang sering muncul. Dengan *dataset* yang digunakan berfokus pada opini berbahasa Indonesia tentang penggunaan transportasi umum darat khusus dalam kota dengan memilih 4 sampel kendaraan dengan perbandingan 80 : 20 persen yaitu angkot 272 kalimat opini, kopaja 184 kalimat opini, metro mini 264 kalimat opini, dan trasjakarta 418 kalimat opini. Dari hasil pengujian untuk kasus pada penelitian ini didaptakan bahwa SVM dapat diimplementasikan dengan nilai akurasi mecapai 78,12% pada transportasi transjakarta.

Penelitian dengan membandingkan metode *Support Vector Machine* dan *Naive Bayes Classifier* pernah dilakukan oleh Ghulam Asrofi Buntoro pada tahun 2016 dengan judul "*Analisis Sentimen Hatespeech pada* Twitter *dengan Metode Naive Bates Classifier dan Support Vector Machine*" yang membahas tentang *HateSpeech*. Dalam penelitian ini datayang digunakan sebanyak 522 *tweet* yang kemudian diklasifikasikan menjadi dua kelas yaitu *HateSpeech* dan *GoodSpeech* yang dalam *preprocessing* menggunakan dua *stopword* yaitu *stopword list* WEKA

dan *stopword list* bahasa Indonesia. Hasil yang didapatkan metode klasifikasi *Support Vector Machine* dengan nilai rata-rata akurasi mencapai 66,6%

Penelitian dengan membandingkan metode *Support Vector Machine* dan *Naive Bayes Classifier* pernah dilakukan oleh Faradhillah, Nuke Y. A,dkk padatahun 2016 dengan judul ”*Eksperimen Sistem Klasifikasi Analisa Sentimen* Twitter *pada Akun Resmi Pemerintah Kota Surabaya Berbasis Pembelajaran Mesin*”. Dalam penelitian ini datayang digunakan sebanyak 5836 *tweet* yang diketahui terdiri dari 7 atribut yaitu *id_message, text_message, date_message, id_kategori, account, lattitude* dan *longitude*. Dalam penelitian ini dilakukan pengklasifikasian opini masyarakat di media sosial Twittermenggunakan algoritma *Naive Bayes Classifier* dan *Support Vector Machine*. Kemudian didapatkan tiga klasifikasi yaitu positif, negatif dan netral. Model klasifikasi terbaik didapatkan menggunakan algoritma *Support Vector Machine* sebesar 79,81%.

Penelitian menggunakan metode *Naive Bayes Classifier* pernah dilakukan oleh Yonnathan Sari Mahardhika dan Eri Zuliarso pada tahun 2018 dengan judul ”*Analisis Sentimen Terhadap Pemerintahan Joko Widodo pada Media Sosial* Twitter *Menggunakan Algoritma Naive Bayes Classifier”* yang membahas tentang berbagai macam komentar publik dalam media sosial Twitter dimana objek pada penelitian ini adalah *tweet* tentang Pemerintahan Presiden Joko Widodo dengan jumlah data sebanyak 400 data *tweet* terdiri dari 300 data latih dan 100 data uji. Dan didapatkan hasil akurasi untuk analisis klasifikasi dengan metode Naive Bayes Classifier sebesar 97%.

Penelitian menggunakan metode *Support Vector Machine* pernah dilakukan oleh Nabila Safina pada tahun 2017 dengan judul “*Analisis Sentimen Pada* Twitter *Terhadap Jasa Transportasi Online di Indonesia dengan Metode Support Vector Machine”* yangmembahas tentang bagaimana data tweet dengan beberapa kata kunci indikasi depresi yang dihasilkan dari wawancara dengan beberapa pakar psikologi.Hasil dari penelitian ini mendapatkan data *tweet*sebanyak 1498 data *tweet* yang dibagi menjadi 1000 data train dan 498 untuk data test. Sehingga didapatkan hasil akurasi sebesar 73,86%.

Penelitian mengenai *Self Disclosure* pernah dilakukan oleh Ajeng Prima Dewi dan Santi Delliana pada tahun 2019 dengan judul *Self Disclosure* Generasi Z di Twitter

menggunakan metode *Naive Bayes Classifier* pernah dilakukan oleh Sigit Suryono,dkk pada tahun 2018 dengan judul ”*Klasifikasi Sentimen pada* Twitter *dengan Naive Bayes Classifier”* dimana objek pada penelitian ini adalah *tweet* tentang Presiden Joko Widodo dan pemerintahannya dengan jumlah data sebanyak 3458 data *tweet.* Setelah dilakukan tiga kali uji cobadidapat hasil pada uji coba pertama dengan pembagian data pelatihan 40% dan data pengujian 60% sebesar 64,95%, pada uji coba kedua dengan pembagian data pelatihan 50% dan data pengujian 50% sebesar 66,36%, dan pada uji coba ketiga dengan pembagian data pelatihan 60% dan data pengujian 40% sebesar 66,79%.

Penelitian menggunakan metode *Support Vector Machine*pernah dilakukan oleh Silvia Aprilia, Muhammad Tanzil Furqon, dan Mochammad Ali Fauzi pada tahun 2018 dengan judul ”*Klasifikasi Penyakit Skixofrenia dan Episode Depresi Pada Pasien Gangguan Kejiwaan Dengan Menggunakan Metode Support Vector Machine (SVM)”* dimana objek pada penelitian ini adalah data rekam mediktentang penyakit skixofrenia dan episode depresi pada pasien gangguan kejiwaan dengan jumlah data sebanyak 200 datadengan ratio perbandingan data train dan data test 80% : 20% menghasilkan tingkat akurasi sebesar 79%.

Penelitian menggunakan metode *Naive BayesClassifier*pernah dilakukan oleh Silvia Aprilia, Muhammad Tanzil Furqon, dan Mochammad Ali Fauzi pada tahun 2018 dengan judul ”*Analisis Sentimen untuk Mengukur Ingkat Indikasi Depresi pada Twitter Menggunakan Text Mining”* dimana objek pada penelitian ini adalah data rekam mediktentang penyakit skixofrenia dan episode depresi pada pasien gangguan kejiwaan dengan jumlah data sebanyak 200 datadengan ratio perbandingan data train dan data test 80% : 20% menghasilkan tingkat akurasi sebesar 79%.

Tabel 2.1 Penelitian Sebelumnya

Peneliti	**Judul Penelitian**	**Metode Penelitian**	**Hasil**
Anita Novantirani , dkk (2015)	Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode *Support Vector Machine*	*Support Vector Machine*	Pada penelitian ini menggunakan empat kategori transportasi yaitu : angkot, kopaja, metro mini, dan transjakarta. Sehingga pada akurasi angkot sebesar 70,95% dengan perbandingan data latih dan data uji 3:1 (4-fold), sedangkan untuk akurasi kopaja sebesar 65,76%, metro mini sebesar 71,96%, dan transjakarta sebesar 78,12% dengan perbandingan data latih dan data uji 1:1 (2-fold)
Ghulam Asrofi Buntoro (2016)	Analisis Sentimen *Hatespeech* pada Twitter dengan Metode *Naive Bates Classifier* dan *Support Vector Machine*	*Support Vector Machine* dan *Naive Bayes Classifier*	Dengan jumlah dataset sebanyak 522 tweet yang didistribusikan secara merata menjadi Dua sentimen HateSpeech dan GoodSpeech. Hasil akurasi tertinggi didapatkan saat menggunakan metode klasisifikasi Support Vector Machine (SVM) dengan nilai rata-rata akurasi mencapai 66,6% nilai presisi 67,1%, recall 66,7%, TP rate 66,7% dan TN rate 75,8%.
	Eksperimen Sistem Klasifikasi Analisa	*Support Vector*	Data yang didapatkan yaitu berupa data mentah yang

Peneliti	Judul Penelitian	Metode Penelitian	Hasil
Nuke Y. A. Faradhillah, dkk (2016)	Sentimen Twitter pada Akun Resmi Pemerintah Kota Surabaya Berbasis Pembelajaran Mesin	*Machine* dan *Naive Bayes Classifier*	berjumlah sebanyak 5836 data. Untuk pembagian data latih dan data uji dalam penelitian ini yaitu sebesar 70 : 30. Untuk nilai akurasi optimal dari metode Naive Bayes Classifier yaitu 78,77%. Sedangkan untuk nilai akurasi optimal dari metode Support Vector Machine yaitu 79,81% dengan kernel RBF.
Yonathan Sari Mahardhika dan Eri Zuliarso (2018)	Analisis Sentimen terhadap Pemerintahan Joko Widodo pada Media Sosial Twitter Menggunakan Algoritma *Naive Bayes Classifier*	*Naive Bayes Classifier*	Dengan jumlah data sebanyak 400 data *tweet* terdiri dari 300 data latih dan 100 data ujimemberikan prediksi kelas sentimen data berdasarkan *confussion matrix* dengan rincian 49 *tweet* sentimen negatif dan 51 *tweet* sentimen positif. Dan didapatkan hasil akurasi untuk analisis klasifikasi dengan metode Naive Bayes Classifier sebesar 97%.
Nabila Safina (2017)	Analisis Sentimen pada Twitter Terhadap Jasa Transportasi *Online* di Indonesia dengan	*Support Vector Machine*	Data dikumpulkan dari Twitter 1000 *tweet* dengan 900 *tweet* untuk data pelatihan dan 100 *tweet* untuk data pengujian. Pada data pelatihan menghasilkan 450 kalimat positif dan 450 kalimat

Peneliti	Judul Penelitian	Metode Penelitian	Hasil
	Metode *Support Vector Machine*		negatif dan pada data pengujian menghasilkan 50 kalimat positif dan 50 kalimaat negatif. Proses klasifikasi menggunakan metode *Support Vector Machine* (SVM) menunjukkan hasil akurasi sebesar 87%.
Sigit Suryono, dkk (2018)	Klasifikasi Sentimen pada Twitter dengan *Naive Bayes Classifier*	*Naive Bayes Classifier*	Didapatkan data *tweet* sebanyak 3485 yang kemudian dilakukan analisis dengan bahasa pemrograman Python. Dilakukan dari 3 skenario uji coba dengan uji coba pertama dengan perbandingan 4:6 menghasilkan tingkat akurasi sebesar 64,95%, uji coba kedua dengan perbandingan 5:5 menghasilkan tingkat akurasi sebesar 66,36%, uji coba ketiga dengan perbandingan 6:4 menghasilkan tingkat akurasi 66,79%.
Silvia Aprilia, Muhammad Tanzil Furqon, dan Mochamma	Klasifikasi Penyakit Skizofrenia dan Episode Depresi Pada Pasien Gangguan Kejiwaan Dengan Menggunakan	*Support Vector Machine*	Data yang dimiliki berupa data rekam medik pasien dengan penyakit skizofrenia hebefrenik dan depresi berat dengan masing-masing sebanyak 200 data dengan ratio perbandingan data train dan data test 80% : 20%

Peneliti	Judul Penelitian	Metode Penelitian	Hasil
d Ali Fauzi (2018)	Metode *Support Vector Machine (SVM)*		menghasilkan tingkat akurasi sebesar 79%.
Bella Nurfadhila (2019)	Analisis Sentimen untuk Mengukur Ingkat Indikasi Depresi pada Twitter Menggunakan *Text Mining*	*Naive Bayes Classifier*	Didapatkan data *tweet* sebanyak 1498 yang kemudian dilakukan analisis dengan bahasa pemrograman Python dengan data *train* sebanya 1000 data, dan data *test* sebanyak 498 data. Pada penelitian ini, tingkat akurasi yang dihasilkan sebesar 73,86%.
Ajeng Prima Dewi dan Santi Delliana (2019)	*Self Disclosure* Generasi Z di Twitter	Metode Kualitatif	Penelitian mengambil sampel remaja dengan rentang usia 19-22 tahun yang aktif bermain sosial media Twitter. Berdasarakan hasil penelitian diketahui bahwa informasn memilih untuk melakukan *self disclosure* karena di Twitter dapat menjadikannya lebih ekspresif dibandingkan dengan media sosial lainnya yang lebih mengutamakan aspek visual.

Setelah peneliti melakukan kajian terhadap beberapa penelitian sebelumnya, terdapat beberapa perbedaan dengan penelitian yang peneliti lakukan yaitu Anita Novantirani, dkk (2015), Nabila Safina (2017), dan Silvia Aprilia dkk (2018) masing-masing menggunakan satu metode *Support Vector Machine*& untuk

penelitian yang dilakukan oleh Sigit Suryono, dkk (2018), Yonathan Sari M. & Eri Zuliarso (2018), dan Bella Nurfadhila (2019) masing-masing hanya menggunakan metode *Naive Bayes Classifier*. Dan studi kasus yang dianalisis pada penelitian-penelitian tersebut tidak membahas tentang kesehatan mental.

Penelitian oleh Ghulan Asrofi B (2016) & Nuke Y. A. Faradhillah, dkk (2016) masing-masing membandingkan metode *Support Vector Machine* dan *Naive Bayes Classifier*. Pada penelitian Ghulan Asrofi B (2016) menggunakan 2 *stopword* yaitu *stopword list WEKA* dan *stopword list* bahasa Indonesia. Pada penelitian Nuke Y. A. Faradhillah, dkk (2016) menggunakan 3 klasifikasi.

BAB III
LANDASAN TEORI

3.1. COVID-19

Covid-19 merupakan penyakit yang diidentifikasikan penyebabnya adalah virus Corona yang menyerang saluran pernapasan. Penyakit ini pertama kali dideteksi kemunculannya di Wuhan, Tiongkok.Sebagaimana diketahui bahwa SARS-Cov-2 bukanlah jenis virus baru. Akan tetapi dalam penjelasan ilmiah suatu virus mampu bermutasi membentuk susunan genetik yang baru, singkatnya virus tersebut tetap satu jenis yang sama dan hanya berganti seragam. Alasan pemberian nama SARS-Cov-2 karena virus corona memiliki hubungan erat secara genetik dengan virus penyebab SARS dan MERS.

3.2. Kesehatan Mental

Kesehatan mental atau kesehatan jiwa merupakan aspek penting dalam mewujudkan kesehatan secara menyeluruh. Kesehatan mental yang baik memungkinkan orang untuk menyadari potensi mereka, mengatasi tekanan kehidupan yang normal, bekerja secara produktif dan berkontribusi pada komunitas mereka (WHO, 2017).

Kondisi mental pada tiap individu tidak dapat disamaratakan. Kondisi inilah yang semakin membuat urgensi pembahasan kesehatan mental yang mengarah pada bagaimana memberdayakan individu, keluarga, maupun komunitas untuk mampu menemukan, menjaga, dan mengoptimalkan kondisi sehat mentalnya dalam menghadapi kehidupan sehari-hari. Menurut WHO, gangguan kesehatan mental terdiri dari berbagai masalah, dengan berbagai gejala. Namun, umumnya dicirikan oleh beberapa kombinasi abnormal pada pikiran, emosi, perilaku dan hubungan dengan orang lain. Contoh dari gangguan kesehatan mental adalah depresi, cacat intelektual dan gangguan karena penyalagunaan narkoba, gangguan afektif bipolar, demensia, dan gangguan perkembangan termasuk autisme (WHO, 2017).

3.3. *Twitter*

Twitter pertama kali resmi diluncuran pada tanggal 13 Juli 2006oleh Jack Dorsey yang berbasis di San Brunomor, California. Twitter adalah sebuah situs *web* yang menawarkan jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirimkan dan membaca pesan yang disebut kicauan atau *tweet*, yang bebas mengekspresikan sesuatu seperti curhat/kritik (Lesmana, 2012). Pada awalnya pesan yang dikirim dan diterima di Twitter tidak lebih dari 140 karakter, namun pada tahun 2017 Twitter menambah jumlah karakternya. Dalam Twitterterdapat berbagai fitur antara lain :

1) Halaman Utama (*Home*) : Pada halaman utama kita bisa melihat kicauan yang dikirimkan oleh orang-orang yang sudah menjadi teman atau pengguna yang diikuti.
2) Profil : menampilkan profil atau data diri serta kicauan yang sudah pernah dikirim, *retweet* dan *likes*.
3) *Follow* : Fitur ini untuk menambah teman ke daftar pengguna yang diikuti.
4) *Unfollow* : Fitur ini merupakan kebalikan dari fitur *follow*, yaitu untuk membatalkan pertemanan atau berhenti mengikuti.
5) *Following* : *Following* adalah status di profil pengguna twitter lain yang menjelaskan bahwa sudah di-follow atau diikuti dan juga menunjukkan jumlah pengguna Twitter yang diikuti.
6) *Follower* : menunjukkan jumlah akun yang mengikuti.
7) *Dirrect Message* : *Dirrect Message* atau *DM* yaitu pesan pribadi antara kedua pengguna akun Twitter.
8) *Tweet* : *Tweet* adalah sebuah kicauan yang diunggah oleh pengguna Twitter.
9) *Hashtag* (Tagar) : *Hashtag* adalah sebuah penanda yang diberikan oleh pengguna Twitter unutk mengelompokkan sebuah topik untuk setiap Twitter.
10) *Trending Topic*: *Trending topic* mengacu pada hal yang sering dibicarakan orang yang berganti secara *real time*, sehingga *trending topic* tersebut

bersifat dinamis, tergantung seberapa banyak orang yang membicarakan topik tersebut.

11) *Reply* : *Reply* adalah membalas *tweet* dari pemilik akun Twitter lainnya. Untuk membalas *tweet* seseorang, dapat dilakukan dengan meletakkan simbol @ dan diikuti nama *user*nya.
12) *Retweet* : *Retweet* adalah cara agar kita dapat mengirim kembali *tweet* pengguna lain.
13) *Mention* : *Mention* digunakan untuk menandai orang dengan menggunakan simbol @.
14) *Like* : Fitur ini digunakan bila pengguna menyukai suatu *tweet* yang muncul pada halaman utama pengguna dengan cukup menekan ikon berbetuk hati.
15) *Capture Photos* and *Videos* : Fitur ini digunakan apabila pengguna ingin mengunggah foto atau video dengan cukup menggeser layar ke kiri.
16) Bisukan (*Mute*) : Fitur ini adalah tidak menampilkan unggahan dari pengguna yang dibisukan.

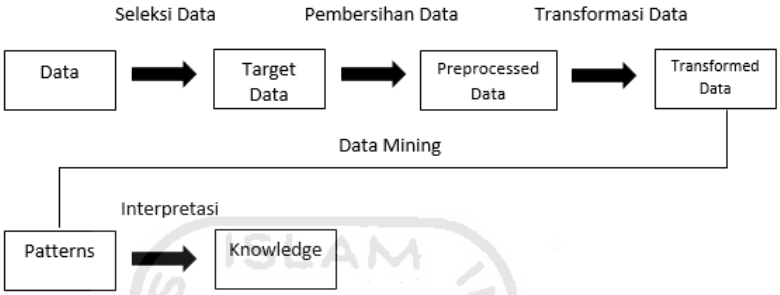
3.3. *Data Mining*

Data mining merupakan metode pengolahan data berskala besar untuk menemukan pola yang tersembunyi dari data tersebut. Hasil dari pengolahan data dengan metode data mining dapat digunakan untuk mengambil keputusan di masa depan. Data *mining* sering juga disebut *Knowledge Discovery in Database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Santosa, 2007). Dalam pelaksanaan aktifitas data *mining* sering kali digunakan berbagai teknik ataupun algoritma yang berasal dari berbagai disiplin ilmu misalnnya statistik, *artificial intelligence* ataupun *machine learning*.

Contoh sederhana penerapan *data mining* misalnya dalam mengelompokkan sebuah dokumen mengnai “Bunga” sesuai dengan konteks. Dalam hal ini konteks yang dimaksud seperti Bunga bank atau kredit, Bunga tanaman dan Bunga nama individu.

3.4.1. Tahapan *Data Mining*

Istilah *data mining* dan *Knowledge Discovery in Database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar (Fayyad,1996). Proses KDD itu ada 5 tahapan yang dilakukan secara terurut, yaitu :



Gambar 3. 1 Proses *Knowledge Discovery in Database* (KDD)

1. **Seleksi Data (*Data Selection*)**

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum taphap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. **Pembersihan data (*data cleaning*)**

Pembersihan data merupakan proses menghilangkan *noise* dan data yang inkonsisten atau data tidak relevan. Pada umumnya data yang diperoleh, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Data-data yang tidak relevan tersebut lebih baik dibuang. Pembersihan data juga akan mempengaruhi performasi dari teknik *data mining* karena data yang ditangani akan berkurang jumlah dan komplektasinya.

3. Transformasi data (*data transformation*)

Pada proses ini dilakukan transformasi bentuk data yang belum memiliki entitas yang jelas ke dalam bentuk data yang valid atau siap untuk dilakukan pada proses *data mining*.

4. Data mining

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. Interpretasi (evaluation)

Pada proses ini selanjutnya adalah menampilkan data tersebut dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Jadi, pola yang ditemukan akan diperiksa dan dicek apakah bertentangan dengan hipotesis sebelumnya atau tidak.

3.5. *Machine Learning*

Machine learning merupakan metode yang membuat sebuah mesin atau komputer dapat belajar dari pengalaman atau bagaimana cara memprogram mesin dapat belajar. *Machine learning* membutuhkan data untuk belajar sehingga biasa disebut *learn from data* (Alpaydin, 2010).Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel pada tahun 1959.

Contoh sederhana *machine learning* yaitu, *twitter* akan mengumpulkan data aktifitas penggunanya seperti orang yang sering diajak berkomunikasi, topik apa yang paling sering diikuti atau *base* (grup) yang sering dilihat. Data tersebut digunakan untuk menampilkan prioritas postingan atau iklan yang akan muncul pada akun pengguna *twitter*.

Secara garis besar ada 3 jenis metode belajar yang digunakan yaitu :

1. *Supervised Learning*

Supervised learning merupakan metode belajar terawasi. Program diberikan beberapa contoh data yang telah diketahui jenis/klasifikasinya sebagai bahan pembelajaran atau pelatihan. Terdapat kemungkinan program akan salah dalam mengklasifikasi sebuah objek setelah dilatih. Oleh karena itu, selain menggunakan *training set*, juga memberikan *test set*. Sehingga akan didapatkan persentase keberhasilannya (Harrington, 2012).

Contoh dari *Supervised learning* adalah pada pengaturan privasi di Twitter, pengguna dapat memasukkan kata apa saja yang berhubungan mengenai suatu hal tidak ditampilkan di beranda. Semisal pengguna memasukkan kata Kekeyi. Maka, *tweet* yang mengandung kataKekeyi tidak akan muncul di beranda pengguna.

2. *Unsupervised Learning*

Unsupervised learning merupakan metode belajar tidak terawasi. Metode ini menggunakan prosedur yang berusaha untuk mencari partisi dari sebuah pola. Berbeda dari *supervised learning*, metode ini tidak memiliki target *output* yang eksplisit sehingga dapat digunakan untuk kebutuhan pengelompokkan (Harrington, 2012).

Contoh dari *unsupervised learning* adalah semisal peneliti ingin mengelompokkan pengguna twitter yang menyukai K-Pop. Maka, peneliti dapat mengelompokkan berdasarkan *retweet* atau *likes*. Dan bisa juga mengelompokkannya berdasarkan profil pengguna twitter yang memakai foto artis Korea.

3. *Reinforcement Learning*

Reinforcement learning adalah sebuh metode learning yang mempelajari aturan kontrol dengan cara berinteraksi dengan lingkungan yang masih asing. Program akan mendapatkan hukuman jika salah dalam pengambilan keputusan dan hadiah jika benar. Pengalaman interaksi tersebut terakumulasi sehingga program dapat mengambil kesimpulan dikemudian hari menggunakan pola-pola yang telah dipelajarinya (Alpaydin, 2010).

Contoh dari *Reinforcement learning*adalah pada permainan Mario Bros. Tujuannya adalah mendapatkan skor tertinggi dan selamat sampai di istana. Ketika Mario berhasil mendapatkan koin & jamur maka akan mendapatkan poin. Akan tetapi jika Mario mengenai musuhnya maka poin & jumlah nyawa akan berkurang. Awalnya Mario hanya berusaha untuk mendapatkan koin. Lama kelamaan muncul musuh-musuhnya. Sehingga dari situlah Mario bisa menyimpulkan bahwa supaya menang harus mendapatkan koin & mengalahkan musuhnya.

3.6. *Crawling Data*

Crawling data di *twitter* adalah suatu proses untuk mengambil atau mengunduh data dari *servertwitter* dengan bantuan *Aplication Programming Interface* (API)twitter baik berupa data user maupun data tweet**.** *Crawling* data ini dilakukan untuk mengambil data dari *twitter* dimana data tersebut dibutuhkan untuk tugas akhir ini. Cara melakukan *crawling* data ialah dengan membuat program dengan memasukkan kata kunci untuk mencari *tweet* yang sesuai yang diinginkan.

3.7. *Text Mining*

Menurut Feldman dan Sanger (Feldman dan Sanger, 2007), *text mining* dapat didefinisikan secara luas sebagai proses pengetahuan yang memungkinkan pengguna berinteraksi dengan koleksi dokumen dari waktu ke waktu menggunakan berbagai macam analisis. *Text mining* bertujuan untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisis keterhubungan antar dokumen. Pada dasarnya proses kerja dari *text mining* banyak mengadopsi penelitian *data mining*, namun yang menjadi perbedaan adalah pola yang digunakan oleh *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur (Han & Kamber, 2006)

3.7.1 *Text Preprocessing*

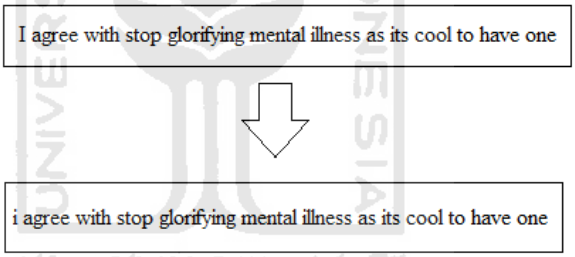
Dalam melakukan *text mining*, teks dokumen yang digunakan harus dipersiapkan terlebih dahulu, setelah itu baru dapat digunakan untuk proses utama. *Text preprocessing* merupakan tahapan awal dari *text mining*yaitu mengubah informasi dari tiap-tiap sumber data ke dalam bentuk atau format yang baku.Pada

text mining, data mentah yang berisi informasi merupakan data yang tidak terstruktur, sehingga diperlukan proses pengubahan bentuk menjadi data yangterstruktur sesuai kebutuhan, yaitu biasanya akan mejadi nilai-nilai numerik(Triawati, 2009).

Secara umum proses yang dilakukan dalam tahapan *preprocessing* adalah sebagai berikut :

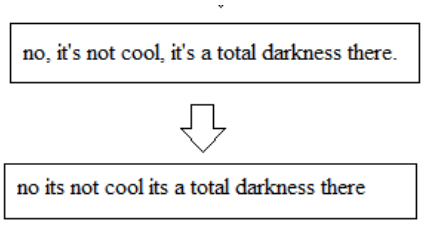
1. ***Case Folding***

   Case folding adalah proses penyamaan *case* dalam sebuah dokumen. Hal ini dilakukan untuk mempermudah pencarian. Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu peran *case folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (dalam hal ini huruf kecil atau *lowercase*).



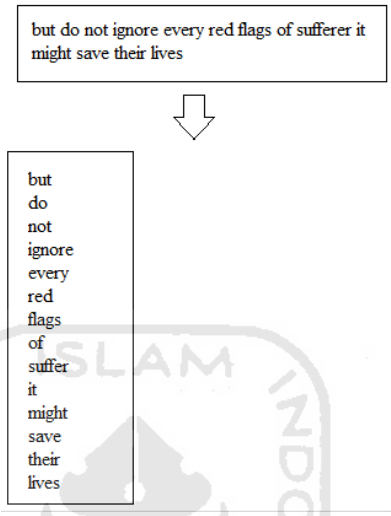
2. ***Cleaning***

   Merupakan suatu proses pembersihan kata pada dokumen yang berfungsi membuang beberapa karakter tertentu yang dianggap sebagai tanda baca, *mention*, RT, *hashtag*, dan lainnya yang kurang penting untuk mengurangi *noise.s*



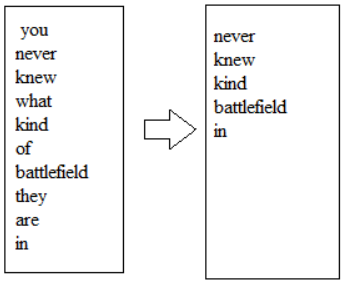
3. *Tokenizing.*

Tokenizing adalah proses pemotongan sebuah dokumen menjadi bagian-bagian yang dapat berupa kalimat, bab dan kata. Pada proses ini juga akan dilakukan penghilangan *whitespace.*.



4. *Stopword Removal*

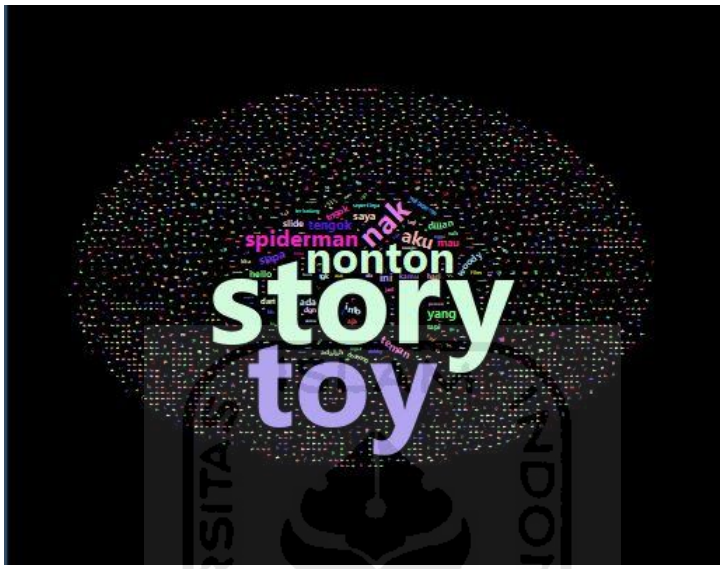
Stopword removal adalah proses penghilangan kata-kata yang tidak berkontribusi banyak pada isi dokumen sesuai dengan input yang dimasukkan. Kata-kata yang termasuk ke dalam *stopword* dihilangkan karena memberikan pengaruh yang tidak baik dalam proses *text mining* seperti kata-kata *"then","so","i","you","and"* dan lain-lain.



3.8. *Wordcloud*

Word cloud adalah salah satu hasil dari metode *text mining*, yang menampilkan kata-kata populer terkait dengan kata kunci internet dan data teks.

Semakin sering kata muncul dalam teks yang dianalisis, semakin besar ukuran kata muncul dalam gambar yang dihasilkan. Biasanya *wordcloud* memplotkan frekuensi kata dilihat dari ukuran katanya. *Wordcloud* sering digunakan untuk menyoroti istilah populer atau tren berdasarkan frekuensi penggunaan kata.



Gambar 3.2. Tampilan*Word Cloud*
(Sumber : Setiabudi, 2015)

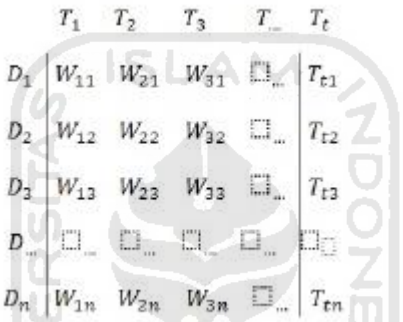
3.9. Asosiasi Kata

Asosiasi kata adalah mencari sebuah nilai hubungan suatu kata. Nilai asosiasi dihitung berdasarkan kata-kata yang sering muncul dan dianggap penting oleh peneliti. Hasil asosiasi teks menunjukkan besarnya nilai asosiasi antar kata dan seberapa sering kata-kata tersebut muncul bersamaan dalam satu kalimat. Semakin sering kata tersebut muncul bersamaan dalam satu kalimat maka semakin besar pula nilai asosiasi. Ada beberapa kategori nilai asosiasi menurut (Jonathan, 2006) yang digunakan sebagai berikut :

0 : Tidak ada asosiasi antara dua variabel
>0 - 0,25 : Asosiasi lemah
>0,25 - 0,5 : Asosiasi cukup
>0,5 – 0,75 : Asosiasi kuat
1 : Asosiasi sangat kua

3.10. *Document Term Matrix* dan Pembobotan Kata *Term Frequency – Inverse Document Frequency* (TF-IDF)

Sebelum melakukan pembobotan maka akan dilakukan tahapn pencarian text processing yaitu *case folding*, *cleaning*, *tokenizing* dan *stopword removal*, lalu selanjutnya dilakukan proses menghitung DTM dan pembobot TF–IDF. DTM atau *Document Term Matrix* merupakan sebuah koleksi dokumen yang direpresentasikan sebagai sebuah matriks. Setiap bagian dalam matriks bersesuaian dengan bobot yang diberikan dari suatu term dalam dokumen yang ditentukan. Nilai 1 akan diberikan pada suatu term apabila terdapat pada suatu dokumen dan nilai 0 akan diberikan apabila suatu term tersebut tidak ada dalam dokumen.



Gambar 3.3. Tampilan *Document Term-Matrix*

Sedangkan TF-IDF merupakan metode yang digunakan untuk menentukan nilai fekuensi sebuah kata di dalam sebuah dokumen atau artikel dan juga frekuensi di dalam banyak dokumen. Perhitungan TF-IDF ini menentukan seberapa relevan sebuah kata di dalam sebuah dokumen (Evan,Pranowo, & Purnomo 2014). Algoritma TF-IDF melakukan pemberian bobot pada setiap kata kunci disetiap kategori utuk mencari kemiripan kata kunci dengan kategori yang tersedia.

Dengan kata lain, pembobotan *term* dilakukan untuk mengukur tingkat similaritas antar dokumen dilakukan dengan membandingkan suatu *keyword* dengan dokumen yang sudah dibuat sebelumnya di *database*. Agar hasil pengukuran tingkat similaritas dokumen dengan *keyword* mendapatkan hasil yang optimal maka digunakanlah algoritma TF-IDF.

Untuk melakukan penghitungan besar nilai bobot menggunakan TF-IDF, beberapa langkah yang dilakukan diantaranya adalah sebagai berikut :

1. ***Term frequency (TF)***

Term frequency (TF) adalah frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan. Semakin besar jumlah kemunculan suatu *term* (TF tinggi) dalam dokumen, maka semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar.

Perhitungan *Term Frequency* (TF)

$$tf_{i,j} = \frac{n_{i,j}}{\sum n_{i,j}} \quad (3.1)$$

Keterangan :

$tf_{i,j}$ = Frekuensi *term*

$n_{i,j}$ = Banyaknya kata *i* dalam dokumen *j*

2. ***Inverse Document Frequency (IDF)***

IDF (*Inverse Document Frequency*) merupakan. Ukuran kemampuan kata untuk membedakan kategori. IDF dalam sebuah kata dapat diperoleh dari jumlah total dokumen yang terdapat kata tersebut dibagi oleh jumlah total dokumen setelah hasil bagi logaritmik. Semakin sedikit jumlah dokumen yang mengandung *term* yang dimaksud, maka nilai IDF semakin besar.

Perhitungan *Document Frequency* (IDF)

$$idf = log \frac{N}{df_1} \quad (3.2)$$

Keterangan :

idf = *Inverse Document Frequency*

N = Jumlah semua dokumen dalam koleksi

df_i = Jumlah dokumen yang mengandung *term (fi)*

3. ***Term frequency - Inverse Document Frequency* (TF-IDF)**

Term-Frequency – Inverse Document Frequency (TF-IDF) merupakan algoritma pembobotan tersusun dari dua nilai yang berasal dari dua algoritma

dengan pembobotan yang berbeda, yaitu nilai*Term Frequency (*TF*)*dikalikan dengan nilai *Inverse Document Frequency (*IDF*).*

Dengan demikian rumus umum untuk TF-IDF adalah penggabungan dari formula persamaan formula TF dan IDF dengan cara mengalikan *nilai term frequency* (TF) dengan nilai *inverse document frequency* (IDF) :

Perhitungan bobot TF-IDF

$$W_{i,j} = tf_{i,j} x\ idf_i$$

$$W_{ij} = tf_{ij} \times \log \frac{D}{df_j} \quad (3.3)$$

Keterangan :

$W_{i,j}$ = Bobot TF-IDF

idf_i = *Inverse Document Frequency*

$tf_{i,j}$ = Frekuensi suatu kata

Berikut adalah contoh perhitungan bobot dokumen terhadap kata/*queryterms* (Q) yang diberikan, dengan menggunakan metode pembobotan TF-IDF. Contoh *query* yang digunakan adalah "rebahan", "hiburan", dan "informasi".

Misal terdapat tiga buah dokumen yaitu :

Dokumen 1 (d_1) = Hiburan malam Minggu buka Twitter terus rebahan.

Dokumen 2 (d_2) = Main Twitter sambil rebahan udah paling *update* informasi

Dokumen 3 (d_3) = Twitter selalu menjadi pusat informasi paling *update*.

Dokumen tersebut kemudian melewati tahap *preprocessing*, maka kalimatnya mengalami perubahan sebagai berikut :

Dokumen 1 (d_1) = hiburan malam minggu Twitter rebahan.

Dokumen 2 (d_2) = main Twitter rebahan *update* informasi

Dokumen 3 (d_3) = twitter pusat informasi *update*.

Sehingga didapatkan *term* (*documents term*) dari ketiga dokumen tersebut yaitu :

- hiburan	- rebahan	- pusat
- malam	- main	- informasi
- minggu	- *update*	- Twitter

Pada tahap ini tiap dokumen dijadikan sebuah vektor dengan elemn sebanyak *term query* yang terdapat dalam tiap dokumen yang berhasil dikenali dari tahap ekstraksi dokumen sebelumnya. Vektor tersebut beranggotakan bobot dari setiap *term query* yang dihitung berdasarkan metode TF-IDF.

Metode TF-IDF berfugsi untuk mencari representasi nilai dari setiap dokumen. Vektor antara dokumen dan *query* yang terbentuk ditentukan oleh nilai bobot *term qery* dalam dokumen. Semakin besar nilai bobot yang dieroleh berarti tingkat similaritas dokumen terhadap *query* juga semakin tinggi. Contohnya untuk menghitung w_{ij} *term query* kata "rebahan" dalam Dokumen 1 (d_1) dapat diketahui sebagai berikut :

- Jumlah kemunculan kata "rebahan" dalam Dokumen 1 (d_1) sebanyak dua kali sehingga $tf_{ij} = tf_{rebahan} = \frac{1}{5} = 0{,}2$
- Total seluruh dokumen yang ada yaitu sebanyak tiga dokumen sehingga D = 3
- Dari ketiga dokumen tersebut, terdapat dua dokumen yang memuat kata "rebahan" sehingga $df_j = df_{rebahan} = 2$

Oleh karena itu, perhitungan nilai bobot *term* "rebahan" pada dokumen 1 (d_1) menggunakan formula persamaan 3.3 yaitu :

$$W_{ij} = tf_{ij} \times \log \frac{D}{df_j}$$

$$W_{rebahan} = 0{,}2 \times \log \frac{3}{2}$$

$$W_{rebahan} = 0{,}0352$$

Selanjutnya dapat dihitung nilai bobot untuk setiap *term* pada *query* dalam masing-masing dokumen seperti berikut :

Tabel 3.1 Nilai TF IDF

Q	TF			*df*	IDF	*W*		
	d_1	d_2	d_3			d_1	d_2	d_3
rebahan	1	1	0	2	0,176	0,176	0,176	0
hiburan	1	0	0	1	0,477	0,176	0	0
informasi	0	1	1	2	0,176	0	0,176	0,176

Dengan

w = bobot dari *term*

tf = jumlah kata dalam masing-masing dokumen

D = jumlah seluruh dokumen

df = jumlah dokumen yang mengandung *termx* di dalamnya

Q = kata/*query*

d_i = dokumen ke-i ,i = 1,2,3,....n

Pada tabel 3.1 bobot pada masing-masing dokumen menunjukkan besarnya tingkat kerelevanan/kesesuaian antara dokumen dengan *query*. Nilai bobot pada dokumen berbanding lurus dengan tingkat similaritas dokumen terhadap *query* yang dicari.

3.11. Analisis Sentimen

Analisis Sentimenatau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan atau beropini negatif atau positif (Liu, 2010).

3.12. Klasifikasi

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep dengan tujuan memprediksikan kelas untuk data yang tidak diketahui kelasnya (Han dan Kamber, 2006). Di dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pedapatan rendah (Kusrini, 2009)

Dalam melakukan klasifikasi data terdapat dua proses yang dilakukan yaitu:

1. Proses *training*

   Pada proses *training* digunakan *training set* yang telah diketahui label-labelnya untuk membangun model atau fungsi.

2. Proses *testing*

   Untuk mengetahui keakuratan model atau fungsi yang akan dibangun pada proses *training*, maka digunakan data yang disebut dengan *testing set* untuk memprediksi label-labelnya

Klasifikasi pada analisis sentimen biasanya digunakan untuk menyelesaikan masalah klasifiaksi dua kelas, seperti positif dan negatif. Pada ulasan *online* memiliki nilai penilaian yang ditetapkan oleh para peneliti, misalnya kelas positif dan negatif ditentukan menggunakan peringkat bintang 1 sampai 5, ulasan dengan bintang 4 atau 5 dianggap sebagai tinjauan positif, dan ulasan dengan bintang 1 sampai 2 dianggap sebagai ulasan negatif. (Liu, 2012)

2.13 *Confusion Matrix*

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting. Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. Salah satu cara untuk melakukan pengukuran kinerja suatu sistem klasifikasi adalah menggunakan *confusion matrix*. Menurut Han dan Kamber (2011), *confusion matrix* adalah alat yang berguna untuk menganalisis seberapa baik *classifier* mengenali tuple dari kelas yang berbeda. TP dan TN memberikan informasi ketika *classifier* benar, sedangkan FP dan FN memberitahu ketika classifier salah, tuple positif dikenali sebagai negatif dan tuple negatif

dikenali sebagai positif. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Tabel berikut menunjukkan *Confussion Matrix* :

Tabel 3.2 Tabel *Confussion Matrix*

	Actual Positive	*Actual Negative*
Predicted Positive	*True* Positif (TP)	*False* Positif (FP)
Predicted Negative	*False* Negatif (FN)	*True* Negatif (TN)

Keterangan :

a. TP (*True Positive*) : banyaknya data yang kelas aktualnya adalah kelas positif dengan kelas prediksinya merupakan kelas positif.
b. FN (*False Negative*) : banyaknya data yang kelas aktualnya adalah kelas positif dengan kelas prediksinya merupakan kelas negatif.
c. FP (*False Positiv*e) : banyaknya data yang kelas aktualnya adalah kelas negatif dengan kelas prediksinya merupakan kelas positif.
d. TN (*True Negative*) : banyaknya data yang kelas aktualnya adalah kelas negatif dengan kelas prediksinya merupakan kelas negatif.

Terdapat beberapa rumus umum yang dapat digunakan untuk menghitung performa klasifikasi. Hasil dari nilai akurasi, presisi, dan recall biasa ditampilkan dalam persentase.

3.13.1 *Accuracy* (Akurasi)

Akurasi merupakan metode pengujian bedasarkan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Pada akurasi menjawab pertanyaan “Berapa persen cuitan pengguna twitter yang benar diprediksi melakukan *hate speech*dan tidak melakukan *hate speech* dari keseluruhan cuitan pengguna twitter?”

Persamaan akurasi seperti pada persamaan berikut :

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3.4)$$

3.13.2 *Precission* (Presisi)

Presisi menggambarkan proporsi jumlah teks yang relevan terkenali diantara semua dokumen teks yang terpilih untuk kebutuhan informasi. *Precission* menjawab pertanyaan "Berapa persen cuitan pengguna Twitter yang benar cuitannya melakukan *hate speech* dari keseluruhan cuitan pengguna Twitter yang diprediksi melakukan *hate speech*?"Persamaan presisi seperti pada persamaan berikut :

$$presisi = \frac{TP}{(TP + FP)} \quad (3.5)$$

3.13.3 *Recall*

Recall merupakan proporsi jumlah yang dapat ditemukan kembali oleh sebuah proses pencarian. *Recall* menjawab pertanyaan "Berapa persen cuitan pengguna Twitter yang diprediksi melakukan *hatespeech* dibandingkan keseluruhan cuitan pengguna Twitter yang sebenarnya melakukan *hate speech*?"

Persamaan *recall* seperti pada persamaan berikut :

$$recall = \frac{TP}{(TP + FN)} \quad (3.6)$$

3.13.4 Nilai *AUC* (*Area Under Curve*)

Nilai *AUC* memberikan gambaran tentang keseluruhan pengukuran atas keseuaian model yang digunakan. Semakin besar nilai *AUC* maka semakin baik pengukuran klasifikasi yang diteliti. Maka dapat dilihat karakteristik dari *AUC* adalah sebagai berikut :

Tabel 3. 3 Nilai Area Under Curve (AUC)

Nilai AUC	*Kategori*

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*

3.14. Teorema Bayes

Teoremea Bayes ditemukan oleh Thomas Bayes pada tahun 1763. Teorema *bayes* digunakan untuk menghitung probabilitas terjadinya suatu peristiwa berdasarkan pengaruh yang didapat dari hasil observasi sebelumnya. Teorema ini menerangkan hubungan ntara probabilitas terjadinya peristiwa A dengan syarat peristiwa B telah terjadi dan probabilitas terjadinya peristiwa B dengan syarat peristiwa A telah terjadi. Teorema Bayes merupakan penyempurnaan dari probabilitas bersyarat yang haya dibatasi dua buah kejadian. Sehingga teorema ini diperluas untuk "n" buah kejadian.

Teorema Bayes merupakan dasar aturan dari *naive bayes classifier* berikut teorema bayes pada persamaan 3.4

$$P(B_n \mid A) = \frac{P(A \mid B_n)P(B_n)}{\sum_{i=1}^{n} P(A \mid B_n)P(B_n)} \qquad (3.7)$$

Dimana:

A : data sampel dengan label kelas belum diketahui

Bn : suatu hipotesis yang akan menentukan A masuk ke dalam suatu kelas

P(Bn|A): peluang Bn yang merupakan data *tuple* atau bukti yang diperoleh pada saat observasi masuk ke dalam suatu kelas, (probabilitas *posterior*, Bn dikondisikan pada A)

P(Bn) : probabilitas *prior*, atau probabilitas sebelumnya

P(Bn|A): probabilitas *posterior* dimana A dikondisikan pada Bn.

P(A) : merupakan probabilitas sebelumnya dari A

3.15 *Naive Bayes Classifier*

Naive bayes classifier (NBC) merupakan salah satu metode pengklasifikasi statistik, dimana pengklasifikasian ini dapat memprediksi probabilitas keanggotaan kelas suatu data yang akan masuk ke dalam kelas tertentu, sesuai dengan perhitungan probabilitas (Handayani et al., 2015) Definisi lain mengatakan *naive bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris yang bernama Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan penglaman di masa sebelumnya (Bustami, 2013)

Pada NBC merupakan model penyederhanaan dari metode Bayes yang cocok dalam pengklasifikasian teks atau dokumen. NBC mengasumsikan independensi di antara kemunculan kata dalam suatu dokumen tanpa memperhitungkan urutan kata dan informasi yang terdapat pada kalimat atau dokumen. Selain itu metode ini memperhitungkan jumlah kemunculan kata dalam dokumen. Metode NBC dalam melakukan perhitungan peluang sebuah kata masuk ke dalam kategori dapat dilakukan dengan menggunakan persamaan probabilitas bersyarat. Pada saat klasifikasi, pendekatan Bayes akan menghasilkan label kategori yang paling tinggi probabilitasnya (v_{MAP}) dengan memasukkan atribut $(a_1, a_2, ..., a_n)$

$$H_{MAP} = \arg\max P(h_j \mid a_1, a_2, ..., a_n) \tag{3.8}$$

Dimana untuk :

H_{MAP} : Nilai *output* hasil klasifikasi

h_j : himpunan kategori kelas

$(a_1, a_2, ..., a_n)$: atribut dokumen *term*, dimana a_1 adalah kata pertama, a_2 kedua dan seterusnya.

Dengan mensubtitusikan teorema Bayes pada persamaan (3.4) maka dapat ditulis :

$$H_{MAP} = \arg\max \frac{P(a_1, a_2, ..., a_n \mid h_j) \cdot P(h_j)}{P(a_1, a_2, ..., a_n)} \tag{3.9}$$

$P(h_j \mid a_1, a_2, ..., a_n)$ nilainya konstan untuk setiap v_j maka nilainya dapat diabaikan, sehingga persamaan ini dapat ditulis sebagai berikut :

$$H_{MAP} = \arg\max P(a_1, a_2, ..., a_n \mid h_j) \cdot P(h_j) \qquad (3.10)$$

Tingkat kesulitan menghitung $P(a_1, a_2, ..., a_n \mid h_j)$ menjadi tinggi karena jumlah *term* $P(a_1, a_2, ..., a_n \mid h_j)$ bisa jadi akan sangat besar. Hal ini disebabkan jumlah *term* tersebut sama dengan jumlah semua kombinasi posisi *term* dikalikan dengan jumlah kategori yang akan diklasifikasikan. *Naive Bayes Classifier* menyederhanakan hal ini dengan mengasumsikan bahwa di dalam setiap kategori, setiap kata tidak bergantung satu sama lain atau independen.

Dengan menggunakan persamaan 3.4, maka persamaan 3.8 dapat dituliskan menjadi :

$$H_{MAP} = \arg\max P(h_j) \prod_i P(a_i \mid h_j)$$

$P(h_j)$ dan probabilitas kata ai untuk setiap katehgori $P(a_i \mid h_j)$ dihitung pada saat pelatihan :

$$P(h_j) = \frac{|docs_j|}{|contoh|}$$

Kelemahan dari probabilitas parameter $P(a_i \mid h_j)$ apabila sangat kecil (mendekati nol). Oleh karena itu digunakan metode *smoothing* pada Naive Byes untuk menyelesaikan masalah tersebut (Hafilizara, 2014)

$$P(a_i \mid h_j) = \frac{n_k + 1}{n + |kosakata|}$$

Dimana untuk :

$P(h_j)$: Probabilitas setiap dokumen terhadap sekumpulan dokumen

$P(a_i \mid h_j)$: Probabilitas kemunculan kata a_i pada suatu dokumen dengan kategori kelas h_j

|*docs*| : jumlah kata pada kategori h_j

|contoh| : jumlah dokumen yang digunakan dalam penelitian

n_k : jumlah kemunculan kata a_i pada kategori h_j

n : jumlah kata pada kategori h_j

kosakata : jumlah kata pada dokumen latihan.

Contoh :

Pada dokumen yang dimiliki mempunyai klasifikasi / *class* yaitu olahraga, teknologi, dan otomotif. Pada dokumen yang ke 7 berisikan "Madrid Anti Barcelona" belum memiliki kategori sehingga dilakukan analisis.

Tabel 3.4 Dokumen Teks

Dokumen	Teks	Kategori
1.	BarcelonaKalahkan Madrid	Olah Raga
2.	4G LTE Indosat Sudah Aktif	Teknologi
3.	Ancelolti : Barcelona "Bantu" Madrid	Olah Raga
4.	Oli Mesin Anti Panas	Otomotif
5.	Barcelona VS Madrid	Olah Raga
6.	Jaringan Baru 4G LTE	Teknologi
7.	Madrid Anti Barcelona	?

Terdapat 17 kata yang terdapat pada dokumen tabel **3.4** diatas, yaitu 4G, Aktif, Ancelotti, Anti, Bantu, Barcelona, Baru, Indosat, Jaringan, LTE, Madrid, Mesin, Oli, Panas, Sudah, Tumbangkan, VS. Dari kumpulan dokumen pada tabel 3.1 akan berbentuk *term document matrix* sebagai berikut :

Tabel 3. 5*Term Document Matrix*

Doc	4G	Aktif	Ancelotti	Anti	Bantu	Barcelona	Baru	Indosat	Jaringan	LTE
1						1				
2	1	1						1		1
3			1		1	1				
4				1						
5						1				
6	1						1		1	1
7						1				

Doc	Madrid	Mesin	Oli	Panas	Sudah	Kalahkan	VS	Class
1	1					1		OlahRaga
2					1			Teknologi
3	1							OlahRaga
4		1	1	1				Otomotif
5	1						1	OlahRaga
6								Teknologi
7	1							?

Kemudian dokumen berdasarkan klasifikasi atau *class* olahraga

Tabel 3. 6 Dokumen Klasifikasi Olahraga

Doc	4G	Aktif	Ancelotti	Anti	Bantu	Barcelona	Baru	Indosat	Jaringan	LTE
1						1				
3			1		1	1				
5						1				

Doc	Madrid	Mesin	Oli	Panas	Sudah	Kalahkan	VS	Class
1	1					1		Olah Raga
2	1							Olah Raga
3	1						1	Olah Raga

Kemudian dihitung P(Ancelotti|Olahraga), P(Bantu|Olahraga), P(Barcelona|Olahraga), P(Madrid|Olahraga), P(Kalahkan|Olahraga), dan P(VS|Olahraga). Pada dokumen class olah raga terdapat 10 kata .

$$P(OlahRaga) = \frac{3}{6} = 0.5$$

$$p(Ancelotti|OlahRaga) = \frac{1+1}{10+17} = 0.074$$

$$p(Bantu|OlahRaga) = \frac{1+1}{10+17} = 0.074$$

$$p(Barcelona|OlahRaga) = \frac{3+1}{10+17} = 0.1481$$

$$p(Madrid|OlahRaga) = \frac{2+1}{10+17} = 0.1481$$

$$p(Kalahkan|OlahRaga) = \frac{1+1}{10+17} = 0.074$$

$$p(VS|OlahRaga) = \frac{1+1}{10+17} = 0.074$$

Kemudian dokumen berdasarkan klasifikasi teknologi dihitung P(4G|Tekonologi), P(Aktif|Tekonologi), P(Baru|Tekonologi), P(4G|Tekonologi), P(Indosat|Olahraga), P(Jaringan|Tekonologi), dan P(Sudah|Tekonologi). Terdapat 9 kata pada dokumen yang memiliki *class* Teknologi.

Tabel 3. 7 Dokumen Klasifikasi Teknologi

Doc	4G	Aktif	Ancelotti	Anti	Bantu	Barcelona	Baru	Indosat	Jaringan	LTE
2	1	1						1		1
6	1						1		1	1

Doc	Madrid	Mesin	Oli	Panas	Sudah	Kalahkan	VS	Class
2					1			Teknologi
6								Teknologi

$$P(Teknologi) = \frac{2}{6} = 0.334$$

$$p(4G|Teknologi) = \frac{2+1}{9+17} = 0.1153$$

$$p(Aktif|Teknologi) = \frac{1+1}{9+17} = 0.0769$$

$$p(Baru|Teknologi) = \frac{1+1}{9+17} = 0.0769$$

$$p(Indosat|Teknologi\ ) = \frac{1+1}{9+17} = 0.0769$$

$$p(Jaringan|Teknologi) = \frac{1+1}{9+17} = 0.0769$$

$$p(LTE|Teknologi) = \frac{2+1}{9+17} = 0.1153$$

$$p(Sudah|Teknologi) = \frac{1+1}{9+17} = 0.0769$$

Kemudian dokumen berdasarkan klasifikasi otomotif dihitung P(Anti|Otomotif), P(Mesin|Otomotif), P(Oli|Otomotif), dan P(Panas|Otomotif). Terdapat 4 kata pada dokumen yang memiliki *class* Otomotif.

Tabel 3. 8 Dokumen Klasifikasi Otomotif

Doc	**4G**	**Aktif**	**Ancelotti**	**Anti**	**Bantu**	**Barcelona**	**Baru**	**Indosat**	**Jaringan**	**LTE**
4				1						

Doc	Madrid	Mesin	Oli	Panas	Sudah	Kalahkan	VS	*Class*
4		1	1	1				Teknologi

$$P(Otomotif) = \frac{1}{6} = 0.167$$

$$p(Anti|Otomotif) = \frac{1+1}{4+17} = 0.0952$$

$$p(Mesin|Otomotif) = \frac{1+1}{4+17} = 0.0952$$

$$p(Oli|Otomotif) = \frac{1+1}{4+17} = 0.0952$$

$$p(Panas|Otomotif) = \frac{1+1}{4+17} = 0.0952$$

Kemudian dihitung pada dokumen 7 berisikan´"Madrid Anti Barcelona"
Hasil probabilitas pada klasifikasi OlahRaga :

P(Olahraga) =P(OlahRaga) + P(Barcelona|Olahraga) + P(Anti|Olahraga) + P(Madrid|Olahraga)

= 0.5 + 0.1481 + 0.0370 + 0.1481

= 0.8332

Hasil probabilitas pada klasifikasi Teknologi :

P(Teknologi) =P(Teknologi) + P(Barcelona|Tekonologi) + P(Anti|Teknologi) + P(Madrid|Teknologi)

= 0.334 + 0.0385 + 0.0385 + 0.385

= 0.4494

Hasil probabilitas pada klasifikasi Otomotif :

P(Otomotif) =P(Otomotif) + P(Barcelona|Otomotif) + P(Anti|Otomotif) + P(Madrid|Otomotif)

= 0.167 + 0.0476 + 0.0952 + 0.0476

= 0.3574

Sehingga diperoleh P(Olah Raga) sebesar 0.8332, P(Teknologi) sebesar 0.4494, dan P(Otomotif) sebesar 0.3574. Karena P(Olahraga) > P(Teknologi) > P(Otomotif) maka dapat disimpulkan bahwa dokumen 7 tersebut diklasifikasikan sebagai dokumen Olah Raga.

3.16 *Support Vector Machine (SVM)*

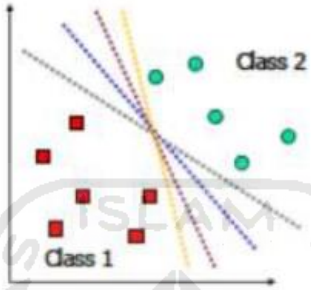
Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik, pertama kali diperkenalkan pada tahun 1992 di *Annual Workshop on Computional Learning Theory* dalam rangkaian dari beberapa konsep-konsep unggulan dalam bidang *pattern recognition*.

Support Vector Machine (SVM) dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas yang berbeda pada ruang *input* (Christianini & Shawe-taylor, 2000). *SVM* memiliki prinsip dasar *linier classifier* dimana kasus klasifikasi dapat dipisahkan secara linier, dan selanjutnya dikembangkan agar dapat bekerja pada problem *non-linier* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Pada ruang berdimensi tinggi, akan dicari *hyperplane* terbaik yang dapat memaksimalkan *margin* antara kelas data. *Margin* adalah jarak antara *hyperplane*

tersebut dengan pola yang terdekat dari masing-masing kelas. *Pattern* yang paling dekat disebut *support vector* (Santosa, 2007).

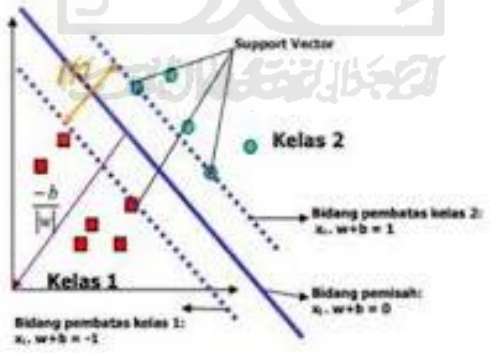
3.16.1. *LinearlySeparable Data*

Menurut Osuna et al. (1997) *linearly separable data* merupakan data yang dapat dipisahkan secara linier. Dalam kasus analisis sentimen, *hyperplane* pada *SVM* akan memisahkan kelas dengan sentimen positif, negatif maupun netral.



Gambar 3.4. Alternatif Bidang Pemisah
(Sumber : Osuna et al. 2007)

Pada Gambar 3.1 dapat dilihat alternatif bidang pemisah yang dapat memisahkan semua dataset sesuai dengan kelasnya.



Gambar 3.5. Bidang Pemisah Terbaik Dengan *Margin* (M) Terbesar
(Sumber : Osuna et al. 2007)

Dalam SVM objek data terluar yang paling dekat dengan *hyperplane* disebut *support vector*. Objek tersebut paling sulit diklasifikasikan karena posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Mengingat sifatnya yang kritis, hanya *support vector* yang diperhitungkan untuk menemukan *hyperplane* paling optimal pada SVM.

Misalkan $x_i = \{x_1,...,x_n\}$ adalah *dataset*, sedangkan label masing-masing dinotasikan $y_i \in \{-1,+1\}$ untuk i = 1,2,....,*l*, yang mana *l* adalah banyaknya data. Diasumsikan kedua kelas -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi *d*, yang didefinisikan :

$$w \cdot x + b = 0 \tag{3.11}$$

Pattern x_i yang termasuk kelas+1 (sampel positif) memenuhi pertidaksamaan :

$$w \cdot x + b \geq +1 \tag{3.12}$$

Sedangkan *pattern* x_i yang termasuk -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan :

$$w \cdot x + b \leq -1 \tag{3.13}$$

Dimana *w* adalah *n*-dimensi bobot vektor dan *b* adalah pengali skala atau nilai bias. Persamaan ini menemukan maksimum margin untuk memisahkan kelas dari kelas positif dan kelas negatif.

keterangan:

xi = titik data

yi = kelas data

w = vektor bobot yang tegak terhadap *hyperplane*

b = posisi bidang relatif terhadap pusat koordinat

Nilai *margin* antara bidang pembatas adalah $\frac{1-b-(-1-b)}{w} = \frac{2}{\| w \|}$. Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan 3.14 dan 3.15. Selain itu, karena memaksimalkan $\frac{1}{\| w \|}$ sama denga meminimumkan $\| w \|^2$ dan jika kedua bidang pembatas pada *constraint* 3.14 dan 3.15 direpresentasikan dalam pertidaksamaan

$$y_i (x_i \cdot w + b) - 1 \geq 0 \tag{3.14}$$

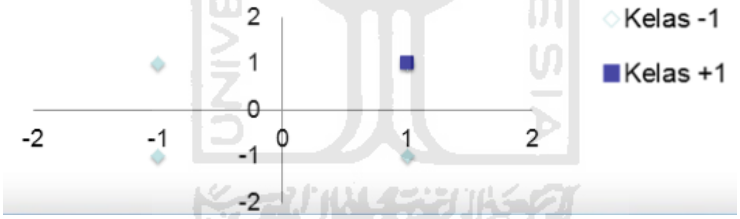
Maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi constraint, yaitu :

$$\min_w \frac{1}{2}\|w\|^2$$
$$s.\cdot t\cdot y_i(x_i\cdot w+b)-1\geq 0 \qquad (3.15)$$

Contoh :

Tabel 3.9 Contoh Perhitungan SVM

x_1	x_2	**Kelas (y)**	**Support Vector (SV)**
1	1	1	1
1	-1	-1	1
-1	1	-1	1
-1	-1	-1	0



Karena ada dua fitur (x1 dan x2), maka w juga akan memiliki 2 fitur (w1 dan w2). Formulasi yang digunakan adalah sebagai berikut :

Meminimalkan nilai margin :

$$\frac{1}{2}\|w\|^2=\frac{1}{2}(w_1{}^2+w_2{}^2)$$

Dengan syarat :

$y_i(\vec{x}_i\cdot\vec{w}+b)\geq 1$, i=1,2,3,...,N

Sehingga didapatkan persamaan berikut :

1. $(w_1+w_2+b)\geq 1$ untuk $y_1=1,\ x_1=1,\ x_2=1$
2. $(-w_1+w_2-b)\geq 1$ untuk $y_2=-1,\ x_1=1,\ x_2=-1$
3. $(w_1-w_2-b)\geq 1$ untuk $y_3=-1,\ x_1=-1,\ x_2=1$

4. $(w_1 + w_2 - b) \geq 1$ untuk $y_1 = -1,\ x_1 = -1,\ x_2 = -1$

Menjumlahkan persamaan 1 dan 2 :

$(w_1 + w_2 + b) \geq 1$

$(-w_1 + w_2 - b) \geq 1$

_____+

$2w_2 = 2$

$w_2 = 1$

Menjumlahkan persamaan 1 dan 3 :

$(w_1 + w_2 + b) \geq 1$

$(w_1 - w_2 - b) \geq 1$

_____+

$2w_1 = 2$

$w_1 = 1$

Menjumlahkan persamaan 2 dan 3 :

$(-w_1 + w_2 - b) \geq 1$

$(w_1 - w_2 - b) \geq 1$

_____+

-2b = 2

b=-1

Berdasarkan hasil di atas didapatkan persamaan hyperlane :

$w_1 x_1 + w_2 x_2 + b = 0$

$x_1 + x_2 - 1 = 0$

$x_2 = 1 - x_1$

3.16.2 *Non-LinearlySeparable Data*

Metode *SVM* pada *Nonliniearly Separable Data* untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier, sehingga harus dimodifikasi karena

tidak adanya solusi yang ditemukan. Oleh karena itu, kedua *constraint* (3.14 & 3.15) harus diubah sehingga lebih fleksibel dengan penambahan variabel ξ_i menjadi $(x_i \cdot w + b) \geq 1 - \xi i$ untuk kelas 1 dan $(x_i \cdot w + b) \leq 1 + \xi i$ untuk kelas 2. Pencarian bidang pemisah terbaik dengan penambahan variabel ξ_i sering juga disebut *soft margin hyperplane.* Dengan demikian formula pencarian bidang pemisah terbaik berubah menjadi seperti persamaan berikut:

$$\begin{aligned} &\min_w \frac{1}{2}|w|^2 + C\left(\sum_{i=1}^{n} \xi_i\right) \\ &s. \cdot t \cdot y_i (w \cdot x_i + b) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \qquad (3.16)$$

Dimana C (*Complexity*) adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Nilai C yang besar berarti akan memberikan penalti yang lebih besar terhadap *error* klasifikasi tersebut (Sembiring, 2007). Tujuan adanya nilai C untuk meminimalkan *error* dan memperkecil nilai *slack*. Jika nilai C mendekati nol, maka lebar margin pada bidang pembatas menjadi maksimum dan jumlah data latih yang berada dalam *margin* atau yang ada posisi yang salah tidak akan dipedulikan. Hal ini berarti akan mengurangi tingkat akurasi pada proses *training*, sehingga mengakibatkan data uji tidak dapat diklasifikasikan dengan baik.

Secara umum, kasus-kasus di dunia nyata adalah kasus yang tidak linier. Data ini sulit dipisahkan secara *linier*. Metode kernel adalah salah satu untuk mengatasinya (Santosa, 2010). Pada mulanya teknik *machine learning* dikembangkan dengan asumsi kelinieran. Fungsi kernel memungkinkan untuk mengimplementasikan suatu model pada ruang dimensi lebih tinggi (ruang fitur). Fungsi kernel memungkinkan untuk memetakan dimensi awal (dimensi yanglebih rendah) himpunan data ke dimensi baru (dimensi yang relatif lebih tinggi).Menurut Prasetyo (2012) macam fungsi kernel diantaranya:

(Prasetyo, 2012)

1. *Kernel Linier*

   Rumus *Kernel Linier* ditunjukkan pada persamaan

$$K = (x, y) = x, y \qquad (3.17)$$

2. *Kernel Polynomial*

   Rumus *kernel Polynomial* ditunjukkan pada persamaan

$$K(x, y) = (x, y + c)^d \tag{3.18}$$

3. *Kernel Radial Basis Function (RBF)*

   Rumus *kernel RBF* ditunjukkan pada persamaan

$$K(x_{i,} x_j) = \exp(-\gamma \mid (x_i - x_j) \mid)^2 \tag{3.19}$$

BAB IV
METODOLOGI PENELITIAN

4.1 Populasi dan Sampel Penelitian

Populasi dalam penelitian ini adalah data *tweet* ataukicauandan *retweet*yang berhubungan dengankesehatan mental di Indonesia melalui media sosial Twitter. Sedangkan sampel yang digunakan dalam penelitian ini adalah data *tweet* yang diambil pada bulan Agustus 2020. Total data yang dikumpulkan berjumlah 498 data *tweet*.

4.2 Pengambilan Data

Proses pengambilan data dari media sosial Twitter dilakukan dengan cara menggunakan teknik *crawling*, yaitu proses pengambilan data *tweet* semi-terstruktur dari web dengan memanfaatkan API yang disediakan oleh Twitter. API *Key* Twitter adalah *Application Programming Interface*, yaitu sekumpulan perintah, fungsi, komponen dan juga protokol yang disediakan untuk mempermudah program pada saat membangun perangkat lunak. API key Twitter memiliki suatu *consumer key, consumer secret, access key*, dan *access secret* yang digunakan sebagai kunci untuk dapat mengakses data Twitter yang akan dibutuhkan. Proses ini dibangun dengan bahasa pemrograman R versi 3.5.1.

4.3 Variabel dan Definisi Operasional Variabel

Variabel yang digunakan dalam penelitian ini ditampilkan dalam Tabel 4.1 tentang penjelasan dan definisi operasional penelitian :

Tabel 4.1 Definisi Operasional Variabel

Variabel	Definisi Operasional Variabel
Data Teks	Data tweet dengan *hashtag*#kesehatanmental, #mentalhealth, #mentalhealthawareness, #mentalhealthawareness, #sehatmental
Scoring	Penilaian yang diberikan terhadap tweet berdasarkan *stopword* positif dan negatif
Klasifikasi	Keputusan klasifikasi berdasarkan proses *scoring* apabila *tweet* mendapat *score* ≤ -1 maka akan mendapat klasifikasi sentimen negatif dan score ≥ 1 akan mendapat klasifikasi sentimen positif.

4.4 Metode Analisis Data

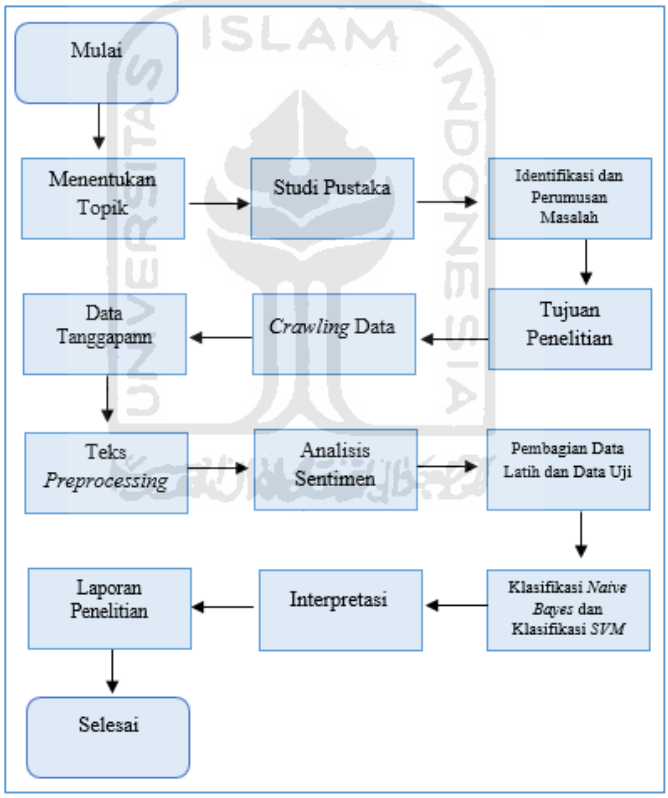
Proses analisis dalam penelitian ini mmenggunakan *software API Key Twitter,Microsoft Excel* 2016 dan *Rstudio* versi 3.5.1. Terdapat beberapa metode yang digunakan dalam penelitian ini yaitu :

1. *Web Crawling*, digunakan untuk mengumpulkan data tweet pengguna media sosial Twitter secara online.
2. *Text Mining*, digunakan untuk melakukan analisis data yang berupa teks yang tidak terstruktur.
3. *Word Cloud*, digunakan untuk menampilkan hasil visualisasi data yang paling banyak digunakan
4. Analisis Sentimen, digunakan untuk melakukan pelabelan data ke dalam sentimen positif atau negatif.

5. Klasifikasi *machine learning* dengan algoritma *Naive Bayes Classifier* dan *Support Vector Machine* yang digunakan untuk mengklasifikasikan tweet berdasarkan sentimen positif atau sentimen negatif serta melihat tingkat akurasi dalam melakukan klasifikasi teks.

4.5 Langkah Penelitian

Tahapan atau langkah penelitian ini digambarkan dalam visualisasi flowchart melalui Gambar 4.1 berikut :



Gambar 4. 2 Tahapan Penelitian

Penjelasannya sebagai berikut :

1. Tahap pertama setelah memulai penelitian adalah menentukan topik, mencari studi pustaka dan menentukan tujuan dari penelitian.

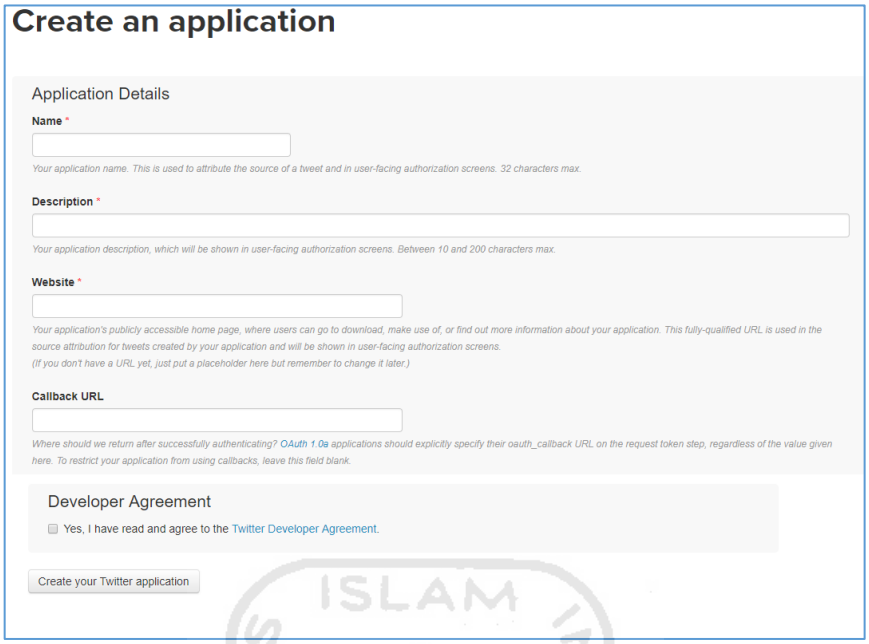
2. Setelah tahap pertama terselesaikan, tahap kedua adalah mencari data di media sosial Twitter dengan cara *web crawling* sesuai dengan topik yang telah ditentukan sebelumnya.
3. Data yang diperoleh masih berbentuk *rds* yang belum terstruktur sehingga perlu dilakukan proses *preprocessing* untuk mengubahnya menjadi data yang lebih terstruktur.
4. Setelah data diproses pada tahap *preprocessing*, tahap selanjutnya yaitu pelabelan kalimat positif dan negatif menggunakan kamus positif dan negatif untuk dilakukan analisis sentimen.
5. Tahap selanjutnya adalah pembagian data untuk memulai proses klasifikasi. Pembagian data dibagi menjadi dua yaitu pembagian data latih dan data uji, dengan jumlah data latih sebanyak 80% dan data uji sebanyak 20%. Pembagian ini merujuk pada jurnal-jurnal sebelumnya.
6. Selanjutnya menginputkan data latih dan menganalisisnya dengan *NBC* dan *SVM* yang dilanjutkan dengan proses pengujian menggunakan data uji.
7. Perhitungan nilai akurasi pada *Naive Bayes* dan *SVM*.
8. Interpretasi hasil pada *wordcloud*, *Naive Bayes*, dan *SVM*.

BAB V
HASIL DAN PEMBAHASAN

Berdasarkan kajian teori dan hasil-hasil penelitian sebelumnya, maka pada bab ini akan memaparkan tentang implementasi NBC dan SVM untuk sentimen analisis data tanggapan tentang kesehatan mental di Indonesia melalui media sosial Twitter.

5.1. Pengumpulan Data dengan *Web Crawling*

Proses *web crawling* dilakukan dengan menggunakan software R 3.5.1. Untuk melakukan *crawling* data pada Twittermembutuhkan kode untuk mengakses data Twittertersebut dengan memasukkan kode dari TwitterAPI. Untuk memiliki TwitterAPIdiharuskan melakukan registasi dahulu. Setelah melakukan registrasi, pilih *create new App* dan isi semua *field* seperti pada **gambar 5.1**:



Gambar 5. 1*Konfigurasi Pendaftaran API*

Pendaftaran APIdigunakan untuk mengonfirmasi pada pihak Twitter agar memberikan izin menjelajahi lebih luas terkait dengan data yang ingin dianalisis berkaitan dengan Twitter. Setelah registrasi dan bergabung dengan TwitterAPIdari TwitterAPI didapatkan beberapa kode berupa *consumer key*, *consumer secret*, *access token* dan *acces key* dari Twitter. Kode API tersebut adalah sebagai penguhubung antara Twitter dengan aplikasi lainnya, dalam penelitian ini kode tersebut dapat digunakan untuk proses integrasi antara TwitterAPI dengan *R studio.*Pada penelitian ini peneliti menggunakan 5 *hashtag* untuk mendapatkan tweet yang berkaitan dengan kesehatan mental yaitu #kesehatanmental, #mentalhealth, #mentalhealthawareness, #mentalhealthawareness, #sehatmental. Dari hasil proses *crawling* pada media sosial Twitter sebesar 498 data yang terdiri dari data *tweet* dan data *retweet*.

5.2. Teks *Preprocessing*

Data teks yang didapat setelah proses *web crawling* masih berupa susunan data yang tidak terstruktur yaitu dengan memiliki karakter-karakter atau tanda baca, penggunaan huruf kapital, dan ejaan yang salah atau singkatan yang tidak

mempunyai arti. Hal ini dapat mempengaruhi dalam pencarian informasi. Maka dari itu dilakukan proses *preprocessing* untuk membersihkan *noise* tersebut.

Tabel 5.1. Contoh Data untuk *Preprocessing*

No.	Tweet Tanggapan
1.	Gatauuu juga sih. Tapi ada yg bilang, *black dog* itu metafora yg dipakai buat melambangkan depresi. CMIIW.
2	Selain lambang *black dog*. Lambang itu "*semicolon*" jg dipakai buat org yg punya depresi
3	RT @qnafess Black dog itu apa? https://t.co/Lciqu5GzAB
4	@askmenfess Itu black dog. Biasanya orang2 yg lagi depresi berat menunjukan kalo dia lagi dlm fase tersebut pakai g... https://t.co/LWKVgykcth

Dalam Tabel 5.2. menunjukkan beberapa contoh *tweet* terkait kesehatan mental selama pandemi covid-19. *Tweet* tanggapan tersebut diambil secara acak pada masing-masing bulan.

5.2.1. *Cleaning*

Cleaning adalah proses menghilangkan delimeter-delimeter atau tanda baca (*remove punctuation*) ditunjukkan *highlight* berwarna kuning, nomor (*remove number*) ditunjukkan *highlight* berwarna kuning, *URL*(*remove URL*) ditunjukkan *highlight* berwarna hijau, *retweet*(*remove retweet*) ditunjukkan highlight berwarna biru dan *username* (*remove username*) ditunjukkan *highlight* berwarna ungu yang ada pada data teks.

Pada tabel 5.2 menunjukkan perubahan sebelum melakukan *cleaning* dan sesudah melakukan proses *cleaning.*

Tabel 5.2. Contoh Data untuk *Cleaning*

Sebelum Proses *Cleaning*	Setelah Proses *Cleaning*
Gatauuu juga sih. Tapi ada yg bilang, *black dog* itu metafora yg dipakai buat melambangkan depresi. CMIIW.	Gatauuu juga sih Tapi ada yg bilang *black dog* itu metafora yg dipakai buat melambangkan depresi CMIIW

Sebelum Proses *Cleaning*	Setelah Proses *Cleaning*
Selain lambang *black dog*. Lambang itu "*semicolon*" jg dipakai buat org yg punya depresi	Selain lambang *black dog* Lambang itu *semicolon* jg dipakai buat org yg punya depresi
RT@qnafessBlack dog itu apa?https://t.co/Lciqu5GzAB	Black dog itu apa
@askmenfess Itu black dog. Biasanya orang2 yg lagi depresi berat menunjukan kalo dia lagi dlm fase tersebut pakai g...https://t.co/LWKVgykcth	Itu black dog Biasanya orang yg lagi depresi berat menunjukan kalo dia lagi dlm fase tersebut pakai g

5.2.2. *Case Folding*

Proses *case folding* semua penggunaan huruf kapital (*highlight* warna oranye) akan diubah menjadi menyamakan seluruh data menjadi bentuk standar atau dalam hal ini huruf kecil atau *lowercase*.

Pada tabel 5.4 menunjukkan perubahan sebelum melakukan *case folding* dan sesudah melakukan proses *case folding*.

Tabel 5.3. Contoh Data untuk *Case Folding*

Sebelum Proses *Case Folding*	Setelah Proses *Case Folding*
Gatauuu juga sih Tapi ada yg bilang *black dog* itu metafora yg dipakai buat melambangkan depresi CMIIW	gatauuu juga sih tapi ada yg bilang *black dog* itu metafora yg dipakai buat melambangkan depresi cmiiw
Selain lambang *black dog*Lambang itu *semicolon* jg dipakai buat org yg punya depresi	selain lambang *black dog* lambang itu *semicolon* jg dipakai buat org yg punya depresi
Black dog itu apa	*black dog* itu apa
Itu *black dog*Biasanya orang yg lagi depresi berat menunjukan kalo dia lagi dlm fase tersebut pakai g	itu *black dog* biasanya orang yg lagi depresi berat menunjukan kalo dia lagi dlm fase tersebut pakai g

5.2.3. *Filtering*

Proses *filtering* data akan dilakukan proses penghilangan kata-kata yang tidak berkontribusi banyak pada isi dokumen seperti kata hubung, ejaan yang disingkat atau penulisan ejaan yang salah berdasarkan kamus *stopword*. Pada tabel 5.4. ditunjukkan dengan pemberian *highlight* berwarna biru

Tabel 5.4. Contoh Data untuk *Filtering*

Sebelum Proses *Filtering*	**Setelah Proses *Filtering***
gatauuujugasihtapiadayg bilang *black dog*itu metafora yg dipakai buat melambangkan depresi cmiiw	bilang *black dog* metafora dipakai melambangkan depresi
selain lambang *black dog* lambang itu*semicolon*jg dipakai buatorgygpunya depresi	lambang *black dog* lambang *semicolon* dipakai depresi
*black dog*itu apa	*black dog*
itu*black dog*biasanyaorangyglagi depresi berat menunjukankalodia lagidlm fase tersebutpakaig	*black dog*depresi berat fase pakai

5.2.4. *Tokenizing*

Proses tokenizing kalimat akan dipecah perkata yang tidak saling berhubungan yang disebut *term* yang nantinya dapat diidentifikasi.

Tabel 5.5. Contoh Data untuk *Tokenizing*

Sebelum Proses *Tokenizing*	**Setelah Proses *Tokenizing***
bilang *black dog* metafora dipakai melambangkan depresi	“bilang”“*black*”“*dog*”“metafora”“dipakai”“melambangkan”“depresi”
lambang *black dog* lambang *semicolon* dipakai depresi	“lambang” “*black*”“*dog*” “lambang” “*semicolon*” “dipakai” “depresi”
black dog	*“black dog”*
*black dog*depresi berat fase pakai	*“black”“dog”* “depresi”“berat”“fase”“pakai”

5.2.Document Term Matrix Pembobotan TF IDF

Pada dasarnya, pembobotan TF IDF bekerja menentukan frekuensi dari suatu kata kemudian dibandingkan dengan proporsi kata tersebut pada seluruh dokumen.

Tahapan awal adalah mencari DTM atau document term matrix yang adalah suatu matriks yang menggambarkan frekuensi kata yang terjadi dalam kumpulan dokumen. Pada DTM perlu mengubah kalimat kalimat pada data yang dimiliki menjadi sebuah angka atau mengubahnya dalam bentuk biner sehingga dapat memudahkan proses perhitungan selanjutnya.

Tabel 5.6 Contoh Tabel DTM

	Sabtu	**September**	**Informasi**	**Orang**	**Tua**	**Lakukan**	**Karantina**	**Pandemi**	**...**
1	1	1	1	1	1	1	1	1	...
2	0	0	0	0	0	0	0	1	...
3	0	0	0	0	0	0	0	1	...
4	0	0	0	0	0	0	0	1	...
5	0	0	0	0	0	0	0	1	...
6	0	0	0	0	0	0	0	1	...
7	0	0	0	0	0	0	0	0	...
8	0	0	0	0	0	0	0	0	...
9	0	0	0	0	0	0	0	1	...
10	0	0	0	0	0	0	0	1	...
11	0	0	0	0	0	0	0	1	...

	Sabtu	September	Informasi	Orang	Tua	Lakukan	Karantina	Pandemi	...
12	0	0	0	0	0	0	0	0	...
13	0	0	0	0	0	0	0	1	...
14	0	0	0	0	0	0	0	1	...
15	0	0	0	1	0	0	0	1	...
....	....	....	...	...	...	...	...	...	...

Pada proses TF atau term frequency digunakan unutk menetukan berapa sering suatu kata muncul dalam sebuah dokumen. Maka semakin banyak frekuensi kemunculan yang dimilki oleh suatu kata, maka semakin besar pula nilainya.

Tabel 5. 1 Contoh Perhitungan TF

	Sabtu	September	Informasi	Orang	Tua	Lakukan	Karantina	Pandemi	...
1	1	1/3=0,33	1/16=0,06	1/36=0,02	1/3=0,33	1	1	1/163=0,06	...
2	0	0	0	0	0	0	0	1/163=0,06	...
3	0	0	0	0	0	0	0	1/163=0,06	...
4	0	0	0	0	0	0	0	1/163=0,06	...
5	0	0	0	0	0	0	0	1/163=0,06	...
6	0	0	0	0	0	0	0	1/163=0,06	...
7	0	0	0	0	0	0	0	0	...
8	0	0	0	0	0	0	0	0	...
9	0	0	0	0	0	0	0	1/163=0,06	...

	Sabtu	**September**	**Informasi**	**Orang**	**Tua**	**Lakukan**	**Karantina**	**Pandemi**	**...**
10	0	0	0	0	0	0	0	1/163=0,06	...
11	0	0	0	0	0	0	0	1/163=0,06	...
12	0	0	0	0	0	0	0	0	...
13	0	0	0	0	0	0	0	1/163=0,06	...
14	0	0	0	0	0	0	0	1/163=0,06	...
15	0	0	0	1/36=0,02	0	0	0	1/163=0,06	...
....	....	....	...	...	...	...	...	...	...

Pada proses IDF atau Inverse Document Frequency merupakan sebuah perhitungan dari bagaimana kata didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Berbeda dengan TF yang semakin banyak frekuensi kemunculan maka nilainya akan semakin besar. Dalam IDF, semakin sedikit frekuensi kata muncul dalam dokumen, maka makin besar nilainya.

Tabel 5.8 Contoh Perhitungan IDF

	Sabtu	**September**	**Informasi**	**Orang**	**Tua**	**Lakukan**	**Karantina**	**Pandemi**	**...**
1	$\log\left(\frac{915}{1}\right)$ = 2,96	$\log\left(\frac{915}{3}\right)$ = 2,48	$\log\left(\frac{915}{16}\right)$ = 1,75	$\log\left(\frac{915}{36}\right)$ = 1,40	$\log\left(\frac{9}{}\right)$ = 2,96	$\log\left(\frac{915}{1}\right)$ = 2,96	$\log\left(\frac{915}{1}\right)$ = 2,96	$\log\left(\frac{915}{163}\right)$ =0,74	...
2	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
3	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
4	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
5	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
6	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
7	0	0	0	0	0	0	0	0	...
8	0	0	0	0	0	0	0	0	...
9	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
10	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
11	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
12	0	0	0	0	0	0	0	0	...

	Sabtu	September	Informasi	Orang	Tua	Lakukan	Karantina	Pandemi	...
13	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
14	0	0	0	0	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
15	0	0	0	$\log\left(\frac{915}{36}\right)$ = 1,40	0	0	0	$\log\left(\frac{915}{163}\right)$ =0,74	...
...	....	....	...	...	...	...	...	...	...

Nilai TF-IDF diperoleh dengan mengalikan nilai TF dengan nilai IDF.Pada dasarnya, TF IDF bekerja dalam menenukan frekuensi relatif suatu kata kemudian dibandingkan dengan proporsi kata tersebut pada seluruh dokumen. Dan dapat diketahui bahwa kata yang muncul di banyak dokumen bukan pembeda yang baik maka harus diberi pembobot yang terjadi dalam beberapa dokumen. Setelah mendapatkan nilai TF dan IDF, kemudian akan menghitung nilai TF-IDF. Perhitungan TF-IDF dapat dilihat pada **tabel 5.9**:

Tabel 5.9 Contoh Perhitungan TF IDF

	Sabtu	September	Informasi	Orang	Tua	Lakukan	Karantina	Pandemi	...
1	2,96	0,818	0,105	0,028	0,088	2,96	2,96	0,044	...
2	0	0	0	0	0	0	0	0,044	...

	Sabtu	September	Informasi	Orang	Tua	Lakukan	Karantina	Pandemi	...
3	0	0	0	0	0	0	0	0,044	...
4	0	0	0	0	0	0	0	0,044	...
5	0	0	0	0	0	0	0	0,044	...
6	0	0	0	0	0	0	0	0,044	...
7	0	0	0	0	0	0	0	0,044	...
8	0	0	0	0	0	0	0	0,044	...
9	0	0	0	0	0	0	0	0,044	...
10	0	0	0	0	0	0	0	0,044	...
11	0	0	0	0	0	0	0	0,044	...
12	0	0	0	0	0	0	0	0,044	...
13	0	0	0	0	0	0	0	0,044	...
14	0	0	0	0	0	0	0	0,044	...
15	0	0	0	0,028	0	0	0	0,044	...
....	....	....	...	...	...	...	...	...	...

Nilai yang diperoleh dari proses pembobotan TF IDF menunjukkan bahwa semakin besar nilai TF IDF dari suatu kata maka semakin besar pula hubungan kata tersebut dengan dokumen tersebut.

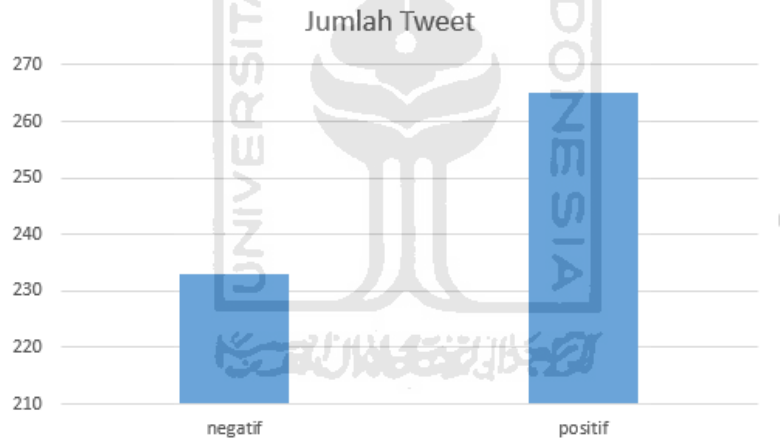
5.3. Analisis Sentimen

Setelah proses *preprocessing*, maka dilanjutkan proses pelabelan yang dilakukan secara otomatis dengan cara menghitung skor sentimen menggunakan kamus *colloquial lexicon* yang kemudian akan menentukan suatu kata akan masuk ke dalam kelas positif atau negatif. Jika skor bernilai > 0 maka akan diklasifikasikan

ke dalam kelas positif, skor bernilai < 0 maka akan diklasifikasikan ke dalam kelas negatif, dan jika skor bernilai = 0 maka akan diklasifikasikan ke dalam kelas netral. Klasifikasi kelas netral didapatkan karena mempunyai 2 kemungkinan :

a. Tidak terdapat kata sentimen dengan jumlah skor tinggi.
b. Jumlah skor kata positif sama dengan jumlah skor negatif.

Akan tetapi pada penelitian ini digunakan dua pelabelan kelas sentimen, yaitu sentimen positif dan sentimen negatif. Hal ini dilakukan karena menimbang kelas netral dianggap tidak memiliki nilai atau pengaruh dan dikhawatirkan akan menjadi bias. Sehingga kelas netral akan direduksi karena peneliti hanya akan menggunakan dua pelabelan kelas yaitu kelas positif dan kelas negatif. Berikut adalah hasil pelabelan kelas sentimen cuitan tanggapan pengguna Twitter di Indonesia tentang kesehatan mental.



Gambar 5.2 *Hasil Pelabelan*

Berdasarkan Gambar **5.2.**hasil pelabelan kelas sentimen menunujukkan bahwa jumlah cuitan positif memiliki frekuensi lebih rendah dibandingkan dengan jumlah cuitan negatif. Jumlah cuitan positif sebanyak 265 cuitan, dan cuitan negatif sebanyak 233 cuitan. Dalam menghitung jumlah skor dilakukan berdasarkan jumlah kata positif dikurangi jumlah kata negatif berdasarkan *colloquial lexicon* bahasa indonesia positif dan negatif.

Tabel 5.6. Skor Penilaian Klasifikasi

Klasifikasi	Teks	Jumlah
Negatif	barusan baca berita jakarta psbb ketat jalan keinget buruknya shock	-1
Negatif	pencegahan bunuh dirisedunia jatuh tanggal september its ok to ask for help bunuh diridicegah kesehatan	-3
Negatif	sayangnya orang berpikir fisik dominan mikir body shaming oke	-1
Negatif	army pencegahan bunuh diriinternasional kalinya kesempatan	-1
Negatif	pandemi sosialisasi psbb psbb ekonomimenuruntergganggu new normal terganggu	-3
Negatif	antisipasi otak temuan korupsi pengadaan alat rapid test	-1
Negatif	menghindari orang toxic kedengaran jahatsometimes menjaga kesehatan	-2
Negatif	mengancam nyawa tenaga medis dihantui risikokeletihanburnout syndrome beban system	-3
Negatif	psbb mempengaruhi pikiran keadaan buruk menyudutkan jarang perubahan sederhana	-1
Negatif	belajar memanusiakan manusia mengurangi stigmatisasi negatif	-2
Positif	gaji standar lingkungan kerjanyamanpengaruh fisik	-3
Positif	gaji tanggungjawab ditambah lingkungan kerja toxic utama	-2
Positif	refleksi bahagiahidup manusia terkadang	-2
Positif	syukur allah sujud syukurlimpahan rejekinya mentari	-3
Positif	manfaat jeda media sosial tubuh tempo cantik	-3
Positif	maaf teman mendekat hati	-1

Klasifikasi	Teks	Jumlah
Positif	selektif memilih teman	-2
Positif	suasana kampus responsif gender lainlain menurutku materi wajib banget dikasih perhatian	-2
Positif	menjaga pikiran sehat hatimu bersihsempurna menghindari	-3
Positif	kampanye sejalan hasil penelitian manfaat udara bersih	-3
Positif	lancarsukses tugas kuliah jaga fisik jaga	-2

Setelah*tweet* yang sudah melewati tahap *preprocessing*, selanjutnya dilakukan penghitungan skor kata pada proses *word scoring* untuk menentukan apakah sebuah *tweet* termasuk *tweet* positif atau negatif berdasarkan *lexicon* positif dan negatif yang berisi kata-kata positif dan negatif. Pada tabel 5.6 terdapat 10 tweet positif dan 10 tweet negatif yang kemudian dilihat kata mana saja yang terdapat pada masing-masing lexicon yang menunjukkan suatu tweet termasuk positif atau negatif. Untuk kata yang terdapat pada lexicon negatif diberi highlight merah, dan highlight hijau untuk kata yang terdapat pada tweet negatif. Cara menentukan kelas sentimen adalah dengan menghitung skor jumlah kata positif dikurangi skor jumlah kata negatif dalam setiap cuitan (Susanti, 2016). Adapun diambil kelas positif sebagai contoh perhitungan skor sentimen dalam proses pelabelan adalah jumlah kata positif – jumlah kata negatif

5.4. *Word Cloud*

Kumpulan kata-kata yang sering muncul tersebut dapat ditampilkan dalam bentuk *wordcloud*. Pada visualisasi *wordcloud* dapat dilihat topik dan kata-kata positif dan negatif yang sering digunakan pernggguna Twitter dalam memberikan tanggapan tentang kesehatan mental. Semakin besar ukutan katanya menggambarkan semakin tinggi frekuensi kata tersebut, yang artinya semakin sering pengguna Twitter menggunakan kata tersebut sebagai topik pembicaraan. Pada **gambar 5.** Merupakan *wordcloud* dari keseluruhan cuitan mengenai

kesehatan mental di Indonesia yang diperoleh dari media sosial Twitter pada bulan Agustus 2020.

Gambar 5.3 *Wordcloud*

Dari hasil visualisasi *wordcloud* pada gambar **Gambar 5. 3**dapat dilihat bahwa kata yang paling sering muncul atau banyak dibicarakan pada *tweet* mengenai kesehatan mental di Indonesia adalah “kesehatan” dengan frekuensi kemunculan sebanyak 152, “mental” dengan frekuensi kemunculan sebanyak 134, “tenaga*”* dengan frekuensi kemunculan sebanyak 74, “burnout” dengan frekuensi kemunculan sebanyak 68, “Indonesia” dengan frekuensi kemunculan sebanyak 68, “kalangan” dengan frekuensi kemunculan sebanyak 68, “keletihan” dengan frekuensi kemunculan sebanyak 68, “mayoritas” dengan frekuensi kemunculan sebanyak 67, “lingkungan” dengan frekuensi kemunculan sebanyak 61, dan “gaji” dengan frekuensi kemunculan sebanyak 59.

Tabel 5.10. Asosiasi Kata Tertinggi Keseluruhan Data

psbb	Jakarta	berita	baca	jalan
Buruknya (0,85)	Buruknya (0,93)	Baca (0,98)	Berita (0,98)	Buruknya (0,97)
Ketat (0,85)	Ketat (0,93)	Buruknya (0,98)	Buruknya (0,98)	Ketat (0,97)
Shock (0,85)	Shock (0,93)	Ketat (0,98)	Ketat (0,98)	Shock (0,97)
Baca (0,84)	Baca (0,91)	Shock (0,98)	Shock (0,98)	Baca (0,95)
Jakarta (0,84)	Berita (0,90)	Jalan (0,95)	Jalan (0,95)	Berita (0,95)

Pada asosiasi kata dilakukan untuk mengetahui keterkaitan antar kata. Pada penelitian ini, peneliti mengambil asosiasi tertinggi dari masing-masing kelas. Dapat dilihat pada **Tabel 5.7.** menunjukkan asosiasi kata dari keseluruhan data yang dimilki setelah melewati proses *preprocessing*. Dapat dilihat bahwa seluruh kata yang saling berkaitan memilki nilai paling rendah 0,84 dan tertinggi 0,98. Yang artinya keseluruhan asosiasi memilki nilai keterhubungan yang kuat berdasarakan (Sarwono, 2006) yaitu :

0 :Tidak ada korelasi antara dua variabel

>0 - 0,25 : Korelasi lemah

>0,25 - 0,5 : Korelasi cukup

>0,5 – 0,75 : Korelasi kuat

1 : Korelasi sangat kuat

Gambar 5.4*Tampilan Word Cloud Sentimen Positif*

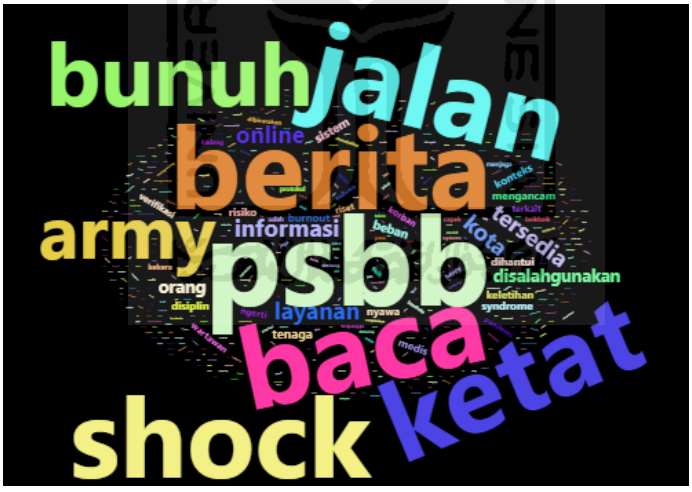
Pada tanggapan positif kata yang paling sering muncul atau banyak dibicarakan pada *tweet* mengenai kesehatan mental di Indonesia adalah, "lingkungan" dengan frekuensi kemunculan sebanyak 63*tweet*, "kerja" dengan frekuensi kemunculan sebanyak 62*tweet*, "gaji" dengan frekuensi kemunculan sebanyak 62*tweet*, "nyaman" dengan frekuensi kemunculan sebanyak 57, "berpengaruh" dengan frekuensi kemunculan sebanyak 56, "standar" dengan frekuensi kemunculan sebanyak 55, "membicarakan" dengan frekuensi kemunculan sebanyak 21, "tubuh" dengan frekuensi kemunculan sebanyak 20, "fisik" dengan frekuensi kemunculan sebanyak 16, dan "pandemi" dengan frekuensi kemunculan sebanyak 14. Perlu diketahui kata yang muncul merupakan kata-kata yang lolos saat melakukan *preprocessing* yaitu pada langkah *Cleaning* berdasarkan data *csvstopword* yang peneliti masukkan dan tidak peneliti ubah. Sehingga terkadang dapat diketemui beberapa kata yang sebenarnya memilki konotasi negatif masuk ke dalam sentimen positif atau sebaliknya.

Tabel 5.8. Asosiasi Kata Tertinggi Kelas Positif

media	sosial	tubuh
aktivitas (0,91)	aktifitas (0,89)	alam (0,76)
akun	akun	bioskop

media	sosial	tubuh
(0,91)	(0,89)	(0,76)
kebutuhan (0,91)	kebutuhan (0,91)	interaksi (0,76)
memiliki (0,79)	memiliki (0,77)	kekebalan (0,76)

Pada asosiasi kata dilakukan untuk mengetahui keterkaitan antar kata. Pada penelitian ini, peneliti mengambil asosiasi tertinggi dari masing-masing kelas. Dapat dilihat pada **Tabel 5.8.** menunjukkan asosiasi kata dari keseluruhan data yang dimilki setelah melewati proses *preprocessing*. Dapat dilihat bahwa seluruh kata yang saling berkaitan memilki nilai paling rendah 0,76 dan tertinggi 0,91. Yang artinya keseluruhan asosiasi memilki nilai keterhubungan yang kuat.



Gambar 5. 5. *Tampilan Word Cloud Sentimen Negatif*

Sedangkan pada tanggapan negatif katayang paling sering muncul atau banyak dibicarakan pada *tweet* mengenai kesehatan mental di Indonesia "psbb" dengan frekuensi kemunculan sebanyak 79, "berita" dengan frekuensi kemunculan sebanyak 74, "baca" dengan frekuensi kemunculan sebanyak 73, "jalan" dengan frekuensi kemunculan sebanyak 73, "Jakarta" dengan frekuensi kemunculan sebanyak 72, "ketat" dan "shock" dengan frekuensi kemunculan sebanyak 71, "bunuh" dengan frekuensi kemunculan sebanyak 56, "Army"

dengan frekuensi kemunculan sebanyak 51, “Layanan” dengan frekuensi kemunculan sebanyak 47, dan “Informasi” dengan frekuensi kemunculan sebanyak 47. Pada asosiasi kata dilakukan untuk mengetahui keterkaitan antar kata. Pada penelitian ini, peneliti mengambil asosiasi tertinggi dari masing-masing kelas. Dapat dilihat pada **Tabel 5.8.** menunjukkan asosiasi kata dari keseluruhan data yang dimilki setelah melewati proses *preprocessing*. Dapat dilihat bahwa seluruh kata yang saling berkaitan memilki nilai paling rendah 0,76 dan tertinggi 0,91. Yang artinya keseluruhan asosiasi memilki nilai keterhubungan yang kuat

Tabel 5.7. Asosiasi Kata Tertinggi Kelas Negatif

jalan	**psbb**	**bunuh**
Cobain (0,23)	ekonomi (0,23)	sedunia (0,50)
kaki (0,23)	sosialisasi (0,23)	dicegah (0,35)
kondisi (0,23)	new (0,22)	september (0,35)
rutin (0,84)	normal (0,22)	tanggal (0,35)
seru (0,84)		utas (0,35)

5.6. Data Latih dan Data Uji

Machine learning adalah sebuah mesin yang dirancang untuk belajar. Maka dalam model *machine learning* tersebut harus ada tujuanyang harus dicapai untuk melihat apakah performa yang diberikan sudah sesuai dengan tingkat akurasi yang diinginkan. Untuk mencapai tujuantersebut, *machine learning* diberikan set data yang harus dicapai atau dilampaui, dan mana set data yang bisa digunakan untuk mencapai atau melampaui tujuan tersebut. Set data untuk data dicapai disebut data uji, sedangkan set data untuk mencapainya disebut data latih. Set data latih ini

nantinya akan digunakan untuk membuat model *machine learning*. Data latih adalah data yang sudah ada sebelumnya berdasarkan fakta yang sudah terjadi. Sedangkan set data uji akan digunakan untuk menguji performa dan kebenaran dalam model yang bersangkutan. Data uji adalah data yang sudah berkelas atau berlabel yang digunakan untuk menghitung akurasi model klasifikasi yang dibentuk.

Total data keseluruhan sebesar 498 yang terdiri dari data *tweet* dan data *retweet*. Peneliti menggunakan proporsi 80% untuk data latih dan proporsi 20% untuk data uji. Dari kedua proporsi tersebut diperoleh :

Tabel 5.9. Data Latih dan Data Uji

Data Latih (80%)	Data Uji (20%)	Jumlah
398	100	498

Berikut perhitungan untuk mencari jumlah data latih dan data uji :

Data latih = 498 x 80%
= 398

Data uji =498x 20%
= 100

Berdasarkan tabel **Tabel 5.10.** jumlah seluruh data yang dimiliki pada penelitian ini sebesar 498, dengan proporsi 80% untuk data latih yaitu sebesar 398 data, dan dengan 20% untuk data uji yaitu sebesar 100 data. Penentuan proporsi jumlah data latih dan data uji tersebut berdasarkan penelitan sebelumnya dan perbandingan proporsi terbaik.

5.7. Kinerja Metode *Naive Bayes Classifier*

Metode *Naive Bayes Classifier* digunakan untuk melakukan pengkategorian, yaitu untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berkategori positif atau negatif. Data yang sudah melalui tahap *preprocessing*dan didapat data *training* dan *testing*nya, kemudian akan melalui tahap klasifikasi menggunakan *Naive Bayes Classifier* untuk mengetahui probabilitas dari data tersebut apakah berkategori

positif atau berkategori negatif. Probabilitas tersebut diperoleh dengan persamaan (rumus sentimen). Pengukuran ketepatan klasifikasi dilakukan dengan membentuk *confusion matrix* berdasarkan hasil prediksi. Untuk menentukan kombinasi data *training* dan data *testing* yang optimal maka dilakukan dengan membandingkan tiga kombinasi yang telah ditentukan. Berikut hasil pembahasan menggunakan metoode NBC.

Tabel 5.14. Probabilitas Prior

Positif	Negatif
0,53	0,46

Berdasarkan **Tabel 5.11.** didapatkanhasil probabilitas prior untuk masing-masing kelas. Dapat dikatakan bahwa nilai prior adalah salah satu langkah mencari nilai probabilitas pada masing-masing kelas yang akan menghasilkan klasifikasi. Total masing-masing prediksi kelas dapat dilihat pada **Tabel 5.11.** Untuk probabilitas prior kelas positif sebesar 0,53 yang artinya terdapat peluang kejadian sebesar 53%. Sedangkan untuk probabilitas prior kelas negatif sebesar 0,46 yang artinya terdapat peluang kejadian sebesar 46%. Berikut persamaannya :

$$\text{Probabilitas Prior Positif} = \frac{\text{Jumlah kelas positif}}{\text{jumlah keseluruhan data}}$$

$$= \frac{265}{498} = 0{,}53$$

$$\text{Probabilitas Prior Negatif} = \frac{\text{Jumlah kelas negatif}}{\text{jumlah keseluruhan data}}$$

$$= \frac{233}{498} = 0{,}46$$

Penelitian ini menggunakan metode *confusion matrix* dalam proses evaluasi. *Confusion matrix* merupakan salah satu *tools* dalam metode evaluasi yang digunakan pada *machine learning* yang biasanya membuat dua kategori atau lebih (Manning, dkk, 2009). Dalam *confusion matrix* menunjukkan jumlah contoh data uji untuk kelas sebenarnya yang digambarkan dalam bentuk baris sedangkan kolom menggambarkan kelas yang diprediksi. Adapun hasil dari *confusion matrix* menggunakan algoritma NBC adalah sebagai berikut :

Tabel 5.15. Confusion Matrix Metode NBC

Prediksi	Aktual	
	Negatif	Positif
Negatif	19	2
Positif	27	51

Pada klasifikasi dikatakan benar jika pada variabel kelas positif dan negatif dalam aktual sama dengan prediksinya. Dari 398 data uji yang dimiliki, didapat data yang diprediksi kelas positif dan aktualnya sama sebanyak 51 data (*true positive*). Artinya terdapat 51 data yang diprediksi dengan tepat oleh mesin pembelajaran dan tidak terjadi *missclassification.* Dan data yang diprediksi kelas negatif dan aktualnya sama sebanyak 19 data (*true negative*). Sedangkan data yang diprediksi positif dan aktualnya berbeda sebanyak 27 data (*false positive*). Dan data yang diprediksi negatif dan aktualnya berbeda sebanyak 2 (*false negative*). Terdapat 29 data yang salah klasifikasi.

Recall = $\frac{TP}{TP+FN}100\% = \frac{51}{51+2}100\% = 96{,}22\%$

Precision = $\frac{TP}{TP+FP}100\% = \frac{51}{51+27}100\% = 65{,}38\%$

Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}100\% = \frac{51+19}{51+27+19+2}100\% = 70{,}71\%$

Spesificity = $\frac{TN}{TN+FP}100\% = \frac{19}{19+27}100\% = 41{,}3\%$

FPR = 1- *Spesificity*= 1 – 0,413= 0,5869

AUC = $\frac{1+recall-FPR}{2} = \frac{1+0{,}9622-0{,}5869}{2} = 0{,}687$

Berdasarkan hasil akurasi diperoleh nilai *accuracy* sebesar 70,71%, nilai *recall*96,22%, dan nilai *precision*65,38%. Berdasarkan nilai *accuracy*, *recall*, dan *precision* yang tinggi maka dapat dismpulkan bahwa proses klasifikasi sudah tepat. Akan tetapi, pada penelitian ini akan dilakukan pembandingan nilai akurasi pada metode NBC dan SVM. Maka, dipilih nilai akurasi yang besar dari kedua metode tersebut. Penentuan akurasi terbaik ditentukan berdasarkan nilai *accuracy, recall, precision* yang lebih tinggi.

5.8. Kinerja Metode *Support Vector Machine*

Metode SVM bekerja dengan cara mencari hyperplane atau garis pemisah terbaik yang memiliki margin atau jarak atar kelas terbesar menggunakan beberapa kernel dalam klasifikasi SVM antara lain *Radial Basic Function* (RBF), Linier dan Polynomial untuk memperoleh klasifikasi dengan akurasi terbaik. Berikut adalah hasil perbandingan dari kernel RBF, linier, dan polynomial.

Tabel 5.12. Model Algoritma SVM Kernel RBF

Cost	Gamma	Error
0,1	0,01	0,2145
1,0	0,01	0,2046
10,0	0,01	0,1869
0,1	0,10	0,2170
1,0	0,10	0,1871
10,0	0,10	0,1447

Berdasarkan **Tabel5.13** diperoleh nilai untuk parameter C dan gamma pada kernel RBF. C atau *cost* adalah jumlah kesalahan klasifikasi yang yang pada aktualnya benar. Sedangkan gamma adalah parameter dari kernel Gaussian untuk menangani klasifikasi nonlinier. Maka dilihat berdasrakan nilai *error* dari masing-masing parameter semakin rendah nilai *error* atau tingkat kesalahan maka semakin baik suatu model dalam melakukan klasifikasi.

Dapat dilihat pada **Tabel 5.13**parameter dengan nilai *cost*=10 dan gamma=0,1 mempunyai *error* yang paling rendah yaitu 0,1447. Maka pada kernel RBF diperoleh parameter terbaik adalah *cost*=10 dan gamma=0,10.

Tabel 5.17. Confusion Matrix Algoritma SVM Kernel RBF

Prediksi	Aktual	
	Negatif	Positif
Negatif	26	1
Positif	20	52

Pada klasifikasi dikatakan benar jika pada variabel kelas positif dan negatif dalam aktual sama dengan prediksinya. Di dapat data yang diprediksi kelas positif dan aktualnya sama sebanyak 52 data (*true positive*). Artinya terdapat 52 data yang diprediksi dengan tepat oleh mesin pembelajaran dan tidak terjadi *missclassification.* Dan data yang diprediksi kelas negatif dan aktualnya sama sebanyak 26 data (*true negative*). Sedangkan data yang diprediksi positif dan aktualnya berbeda sebanyak 20 data (*false positive*). Dan data yang diprediksi negatif dan aktualnya berbeda sebanyak 1 (*false negative*). Terdapat 21 data yang salah klasifikasi.

Recall = $\frac{TP}{TP+FN}100 = \frac{52}{52+1}100 = 98{,}11\%$

Precision = $\frac{TP}{TP+FP}100 = \frac{52}{52+20}100 = 72{,}22\%$

Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}100 = \frac{52+26}{52+20+26+1}100 = 78{,}79\%$

Spesificity = $\frac{TN}{TN+FP}100 = \frac{26}{26+20}100\% = 56{,}52\%$

FPR = 1- *Spesificity*= 1 – 0,5652= 0,4347

AUC = $\frac{1+recall-FPR}{2} = \frac{1+0{,}9811-0{,}4347}{2} = 0{,}7732$

Berdasarkan hasil akurasi diperoleh nilai *accuracy* sebesar 78,79%, nilai *recall*98,11%, nilai *precision*72,22%. Semakin besar nilai akurasi, semakin baik metode tersebut dalam melakukan klasifikasi. Dapat dilihat bahwa nilai AUC yang didapatkan sebesar 0,7731 yang artinya bahwa nilai tersebut *excellent classification* atau klasifikasi sangat baik.

Tabel 5.18. Model Algoritma SVM Kernel Linier

Cost	Error
1e-02	0,2120
1e-01	0,1996
1e+00	0,1496
1e+01	0,1496

Berdasarkan **Tabel5.15** diperoleh nilai untuk parameter C kernel linier. C atau *cost* adalah jumlah kesalahan klasifikasi yang yang pada aktualnya benar. Maka dilihat berdasarkan nilai *error* dari masing-masing parameter semakin rendah nilai *error* atau tingkat kesalahan maka semakin baik suatu model dalam melakukan klasifikasi. Dapat dilihat pada **Tabel 5.12** parameter dengan nilai cost=10 dan mempunyai *error* yang paling rendah yaitu 0,1496. Maka pada kernel linier diperoleh parameter terbaik adalah *cost*=10.

Tabel 5.19. *Confusion Matrix* Algoritma SVM Kernel Linier

Prediksi	Aktual	
	Negatif	Positif
Negatif	19	0
Positif	27	53

Di dapat data yang diprediksi kelas positif dan aktualnya sama sebanyak 53 data (*true positive*). Artinya terdapat 53 data yang diprediksi dengan tepat oleh mesin pembelajaran dan tidak terjadi *missclassification.* Dan data yang diprediksi kelas negatif dan aktualnya sama sebanyak 19 data (*true negative*). Sedangkan data yang diprediksi positif dan aktualnya berbeda sebanyak 27 data (*false positive*). Dan data yang diprediksi negatif dan aktualnya berbeda sebanyak 0 (*false negative*). Terdapat 27 data yang salah klasifikasi.

$$Recall = \frac{TP}{TP+FN}100 = \frac{56}{56+71}100 = 44{,}09\%$$

$$Precision = \frac{TP}{TP+FP}100 = \frac{56}{56+0}100 = 100\%$$

Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}100 = \frac{56+19}{56+0+19+71}100 = 85{,}66\%$

Spesificity = $\frac{TN}{TN+FP}100 = \frac{19}{19+0}100 = 100\%$

FPR = 1- *Spesificity*= 1 – 1= 0

AUC = $\frac{1+recall-FPR}{2} = \frac{1+0{,}4409-0}{2} = 0{,}72$

Berdasarkan hasil akurasi diperoleh nilai *accuracy* sebesar 85,66%, nilai *recall* 44,09%, dan nilai *precision* 100%. Semakin besar nilai akurasi semakin baik metode tersebut dalam melakukan klasifikasi. Dapat dilihat bahwa nilai AUC yang didapatkan sebesar 0,72 yang artinya bahwa nilai tersebut fair classification atau klasifikasi cukup.

Tabel 5.16. Model Algoritma SVM Kernel Polynomial

Degree	Gamma	Error
1	1e-03	0,2171
2	1e-03	0,4666
3	1e+00	0,1423
1	1e-02	0,2121
2	1e+02	0,2569
1	1e-01	0,1971

Berdasarkan **Tabel5.16** diperoleh nilai untuk parameter *degree* dan *cost* kernel polynomial. C atau *cost* adalah jumlah kesalahan klasifikasi yang yang pada aktualnya benar sedangkan d atau *degree* menunjukkan pada setiap *dataset* yang diuji, nilai parameter d akan menemukan nilai optimal pada setiap *dataset*. Dapat dilihat pada **Tabel 5.16**parameter dengan nilai *cost*=1 dan *gamma*=10 mempunyai *error* yang paling rendah yaitu 0,1423. Maka pada kernel polynomial diperoleh parameter terbaik adalah *cost*=1 dan *gamma*=10.

Tabel 5.17. *Confusion Matrix* Algoritma SVM Kernel Polynomial

Prediksi	Aktual	
	Negatif	Positif
Negatif	28	1
Positif	18	52

Di dapat data yang diprediksi kelas positif dan aktualnya sama sebanyak 97 data (*true positive*). Artinya terdapat 97 data yang diprediksi dengan tepat oleh mesin pembelajaran dan tidak terjadi *missclassification.* Dan data yang diprediksi kelas negatif dan aktualnya sama sebanyak 28 data (*true negative*). Sedangkan data yang diprediksi positif dan aktualnya berbeda sebanyak 18 data (*false positive*). Dan data yang diprediksi negatif dan aktualnya berbeda sebanyak 1 (*false negative*). Terdapat 19 data yang salah klasifikasi.

Recall = $\frac{TP}{TP+FN}100 = \frac{52}{52+1}100\% = 98{,}11\%$

Precision = $\frac{TP}{TP+FP}100 = \frac{52}{52+18}100\% = 74{,}28\%$

Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}100 = \frac{52+28}{52+18+28+1}100\% = 80{,}81\%$

Spesificity = $\frac{TN}{TN+FP}100 = \frac{28}{28+18}100\% = 60{,}87\%$

FPR = 1- *Spesificity*= 1 –0,6087= 0,3913

AUC = $\frac{1+recall-FPR}{2} = \frac{1+0{,}9811-0{,}3913}{2} = 0{,}79$

Berdasarkan hasil akurasi diperoleh nilai *accuracy* sebesar 80,81%, nilai *recall*98,11%, nilai *precision*74,28%. Semakin besar nilai akurasi semakin baik metode tersebut dalam melakukan klasifikasi. Dapat dilihat bahwa nilai AUC yang didapatkan sebesar 0,79 yang artinya bahwa nilai tersebut *good classification* atau klasifikasi baik.

5.9. Perbandingan Hasil Metode NBC dan SVM

Perbandingan hasil metode NBC dan SVM pada penelitian ini digunakan untuk menentukan metode mana yang terbaik dalam melakukan klasifikasi. Dalam

menentukan metode yang terbaik dilihat berdasarkan akurasinya yang tertinggi. Hasil akurasi kedua metode dapat dilihat sebagai berikut

Tabel 5.18. Hasil Perbandingan

	NBC	SVM RBF	SVM Linier	SVM Polynomial
Akurasi	70,71%	78,79%	72,73%	80,81%

Berdasarkan **Tabel 5.18.** dapat dilihatbahwa hasil akurasi tertinggi adalah metode SVM dengan kernel Polinomial yaitu sebesar 80,81%. Oleh karena itu, dapat ditarik kesimpulan bahwa metode SVM dengan kernel Polinomial merupakan metode yang paling baik digunakan untuk klasifikasi katapada data tersebut.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan dalam bab sebelumnya, didapat kesimpulan guna menjawab rumusan masalah sebagai berikut:

1. Dari penelitian yang telah dilakukan maka didapat kesimpulan bahwa penelitian ini menggunakan lima *hashtag* yang berkaitan mengenai kesehatan mental selama pandemi Covid-19 dari sosial media Twitter dengan proses *web crawling* sebanyak 498*tweet* menunjukkan 265*tweet* diklasifikasikan sebagai kelas positif, sedangkan 233*tweet* diklasifikasikan sebagai kelas negative dengan kata yang paling banyak muncul yaitu "kesehatan" dengan frekuensi kemunculan sebanyak 152, "mental" dengan frekuensi kemunculan sebanyak 134, "tenaga*"* dengan frekuensi kemunculan sebanyak 74, "burnout" dengan frekuensi kemunculan sebanyak 68, "Indonesia" dengan frekuensi kemunculan sebanyak 68, "kalangan" dengan frekuensi kemunculan sebanyak 68, "keletihan" dengan frekuensi kemunculan sebanyak 68, "mayoritas" dengan frekuensi kemunculan sebanyak 67, "lingkungan" dengan frekuensi kemunculan sebanyak 61, dan "gaji" dengan frekuensi kemunculan sebanyak 59. Yang artinya dari data tweet yang di dapat dapat disimpulkan bahwa kesehatan mental selama pandemi mempunyai peran penting selama masa pandemi terutama tenaga kesehatan di Indonesia yang banyak mengalami *burnout*atau keletihan fisik akibat bertambahnya jumlah pasien baik yang dalam pengawasan atau sudah positif Covid-19 setiap harinya.
2. Dari hasil perbandingan antara metode NBC dengan metode SVM, didapatkan metode SVM dengan kernel Polinomialmerupakan metode yang paling baik digunakan untuk klasifikasi kata pada data tersebut dengan hasil *accuracy* sebesar 80,81%, recall 98,11%, nilai precision 74,81%, dan nilai

AUC sebesar 0,79. Artinya terdapat 97 data yang diprediksi dengan tepat oleh mesin pembelajaran dan tidak terjadi *missclassification.* Dan data yang diprediksi kelas negatif dan aktualnya sama sebanyak 28 data (*true negative*). Sedangkan data yang diprediksi positif dan aktualnya berbeda sebanyak 18 data (*false positive*). Dan data yang diprediksi negatif dan aktualnya berbeda sebanyak 1 (*false negative*). Terdapat 19 data yang salah klasifikasi. Dapat dilihat juga bahwa nilai AUC yang didapatkan sebesar 0,79 yang artinya bahwa nilai tersebut termasuk *excellent classification* atau klasifikasi sangat baik. Sedangkan hasil akurasi untuk metode lain yaitu metode SVM dengan kernel RBF sebesar 78,79%, SVM dengan kernel Linier sebesar 71,73% dan NBC sebesar 70,71%.

6.2 Saran

Berdasarkan hasil analisis dan kesimpulan, peneliti dapat memberi saran sebagai berikut:

1. Untuk penelitian selanjutnya, dalam memperoleh data dapat memperluas periode waktu agar data yang didapat lebih bervariasi dan menggunakan metode klasifikasi yang lain, sehingga diperoleh hasil klasifikasi yang lebih spesifik,
2. Akibat adanya kemiripan antara bahasa Indonesia dan Melayu maka pada saat proses *crawling* terdapat *tweet* yang bukan berasal dari pengguna Twitter Indonesia yang ikut terambil, sehingga diharapkan pada penelitian selanjutnya diharapkan dapat bekerja sama dengan ahli bahasa sehingga memberikan hasil yang lebih baik.
3. Selain itu beberapa pengguna Twitter menggunakan kata tidak baku yang terkadang memiliki dual makna sehingga membuat rancu hasil sentimen.

DAFTAR PUSTAKA

Agrawal, A., (2016). Clickbait Detection Using Deep Learning. *2nd International Conference on Next Generation Computing echnologies NCGT*. hal. 268-272. IEEE.

Alpaydin, E., (2010). Introduction to Machine Learning Second Edition. London: MIT Press.

American Psychiatric Association (2000). *Diagnostic and Statistical Manual of Mental Disorders Fifth Edition.* Washington: APA Publishing .

Baeza-Yates, R., & Ribier-Neto, B. (1999). *Modern Information Retrieval*. New York: Addison-Wesley.

Bramer, M. (2007). *Principles of Data Mining: Undergraduate Topics in Computer Science*. London: Springer-Verlag.

Bustami. (2013). Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi,TECHSI *:* Jurnal Penelitian Teknik Informatika,Vol. 3, No.2 (hal. 127-146).

Buntoro, Ghulam Asrofi. (2016). Analisis Sentimen Hatespeech pada Twitter dengan Metode Naive Bates Classifierdan Support Vector Machine.Jurnal Dinamika InformatikaV*ol. 5 No. 2*, ISSN 1978-1660.

Carr, Caleb T & Hayes, Rebecca A. (2015). Social Media: Defining, Developing, and Divining,Atlantic Journal of Communication 23, no.1.hal. 46-45.

Christianini, N. And Shawe Taylor, J. (2000). An Introduction to Support Vector Machine and other Kernel Based Learning methods, Cambridge University Press.

Cimsa. (2016). *A Life With The Black Dog: Depression*Diambil kembali dari: HYPERLINK "https://cimsa.fk.ugm.ac.id/2016/10/15/a-life-with-the-black-dog-depression/" https://cimsa.fk.ugm.ac.id/2016/10/15/a-life-with-the-black-dog-depression/

Evan, F. H., Pranowo, & Purnomo, S. Y. (2014).Pembangunan Perangkat Lunak Peringkas Dokumen dari Banyak Sumber Menggunakan Sentence Scoring dengan Metode TF-IDF. *Seminar Nasional Aplikasi Teknologi Informasi*(hal. 17-22)

Faradhillah, Nuke Y. A dkk (2016). Eksperimen Sistem Klasifikasi Analisa Sentimen Twitter pada Akun Resmi Pemerintah Kota Surabaya Berbasis Pembelajaran Mesin. *S*eminar Nasional Sistem Informasi Indonesia (SESINDO).

Fayyad, Usama. (1996). *Advances in Knowledge Discovery and Data Mining*. London: MIT Press.

Feldman, R., & Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* New York: Cambridge

Foley, Paul. (2014). *'Black Dog' as a Metaphor for Depression: a Brief History*. Diambil kembali dari Alienson: HYPERLINK "http://alienson.com/files/Black-dog-as-a-metaphor-for-depression_a-brief-history_by-Paul-Foley.pdf" http://alienson.com/files/Black-dog-as-a-metaphor-for-depression_a-brief-history_by-Paul-Foley.pdf

Gorunescu, F. (2011). Data Mining Concepts, Models and Techniques. Verlag Berlin Heidelberg: Springer.

Han, J., dan Kamber, M. (2006). Data Mining : Concepts and Techniques Second Eddition. USA: Elsevier.

Han, J., Kamber, M., & Pei, J. (2011). Data Mining Concepts and Techniques Third Edition. Waltham: Elsevier Inc.

Handayani, F., dan Pribadi, F. S. (2015). Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat Melalui Layanan *Call Center* 110. *Jurnal Teknik Elektro, 7(1),* (hal.19–24).

Harrington, Peter. (2012). Machine Learning in Action. New York: Manning.

Hotsuite (We Are Social). (2019). *Berapa Pengguna Media Sosial Indonesia?* Diambil kembali dari HYPERLINK "https://databoks.katadata.co.id/datapublish/2019/02/08/berapa-pengguna-

media-sosial-indonesia" https://databoks.katadata.co.id/datapublish/2019/02/08/berapa-pengguna-media-sosial-indonesia.

Johns Hopkins University. COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE). https://coronavirus.jhu.edu/map.html (diakses pada 1 September 2020).

Jonathan, S. (2006). *Metode Penelitian Kuantitatif dan Kualitatif*. Yogyakarta :Graha Ilmu.

Katadata (2019). *Survei APJII: Penetrasi Pengguna Internet di Indonesia* Diambil kembali dari HYPERLINK "https://katadata.co.id/berita/2019/05/16/survei-apjii-penetrasi-pengguna-internet-di-indonesia-capai-648" https://katadata.co.id/berita/2019/05/16/survei-apjii-penetrasi-pengguna-internet-di-indonesia-capai-648.

Keltner, N. L., Schwecke, L. H., & Bostrom, C. E. (1999). *Psychiatric Nursing.* Philippines*:* Mosby Inc.

Kominfo (2019). *Pelaporan Akun Negatif di Media Sosial* Diambil kembali darihttps://kominfo.go.id/content/detail/15852/siaran-pers-no-08hmkominfo012019-tentang-pelaporan-konten-negatif-di-media-sosial/0/siaran_pers

Kramer, Adam.D.I., Guillory, J.E., & Hancock. (2014). Experimental Evidence of Mssive-Scale Emotional Contagion Through Social Networks. *Proceedings of the national Academy of Sciences of the United States of America*. (hal. 29)

Kurniawan Sidiq, Hadi., Sulistyo Kusumo, Dana., & Lukmana Sardi, Indra. (2019). Mendeteksi *Cyberhate* pada Twitter Menggunakan Text

Kusrini,& Luthfi, E. (2009).*Algoritma Data Mining*. Yogyakarta: Andi Offset.

Larose, D. T., (2006). Data Mining Methods And Models, John Wiley & Sons, Inc., Hoboken, New Jersey

Lesmana, I. G. N. A. (2012). *Analisis Pengaruh media Sosial Twitter terhadap Pembentukan Brand Attachment (Study: PT. XL AXIATA).* Tesis. Pascasarjana Universitas Indonesia.

Liu, Bing. (2010). Sentiment Analysis and Subjectivity. Chicago: University of Illinois.

Liu, Bing. (2012). Sentiment Analysis and Opinion Mining. Chicago: University of Illinois

Luqyana, W. A., Cholissodin, I., & Perdana, R. S. (2018). Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 4704-4713.

Mahardhika, Y. S, & Zuliarso, Eri. (2018). Analisis Sentimen terhadap Pemerintahan Joko Widodo pada Media Sosial Twitter Menggunakan Algoritma *Naive Bayes Classifier*. Prosiding SINTAK 2018. ISBN: 978-602-8557-20-7.

Mudjiono, Yoyon. (2015). *Ilmu Komunikasi*. Surabaya: Jaudar Press

Nasrullah, Rulli. (2015). *Media Sosial Perspektif Komunikasi, Budaya, dan Sosioteknologi.* Bandung: Simbiosa Rekatama Media.

Noted. (2018). *Where the Black Dog Metaphor for Depression Comes From* Diambil kembali dari Noted: HYPERLINK "https://www.noted.co.nz/health/health-psychology/black-dog-where-depression-metaphor-comes-from" https://www.noted.co.nz/health/health-psychology/black-dog-where-depression-metaphor-comes-from.

Novantirani, Anita (2015). Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine. eProceedings of Engineering: Vol.2. ISSN: 2355-936.

Osuna E, Freud R, Girosi F. (1997). An Improved Training Algorithm for Support Machine. *Proc. of the 1997 IEEE Workshop Neural Networks for Signal Processing VII: Amelia Island, 24-26 September 1997.* Amelia Island: IEEE Computer Society (hal.276-285).

Prasetyo, E. (2012). *Data Mining Konsep dan Aplikasi menggunakan MATLAB.* Yogyakarta: ANDI.

Prima, Dewi., Ajeng & Delliana, Santi. (2019). *Self Disclosure Generasi Z di Twitter*. e-ISSN : 2656-050X.

Putri, W. S. R., dkk. (2016). 7 Pengaruh Media Sosial Terhadap Perilaku Remaja. *Prosiding Ks: riset & PKM,* 3(1), (hal. 1-154)

Safina, Nabila. (2018). Analisis Sentimen pada Twitter Terhadap Jasa Transportasi Online di Indonesia dengan Metode Support Vector Machine. Universitas Dian Nuswantoro

Santosa, A. P. (2016). Naive Bayes Classification pada Klasifikasi Dokumen untuk Identifikasi Konten E-Goverment. *Applied Intelligent System*, 48-55.

Santosa, Budi. (2010). Tutorial Support Vector Machine. Teknik Industri, ITS.[Online]. Tersedia: http://www. google. co. id/url.

Sembiring, K. (2007). Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan. Bandung: Institut teknologi Bandung.

Soetjiningsih. (2007). *Tumbuh Kembang Remaja dan Permasalahannya.* Jakarta: CV. Sagung Seto.

Suryono, Sigit, E. U. (2018). Klasifikasi Sentimen pada *Twitter* dengan Naive Bayes Classifier. *Jurnal Ilmiah Bidang Teknologi*, 89-86.

S., Furqon, Aprilia M., & Fauzi, M. (2008).*Klasifikasi Penyakit Skizofrenia dan Episode Depresi Pada Gangguan Kejiwaan Dengan Menggunakan Metode Support Vector Machine (SVM)*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 11, p. 5611-5618, juli 2018. ISSN 2548-964X

Triawati, Candra., dkk. (2009). *Pemodelan Berbasis Konsep untuk Kategorisasi Artikel Berita Berbahasa Indonesia.* Seminar Nasional Apilasi teknologi Informasi 2009. ISSN: 1 907-5022.

Vapnik, V., & Cortes, C. (1995). Support Vector Networks. Machine Learning, 273-297.

WHO. (2017) Depression and Other Common Mental Disorders (Global Health Estimates). (WHO/MSD/MER/2017.2) (hal. 5)

Zarella, Dan. (2010). *The Social Media Marketing Book*. Jakarta: PT. Serambi Ilmu Semesta.

LAMPIRAN

Lampiran 1 Proses *Preprocessing*

```
#web scraping twitter
library(tm)
library(wordcloud2)
library(twitteR)
library(rtweet)

# Ganti Sesuai dengan Key Milik Kita
consumer_key <- "...."
consumer_secret <- "..."
access_token    <- "..."
access_secret   <- "..."
setup_twitter_oauth(consumer_key,  consumer_secret,  access_token,
access_secret)

#mengambil data dari twitter
data2bd = searchTwitter('black + dog',
                        n = 1000, lang = 'id',
       retryOnRateLimit = 10e3)
#tw <- searchTwitter('mental+illness', lang="id", n=100,resultType
= "recent")
#tw <- sapply(minningtweets,function(x) x$getText())
#str(tw)

#simpan datanya
saveRDS(data2bd,file = 'tweet-data2.rds')

#Load dataset
data2bd <- readRDS('tweet-data2.rds')
bd2 = twListToDF(data2bd)
View(bd2)

#asosiasi kata
komen <- bd2$text
komenc <- Corpus(VectorSource(komen))

##Cleaning data
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
twitclean <- tm_map(komenc, removeURL)
removeNL <- function(y) gsub("\n", " ", y)
twitclean <- tm_map(twitclean, removeNL)
replacecomma <- function(y) gsub(",", "", y)
twitclean <- tm_map(twitclean, replacecomma)
removeRT <- function(y) gsub("RT ", "", y)
twitclean <- tm_map(twitclean, removeRT)
removetitik2 <- function(y) gsub(":", "", y)
twitclean <- tm_map(twitclean, removetitik2)
removetitikkoma <- function(y) gsub(";", " ", y)
twitclean <- tm_map(twitclean, removetitikkoma)
removeamp <- function(y) gsub("&amp;", "", y)
twitclean <- tm_map(twitclean, removeamp)
```

```
removeUN <- function(z) gsub("@\\w+", "", z)
twitclean <- tm_map(twitclean, removeUN)
remove.all <- function(xy) gsub("[^[:alpha:][:space:]]*", "", xy)
twitclean <- tm_map(twitclean,remove.all)

#menghapus extra white space dan angka
twitclean<-tm_map(twitclean, stripWhitespace)
twitclean <- tm_map(twitclean, removeNumbers)
twitcleanCopy<-twitclean

#tahap case folding & tokenizing
#menghapus titik koma, menjadi non kapital
twitclean <- tm_map(twitclean, removePunctuation)
twitclean <- tm_map(twitclean, tolower)

#Menghapus stopword #disimpen di work directory tahap filtering &
stemming
myStopwords = readLines('E://stopwordku.csv')
twitclean <- tm_map(twitclean,removeWords,myStopwords)

#Build a term-document matrix
{
  dtm <- TermDocumentMatrix(twitclean)
  m <- as.matrix(dtm)
  v <- sort(rowSums(m),decreasing=TRUE)
  bd2 <- data.frame(word = names(v),freq=v)
  head(bd2, 10)
}

#membuat wordcloud pt 1
wordcloud2(bd2,shape = "cloud",
           backgroundColor = "black",
           color = 'random-light' ,
           size = 0.5)

#membuat asosiasi kata
v<-as.list(findAssocs(dtm,
                      terms= c('depresi),
                      corlimit=
                        c(0.50,0.3)))
v

## save data
dataframe<-data.frame(text=unlist(sapply(twitclean,`[`)),
stringsAsFactors=F)
View(dataframe)
write.csv(dataframe,file = 'twitbismillah.csv')
```

Lampiran 2 Proses Pelabelan Kata

```
## ambil data yang sudah di cleaning di tahap pertama
data_clean <- read.csv('twitbismillah.csv')
str(data_clean)
View(data_clean)

## ambil data untuk word scoring
positif        <-        scan("E://Kuliah//Kumpulan        stop
word//positive_word.txt",what="character",comment.char=";")
negatif        <-        scan("E://Kuliah//Kumpulan        stop
word//negative_word.txt",what="character",comment.char=";")

## fungsi untuk melnalankan penilaian atau pembobotan tehadap kata-
kata
score.sentiment    =    function(kalimat2,    positif,    negatif,
.progress='none')
{
  require(plyr)
  require(stringr)
  scores = laply(kalimat2, function(kalimat, positif, negatif) {
    kalimat = gsub('[[:punct:]]', '', kalimat)
    kalimat = gsub('[[:cntrl:]]', '', kalimat)
    kalimat = gsub('\\d+', '', kalimat)
    kalimat = tolower(kalimat)

    list.kata = str_split(kalimat, '\\s+')
    kata2 = unlist(list.kata)
    positif.matches = match(kata2, positif)
    negatif.matches = match(kata2, negatif)
    positif.matches = !is.na(positif.matches)
    negatif.matches = !is.na(negatif.matches)
    score = sum(positif.matches) - (sum(negatif.matches))
    return(score)
  }, positif, negatif, .progress=.progress )
  scores.df = data.frame(score=scores, text=kalimat2)
  return(scores.df)
}

#melakukan skoring text
hasil = score.sentiment(data_clean$text, positif, negatif)
head(hasil)
# melakukan labeling pada nilai yang kurang dari 0 sebagai negatif
dan lebih dari = 0 adalah positif
hasil$klasifikasi<- ifelse(hasil$score<0, "Negatif","Positif")
hasil$klasifikasi
View(hasil)

#Tukar Row
datascoring <- hasil[c(3,1,2)]
View(datascoring)
write.csv(datascoring, file = "data bismillah.csv")

#Memisahkan dan menyimpan twit positif dan negatif
data.pos <- datascoring[datascoring$score>0,]
View(data.pos)
write.csv(data.pos, file = "data-bdpositif.csv")
```

```
#Memisahkan dan meyimpan twit positif dan negatif
data.neg <- datascoring[datascoring$score<0,]
View(data.neg)
write.csv(data.neg, file = "data-bdnegatif.csv")
```

Lampiran 3 Proses Klasifikasi Naive Bayes

```
# Load required libraries
library(tm)
library(e1071)
library(dplyr)
library(caret)
library(DMwR)
library(ggplot2)
library(ROSE)
library(klaR)
library(MASS)
library(pROC)
library(sos)
library(brew)

#Input Data
bismillah=  read.csv("E:\\Kuliah\\skrpsweet\\data  bismillah.csv",
header = TRUE, sep = ";")
glimpse(bismillah)
View(bismillah)

#melakukan pembagian data training dan testing
splitIndex   <- createDataPartition (bismillah$class, p = 0.80 ,
list = FALSE)
traindata  <-reviews[ splitIndex, ]
testdata  <-reviews[-splitIndex, ]
traindata
testdata
dim(traindata)
dim(testdata)
yaAllah$class = as.factor(reviews$klasifikasi)
yaAllah$class


#==========Naive Bayes============#
set.seed(42)
modelNVB <- naiveBayes(traindata, traindata$class, laplace = 1)
class(modelNVB)
summary(modelNVB)
print(modelNVB)
confusionMatrix(predict(modelNVB,   traindata),   traindata$class,
positive = 'Positif')
preds <- predict(modelNVB, newdata = testdata)
conf_matrix <- table(preds, testdata$class)
conf_matrix
confusionMatrix(predict(modelNVB,    testdata),    testdata$class,
positive = 'Positif')
```

Lampiran 4 Proses Klasifikasi SVM

```
#membangun model SVM#
set.seed(42)
modelsvm <- svm(klasifikasi ~ ., data = traindata)
class(modelsvm)
summary(modelsvm)
print(modelsvm)

preds <- predict(modelsvm, newdata = testdata)
conf_matrix <- table(preds, testdata$klasifikasi)
conf_matrix
confusionMatrix(predict(modelsvm, testdata), testdata$klasifikasi,
positive = 'Positif')

svm_tune <- tune(svm, klasifikasi~. , data = traindata,
kernel="radial",types = "C-clasification",
                 ranges= list( cost = c(0.1, 1, 10),gamma=c(0.01,
0.1, 1,5,10)))
print(svm_tune)
summary(svm_tune)
modelsvm <- svm(klasifikasi~ ., data = traindata, method =
"svmRadial", ranges= cost(10), gamma= 0.01)
class(modelsvm)
summary(modelsvm)
print(modelsvm)
#confusionMatrix(predict(modelsvm,                    traindata),
traindata$klasifikasi, positive = 'Positif')
#preds <- predict(modelsvm, newdata = testdata)
#conf_matrix <- table(preds, testdata$klasifikasi)
#conf_matrix
#confusionMatrix(predict(modelsvm,                     testdata),
testdata$klasifikasi, positive = 'Positif')

#kernel linear
linear <- tune(svm, klasifikasi~. , data = traindata,
kernel="linear",types = "C-clasification",ranges= list( cost =
c(0.01, 0.1, 1, 10, 100)))
print(linear)
summary(linear)
modelsvmlinear <- svm(klasifikasi~ ., data = traindata, method =
"svmlinaer", ranges= cost(0.01))
class(modelsvmlinear)
summary(modelsvmlinear)
print(modelsvmlinear)
confusionMatrix(predict(modelsvmlinear,                traindata),
traindata$klasifikasi, positive = 'Positif')

preds <- predict(modelsvmlinear, newdata = testdata)
conf_matrix <- table(preds, testdata$klasifikasi)
conf_matrix
confusionMatrix(predict(modelsvmlinear,                 testdata),
testdata$klasifikasi, positive = 'Positif')
```

```
#polynom
polynom <- tune(svm, klasifikasi~. , data = traindata,
kernel="polynomial",types = "C-clasification",ranges= list( degree
= c(1,2,3), gamma=c(0.001, 0.01, 0.1, 1,5,10)))
print(polynom)
summary(polynom)
modelsvmpoly <- svm(klasifikasi~ ., data = traindata, method =
"svmpolynomial", degree=1, gamma=0.1)
class(modelsvmpoly)
summary(modelsvmpoly)
print(modelsvmpoly)
confusionMatrix(predict(modelsvmpoly, traindata),
traindata$klasifikasi, positive = 'Anjuran')
preds <- predict(modelsvmpoly, newdata = testdata)
conf_matrix <- table(preds, testdata$klasifikasi)
conf_matrix
confusionMatrix(predict(modelsvmpoly, testdata),
testdata$klasifikasi, positive = 'Anjuran')
```

Lampiran 5 Output R

Perbandingan 7:3

```
> A=nrow(traindata)
> A
[1] 1736
> testdata =yaAllah [-index,]
> B=nrow(testdata)
> B
[1] 744
```

Svm kernel radial

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
   10   0.1

- best performance: 0.07199522

- Detailed performance results:
   cost gamma      error dispersion
1   0.1  0.01 0.24307355 0.04089971
2   1.0  0.01 0.19123646 0.02971927
3  10.0  0.01 0.13016743 0.02607458
4   0.1  0.10 0.19123646 0.02971927
5   1.0  0.10 0.13016743 0.02607458
6  10.0  0.10 0.07199522 0.02300486
7   0.1  1.00 0.14170487 0.02844290
8   1.0  1.00 0.07199522 0.02300486
9  10.0  1.00 0.07199522 0.02300486
10  0.1  5.00 0.14170487 0.02844290
11  1.0  5.00 0.07199522 0.02300486
12 10.0  5.00 0.07199522 0.02300486
13  0.1 10.00 0.14170487 0.02844290
14  1.0 10.00 0.07199522 0.02300486
15 10.0 10.00 0.07199522 0.02300486
```

```
> conf_matrix

preds     Negatif Positif
  Negatif     529     101
  Positif       0     114
> confusionMatrix(predict(modelsvm, testdata
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     529     101
   Positif       0     114

               Accuracy : 0.8642
                 95% CI : (0.8375, 0.888)
    No Information Rate : 0.711
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6161

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5302
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8397
             Prevalence : 0.2890
         Detection Rate : 0.1532
   Detection Prevalence : 0.1532
      Balanced Accuracy : 0.7651

       'Positive' Class : Positif
```

SVM kernel linier

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
    1

- best performance: 0.07313468

- Detailed performance results:
   cost      error dispersion
1 1e-02 0.24309348 0.03560877
2 1e-01 0.14171484 0.01911776
3 1e+00 0.07313468 0.01641342
4 1e+01 0.07313468 0.01641342
5 1e+02 0.07313468 0.01641342
```

```
> confusionMatrix(predict(modelsvmlinear, test
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     529     215
   Positif       0       0

               Accuracy : 0.711
                 95% CI : (0.677, 0.7434)
    No Information Rate : 0.711
    P-Value [Acc > NIR] : 0.5184

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.000
            Specificity : 1.000
         Pos Pred Value :   NaN
         Neg Pred Value : 0.711
             Prevalence : 0.289
         Detection Rate : 0.000
   Detection Prevalence : 0.000
      Balanced Accuracy : 0.500

       'Positive' Class : Positif
```

SVM kernel polinom

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 degree gamma
      1     1

- best performance: 0.07545678

- Detailed performance results:
   degree gamma      error dispersion
1       1 1e-03 0.24308352 0.02702568
2       2 1e-03 0.24308352 0.02702568
3       3 1e-03 0.24308352 0.02702568
4       1 1e-02 0.24308352 0.02702568
5       2 1e-02 0.24308352 0.02702568
6       3 1e-02 0.24308352 0.02702568
7       1 1e-01 0.14172480 0.01802042
8       2 1e-01 0.24308352 0.02702568
9       3 1e-01 0.24308352 0.02702568
10      1 1e+00 0.07545678 0.01364579
11      2 1e+00 0.07545678 0.01364579
12      3 1e+00 0.07545678 0.01364579
13      1 5e+00 0.07545678 0.01364579
14      2 5e+00 0.07545678 0.01364579
15      3 5e+00 0.07545678 0.01364579
16      1 1e+01 0.07545678 0.01364579
17      2 1e+01 0.07545678 0.01364579
18      3 1e+01 0.07545678 0.01364579
```

```
> confusionMatrix(predict(modelsvmpoly, testda
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif      529      47
   Positif        0     168

               Accuracy : 0.9368
                 95% CI : (0.9169, 0.9532)
    No Information Rate : 0.711
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8356

 Mcnemar's Test P-Value : 1.949e-11

            Sensitivity : 0.7814
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9184
             Prevalence : 0.2890
         Detection Rate : 0.2258
   Detection Prevalence : 0.2258
      Balanced Accuracy : 0.8907

       'Positive' Class : Positif
```

Naive Bayes

```
> confusionMatrix(predict(modelNVB, testdata), testdata$class, positive = 'Positif')
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif      528       6
   Positif       24     185

               Accuracy : 0.9596
                 95% CI : (0.9429, 0.9726)
    No Information Rate : 0.7429
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8975

 Mcnemar's Test P-Value : 0.001911

            Sensitivity : 0.9686
            Specificity : 0.9565
         Pos Pred Value : 0.8852
         Neg Pred Value : 0.9888
             Prevalence : 0.2571
         Detection Rate : 0.2490
   Detection Prevalence : 0.2813
      Balanced Accuracy : 0.9626

       'Positive' Class : Positif
```

Lampiran 6 Output R

Perbandingan 8:2

```
> dim(train)
[1] 1984    2
> dim(test)
[1] 496   2
```

Svm kernel radial

```
- best parameters:
 cost gamma
   10   0.1

- best performance: 0.04383026

- Detailed performance results:
   cost gamma      error dispersion
1   0.1  0.01 0.12393280 0.02402920
2   1.0  0.01 0.09421349 0.02143071
3  10.0  0.01 0.08010507 0.02286943
4   0.1  0.10 0.10731435 0.02664076
5   1.0  0.10 0.08010507 0.02237330
6  10.0  0.10 0.04383026 0.01700049
7   0.1  1.00 0.12947312 0.03677808
8   1.0  1.00 0.05239836 0.01893224
9  10.0  1.00 0.05139079 0.01857008
10  0.1  5.00 0.12997817 0.03583703
11  1.0  5.00 0.05945383 0.02192672
12 10.0  5.00 0.05945383 0.02192672
13  0.1 10.00 0.12997817 0.03583703
14  1.0 10.00 0.05945383 0.02192672
15 10.0 10.00 0.05945383 0.02192672
```

```
Prediction Negatif Positif
   Negatif     366      44
   Positif       2      83

               Accuracy : 0.9071
                 95% CI : (0.878, 0.9312)
    No Information Rate : 0.7434
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7268

 Mcnemar's Test P-Value : 1.493e-09

            Sensitivity : 0.6535
            Specificity : 0.9946
         Pos Pred Value : 0.9765
         Neg Pred Value : 0.8927
             Prevalence : 0.2566
         Detection Rate : 0.1677
   Detection Prevalence : 0.1717
      Balanced Accuracy : 0.8241

       'Positive' Class : Positif
```

Svm kernel linier

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
   10

- best performance: 0.04735039

- Detailed performance results:
   cost      error  dispersion
1 1e-02 0.09722349 0.019554544
2 1e-01 0.08059236 0.014747986
3 1e+00 0.04835541 0.009800599
4 1e+01 0.04735039 0.010095461
5 1e+02 0.04735039 0.010095461
```

```
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     368      71
   Positif       0      56

               Accuracy : 0.8566
                 95% CI : (0.8225, 0.8862)
    No Information Rate : 0.7434
    P-Value [Acc > NIR] : 6.871e-10

                  Kappa : 0.5398

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4409
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8383
             Prevalence : 0.2566
         Detection Rate : 0.1131
   Detection Prevalence : 0.1131
      Balanced Accuracy : 0.7205

       'Positive' Class : Positif
```

SVM kernel polynomial

```
- best parameters:
 degree cost
      1   10

- best performance: 0.09574895

- Detailed performance results:
   degree  cost      error dispersion
1       1 1e-03 0.25693112 0.01307557
2       2 1e-03 0.25693112 0.01307557
3       3 1e-03 0.25693112 0.01307557
4       1 1e-02 0.25693112 0.01307557
5       2 1e-02 0.25693112 0.01307557
6       3 1e-02 0.25693112 0.01307557
7       1 1e-01 0.19599767 0.01753915
8       2 1e-01 0.25693112 0.01307557
9       3 1e-01 0.25693112 0.01307557
10      1 1e+00 0.13403381 0.02487718
11      2 1e+00 0.19599767 0.01753915
12      3 1e+00 0.25693112 0.01307557
13      1 5e+00 0.09977159 0.02107993
14      2 5e+00 0.19549515 0.01792735
15      3 5e+00 0.19599767 0.01753915
16      1 1e+01 0.09574895 0.02317897
17      2 1e+01 0.17482361 0.01553859
18      3 1e+01 0.19599767 0.01753915
```

```
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     368       30
   Positif       0       97

               Accuracy : 0.9394
                 95% CI : (0.9146, 0.9587)
    No Information Rate : 0.7434
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8278

 Mcnemar's Test P-Value : 1.192e-07

            Sensitivity : 0.7638
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9246
             Prevalence : 0.2566
         Detection Rate : 0.1960
   Detection Prevalence : 0.1960
      Balanced Accuracy : 0.8819

       'Positive' Class : Positif
```

Naive Bayes

```
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     359       6
   Positif       9     121

               Accuracy : 0.9697
                 95% CI : (0.9505, 0.9829)
    No Information Rate : 0.7434
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9212

 Mcnemar's Test P-Value : 0.6056

            Sensitivity : 0.9528
            Specificity : 0.9755
         Pos Pred Value : 0.9308
         Neg Pred Value : 0.9836
             Prevalence : 0.2566
         Detection Rate : 0.2444
   Detection Prevalence : 0.2626
      Balanced Accuracy : 0.9641

       'Positive' Class : Positif
```

Lampiran 7 Output R

Perbandingan 9:1

```
> A=nrow(traindata)
> A
[1] 2232
> testdata =yaAllah [-index,]
> B=nrow(testdata)
> B
[1] 248
```

SVM kernel radial

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
   10   0.1

- best performance: 0.05867433

- Detailed performance results:
   cost gamma      error dispersion
1   0.1  0.01 0.25313501 0.03011168
2   1.0  0.01 0.19533953 0.02316310
3  10.0  0.01 0.12319827 0.01473072
4   0.1  0.10 0.19533953 0.02316310
5   1.0  0.10 0.12409513 0.01515737
6  10.0  0.10 0.05867433 0.01519353
7   0.1  1.00 0.12812900 0.01348234
8   1.0  1.00 0.05867433 0.01519353
9  10.0  1.00 0.05867433 0.01519353
10  0.1  5.00 0.12812900 0.01348234
11  1.0  5.00 0.05867433 0.01519353
12 10.0  5.00 0.05867433 0.01519353
13  0.1 10.00 0.12812900 0.01348234
14  1.0 10.00 0.05867433 0.01519353
15 10.0 10.00 0.05867433 0.01519353
```

```
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     176      39
   Positif       0      33

               Accuracy : 0.8427
                 95% CI : (0.7914, 0.8857)
    No Information Rate : 0.7097
    P-Value [Acc > NIR] : 7.623e-07

                  Kappa : 0.5457

 Mcnemar's Test P-Value : 1.166e-09

            Sensitivity : 0.4583
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8186
             Prevalence : 0.2903
         Detection Rate : 0.1331
   Detection Prevalence : 0.1331
      Balanced Accuracy : 0.7292

       'Positive' Class : Positif
```

SVM kernel linier

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
    1

- best performance: 0.06398914

- Detailed performance results:
   cost      error dispersion
1 1e-02 0.20662911 0.04151834
2 1e-01 0.13201868 0.02811656
3 1e+00 0.06398914 0.01530659
4 1e+01 0.06398914 0.01530659
5 1e+02 0.06398914 0.01530659
```

```
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     350     146
   Positif       0       0

               Accuracy : 0.7056
                 95% CI : (0.6634, 0.7454)
    No Information Rate : 0.7056
    P-Value [Acc > NIR] : 0.5223

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.7056
             Prevalence : 0.2944
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : Positif
```

SVM kernel polinomial

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 degree gamma
      1     1

- best performance: 0.0595872

- Detailed performance results:
   degree gamma      error dispersion
1       1 1e-03 0.2531550 0.02846364
2       2 1e-03 0.2531550 0.02846364
3       3 1e-03 0.2531550 0.02846364
4       1 1e-02 0.1953515 0.02430353
5       2 1e-02 0.2531550 0.02846364
6       3 1e-02 0.2531550 0.02846364
7       1 1e-01 0.1281410 0.02281468
8       2 1e-01 0.1953515 0.02430353
9       3 1e-01 0.2531550 0.02846364
10      1 1e+00 0.0595872 0.02267015
11      2 1e+00 0.0595872 0.02267015
12      3 1e+00 0.0595872 0.02267015
13      1 5e+00 0.0595872 0.02267015
14      2 5e+00 0.0595872 0.02267015
15      3 5e+00 0.0595872 0.02267015
16      1 1e+01 0.0595872 0.02267015
17      2 1e+01 0.0595872 0.02267015
18      3 1e+01 0.0595872 0.02267015
```

```
> confusionMatrix(predict(modelsvmpoly, testd
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     176      15
   Positif       0      57

               Accuracy : 0.9395
                 95% CI : (0.9022, 0.9658)
    No Information Rate : 0.7097
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8436

 Mcnemar's Test P-Value : 0.0003006

            Sensitivity : 0.7917
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9215
             Prevalence : 0.2903
         Detection Rate : 0.2298
   Detection Prevalence : 0.2298
      Balanced Accuracy : 0.8958

       'Positive' Class : Positif
```

Naive Bayes

```
> confusionMatrix(predict(modelNVB, testdata)
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     174       4
   Positif      10      59

               Accuracy : 0.9433
                 95% CI : (0.9067, 0.9687)
    No Information Rate : 0.7449
    P-Value [Acc > NIR] : 2.363e-16

                  Kappa : 0.8554

 Mcnemar's Test P-Value : 0.1814

            Sensitivity : 0.9365
            Specificity : 0.9457
         Pos Pred Value : 0.8551
         Neg Pred Value : 0.9775
             Prevalence : 0.2551
         Detection Rate : 0.2389
   Detection Prevalence : 0.2794
      Balanced Accuracy : 0.9411

       'Positive' Class : Positif
```

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
   10   0.1

- best performance: 0.06605756

- Detailed performance results:
   cost gamma      error dispersion
1   0.1  0.01 0.24749251 0.01772893
2   1.0  0.01 0.19053855 0.02125602
3  10.0  0.01 0.12602406 0.01971781
4   0.1  0.10 0.19053855 0.02125602
5   1.0  0.10 0.12703162 0.01968404
6  10.0  0.10 0.06605756 0.01917659
7   0.1  1.00 0.13761484 0.01869060
8   1.0  1.00 0.06605756 0.01917659
9  10.0  1.00 0.06605756 0.01917659
10  0.1  5.00 0.13257956 0.02002860
11  1.0  5.00 0.06605756 0.01917659
12 10.0  5.00 0.06605756 0.01917659
13  0.1 10.00 0.13257956 0.02002860
14  1.0 10.00 0.06605756 0.01917659
15 10.0 10.00 0.06605756 0.01917659
```

```
Confusion Matrix and Statistics

          Reference
Prediction Negatif Positif
   Negatif     350      69
   Positif       0      77

               Accuracy : 0.8609
                 95% CI : (0.8273, 0.8901)
    No Information Rate : 0.7056
    P-Value [Acc > NIR] : 3.213e-16

                  Kappa : 0.6116

 Mcnemar's Test P-Value : 2.695e-16

            Sensitivity : 0.5274
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8353
             Prevalence : 0.2944
         Detection Rate : 0.1552
   Detection Prevalence : 0.1552
      Balanced Accuracy : 0.7637

       'Positive' Class : Positif
```