BAB II

TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis Sentimen merupakan proses dalam mengolah, memahami, dan mengekstrak data dalam bentuk teks terhadap suatu topik atau kejadian untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini secara otomatis. Analisis sentimen digunakan dalam kajian tentang *opinion mining* yang biasanya digunakan oleh pengiklan, pencipta film, dan organisasi lainnya untuk mendapatkan reaksi pelanggan mereka pada topik tertentu. Penerapan analisis sentimen dapat digunakan untuk melacak persepsi produk baru, persepsi merek, dan sebuah pandangan tentang suatu opini pada skala global. (Troussas & Virvou, 2017)

Analisis sentimen juga dapat didefinisikan sebagai proses analisa teks yang dilihat dari sudut pandang polaritas sentimen yang dimiliki. Setiap teks merupakan opini yang memiliki maksud sesuai dengan sumber penulisannya yang menunjukkan kesubjektifitasan, dimana subjektifitas mempengaruhi hasil penelitian dari orang atau alat yang memproses dan membaca teks tersebut. II-3 Polaritas emosi ataupun polaritas dokumen dikelompokkan menjadi positif, negatif, dan netral. Polaritas emosi ditimbulkan dari pengalaman yang diperoleh kemudian dirasakan lalu dicurahkan dalam bentuk tulisan. Sentiment analiysist atau analisis sentimen dalam bahasa Indonesia adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikatagorikan sebagai sentimen positif maupun sentimen negatif. Hasil sistem prototipe mencapai tinggi presisi (75-95% tergantung pada data) dalam mencari sentimen pada halaman web dan artikel berita. Analisis sentimen atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi *linguistic* dan *text mining* yang memiliki tujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu. (Liu, 2011) Tugas analisis

sentimen yaitu mengelompokkan teks ke dalam kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kalimat atau dokumen yang dianalisis apakah bersifat positif atau negatif. (Dehhaf, 2010)

2.2 Pandemi Virus Corona

Pandemi virus corona adalah peristiwa menyebarnya penyakit koronavirus 2019 atau *coronavirus disease 2019* di seluruh dunia. Penyakit ini disebabkan oleh koronavirus jenis baru yang diberi nama SARS-CoV-2. Pandemi ini pertama kali dideteksi di Kota Wuhan, Provinsi Hubei, Tiongkok pada bulan Desember 2019, dan ditetapkan sebagai pandemi oleh Organisasi Kesehatan Dunia (WHO) pada 11 Maret 2020. Hingga 23 April 2020, lebih dari 2.000.000 kasus pandemi virus corona telah dilaporkan di lebih dari 210 negara dan wilayah, mengakibatkan lebih dari 195,755 orang meninggal dunia dan lebih dari 781,109 orang sembuh. (Wikipedia, 2020)

Pandemi ini telah menyebabkan gangguan sosioekonomi global, penundaan atau pembatalan acara olahraga dan budaya, dan kekhawatiran luas tentang kekurangan persediaan barang yang mendorong pembelian panik. Mis informasi dan teori konspirasi tentang virus telah menyebar secara daring, dan telah terjadi insiden xenophobia dan rasisme terhadap orang Tiongkok dan orang-orang Asia Timur atau Asia Tenggara lainnya. (Wikipedia, 2020)

2.3 Twitter

Twitter merupakan layanan jejaring sosial yang berguna untuk saling menghubungkan antara pengguna satu dengan pengguna lainnya (Basri, 2017). Berdasarkan data yang dilansir oleh statista.com, bahwa pada kuartal ketiga tahun 2020, twitter memiliki 187 juta pengguna aktif harian yang dapat dimonetisasi di seluruh dunia serta jika dilihat dari data statistik bahwa Indonesia berada di peringkat ke-6 dengan jumlah pengguna twitter sebanyak 15,1 juta per April 2021 (Tankovka, 2021). *Twitter* adalah sebuah situs web yang dimiliki dan dioperasikan oleh *Twitter* Inc., yang menawarkan jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan

Tweets. Mikroblog adalah salah satu jenis alat komunikasi online dimana pengguna dapat memperbarui status tentang mereka yang sedang memikirkan dan melakukan sesuatu, apa pendapat mereka tentang suatu objek atau fenomena tertentu. *Tweets* adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna. *Tweets* bisa dilihat secara publik, namun pengirim dapat membatasi pengiriman pesan ke daftar teman-teman mereka saja. Pengguna dapat melihat *Tweets* pengguna lain yang dikenal dengan sebutan pengikut (*follower)*. (*Twitter*, 2013). Tidak seperti *Facebook, LinkedIn*, dan *MySpace*, *Twitter* merupakan sebuah jejaring sosial yang dapat digambarkan sebagai sebuah graph berarah (Wang, 2010), yang berarti bahwa pengguna dapat mengikuti pengguna lain, namun pengguna kedua tidak diperlukan untuk mengikutinya kembali. Kebanyakan akun berstatus publik dan dapat diikuti tanpa memerlukan persetujuan pemilik..

Semua pengguna dapat mengirim dan menerima *Tweets* melalui situs *Twitter*, aplikasi eksternal yang kompatibel (telepon seluler), atau dengan pesan singkat (SMS) yang tersedia di negara-negara tertentu. Pengguna dapat menulis pesan berdasarkan topik dengan menggunakan tanda # (*hashtag*). Sedangkan untuk menyebutkan atau membalas pesan dari pengguna lain bisa menggunakan tanda @. (*Twitter*, 2013)

Pesan pada awalnya diatur hanya mempunyai batasan sampai 140 karakter disesuaikan dengan kompatibilitas dengan pesan SMS, memperkenalkan singkatan notasi dan slang yang biasa digunakan dalam pesan SMS. Batas karakter 140 juga meningkatkan penggunaan layanan memperpendek URL seperti bit.ly, goo.gl, dan tr.im, dan jasa hosting konten, seperti Twitpic, Tweephoto, memozu.com dan NotePub untuk mengakomodasi multimedia isi dan teks yang lebih panjang daripada 140 karakter (*Twitter*, 2013).

2.4 Twitter API (Application Programming Interface)

Application Programming Interface merupakan sebuah fungsi atau perintah-perintah yang digunakan untuk mengembangkan bahasa dalam *system calls* untuk memudahkan *develover* dengan bahasa atau fungsi yang lebih

terstruktur. Para *develover* menggunakan *twitter API* untuk membuat aplikasi, website, dan informasi lain yang berinteraksi dengan *twitter*.

System call interface berfungsi sebagai penghubung antara *API* dan *system call* yang dimengerti oleh sistem operasi. *System call interface* menerjemahkan perintah dalam *API* dan memanggil fungsi yang tersedia dalam *system call* yang diperlukan. Perintah dari *user* tersebut, diterjemahkan oleh program menjadi perintah *open* (). *Twitter* API terdiri dari 3 bagian yaitu sebagai berikut:

1. *Search API*

   Dirancang untuk memudahkan *user* dalam mengelola query search di konten *twitter*. Pengguna dapat menggunakannya untuk mencari keyword berdasarkan kata khusus atau mencari *tweet* lebih spesifik berdasarkan *user*name *twitter*.

2. *Representational State Transfer (REST) API*

   Twitter REST memberikan *core data* dan *core twitter objects* memperbolehkan *develover* untuk mengakses inti dari *twitter* seperti timeline, status update dan informasi *user*. *REST API* digunakan dalam membangun sebuah aplikasi *twitter* yang kompleks yang memerlukan inti dari *twitter*.

3. *Streaming API*

   Streaming API digunakan *develover* untuk kebutuhan yang lebih insentif seperti melakukan penelitian dan analisis data. *Streaming API* dapat menghasilkan aplikasi yang dapat mengetahui statistik update, follower, dan lain sebagainya.

2.5 Text Mining

Text mining (penambangan teks) adalah penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, sesuatu yang tidak diketahui sebelumnya atau menemukan kembali informasi yang tersirat secara implisit, yang berasal dari informasi yang diekstrak secara otomatis dari sumber-sumber data teks yang berbeda-beda (Feldman & Sanger, 2007). *Text mining* merupakan teknik yang digunakan untuk menangani masalah klasifikasi,

clustering, *information extraction* dan *information retrival.* (Berry & Kogan, 2010) Pada dasarnya proses kerja dari *text mining* banyak mengapdopsi dari penelitian *Data Mining* namun yang menjadi perbedaan adalah pola yang digunakan oleh *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur sedangkan dalam *Data Mining* pola yang diambil dari *database* yang terstruktur (Han & Kamber, 2006).

Text Mining dapat menentukan seberapa jauh keterhubungan dengan term atau kata, yang kata tersebut biasanya berbentuk dokumen yang akan diproses, tetapi dokumen – dokumen tersebut belum terstruktur. Jadi butuh tahapan untuk menambang term atau kata yang terdapat dalam dokumen sehingga bisa memperoleh informasi yang lebih akurat dan jelas. Pada tahapan text mining dibagi menjadi beberapa tahapan (fase pre-processing) yaitu tahapan yang dilakukan pada text mining, meliputi tahap pertama Tokennizing yaitu proses pemotongan string input.Tahap kedua Filtering proses penyaringan kata, Tahap ketiga Case Folding merupakan proses untuk merubah semua huruf besar dalam dokumen menjadi huruf kecil, Tagging proses mencari bentuk asal dari kata lampau, dan Analyzing merupakan proses untuk menentukan hubungan antara kata-kata dengan dokumen yang sudah ada. Text mining dengan pencarian otomatis sangat berkaitan karena tujuan text mining dan pencarian otomatis yaitu untuk mendapatkan atau menghasilkan informasi yang berguna dari beberapa dokumen. text mining sumber data yang digunakan berasal dari teks yang relatif tidak testruktur karena menggunakan tata Bahasa manusia atau biasa disebut (natural language). Secara umum basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks ditulis untuk dibaca langsung oleh manusia (Wijaya & Jananto, 2018). Dalam text mining terdapat beberapa tahapan untuk memproses data teks tersebut

1) Case Folding dan Tokenizing

   Case Folding biasa disebut penyeragaman kata dengan cara mengubah seluruh kata menjadi huruf kecil (lowercase). Hanya huruf a sampai z yang dapat diterima karaktek selain huruf dihilangkan. Terdapat juga kata-kata

tertentu yang harus sesuai dengan kaidah yang tidak bisa dilakukan penyeragaman kata sperti kata lembaga atau institusi yang selalui diawali huruf kapital dan juga nama gelar seperti halnya ST, M.Psi dan lain sebagainya. Tergantung dari sumber data yang digunakan untuk diproses. Tokenizing adalah suatu tahapan pemotongan string kata berdasrakan penyusunan kata tersebut.

2) Filtering

   Filtering adalah pengambilan kata-kata penting dari hasil Tokenizing atau biasa disebut pengeliminasi sebuah kata-kata sesuai dengan kaidahnya. Algoritma stop-word removal adalah salah satu yang digunakan untuk melakukan tahapan filtering

3) Word Normalization

   Word Normalization adalah suatu proses untuk memecah sutau varian-varian kata menjadi kata dasar sesuai dengan kata yang sedang diproses. Jika kata yang diproses adalah Bahasa Indonesia untuk memecah varian kata menjadi kata dasar harus sesuai dengan aturan Bahasa Indonesia salah satu algoritma yang digunakan adalah Nazief & Adriani.

4) Analyzing

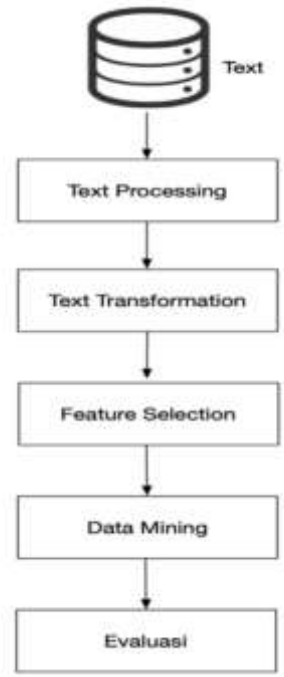
   Analyzing adalah suatu tahapan menganlisa data teks yang sedang diproses untuk menentukan kemiripan antar dokumen teks salah satu metode yang digunakan adalah cosine similarity.

2.5.1 Tujuan text mining

Tujuan dari *Text Mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen tetapi tujuan utama *text mining* adalah mendukung proses *knowledge discovery* pada koleksi dokumen yang besar. Jadi, sumber data yang digunakan pada text mining adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari text mining antara lain yaitu pengkategorisasian teks(*text categorization*) dan pengelompokkan teks (*text clustering*) (Firdaus & Firdaus, 2021)

2.5.2 Proses text mining

Beberapa tahapan proses pokok dalam text mining, yaitu pemrosesan awal text, (*text preprocessing*), transformasi teks (*text transformation*)/ (*Feature Generation*), pemilihan fitur (*feature selection*), dan penemuan pola text/data mining (*pattern discovery*). Berikut ini proses text mining yang terdapat pada gambar 2.1



Gambar 2.1 Proses Text Mining

Berikut ini keterangan gambar 2.1 tentang proses text mining:

a) *Text*

Tahap pertama adalah permasalahan yang dihadapi pada text mining sama dengan permasalahan yang terdapat pada data mining, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data noise. Perbedaan di antara keduanya adalah pada data yang digunakan.

Pada data mining, data yang digunakan adalah structured data, sedangkan pada text mining, data yang digunakan text mining pada umumnya adalah unstructured data, atau minimal semistructured. Hal ini menyebabkan adanya tantangan tambahan pada text mining yaitu struktur text yang complex dan tidak lengkap, arti yang tidak jelas dan tidak standar, dan bahasa yang berbeda ditambah translasi yang tidak akurat.

b) *Text Preprocessing*

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) terhadap teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Operasi yang dapat dilakukan pada tahap ini meliputi :[4]

1) *Text clean up.*

Menghapus iklan dari halaman web, menormalkan teks dikonversi dari format biner.

2) *Folding*

adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf „a‟ sampai dengan „z‟ yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter.

3) *Tokenization*

Sebelum pengolahan yang lebih canggih, aliran karakter berkelanjutan harus dipecah menjadi konstituen bermakna. Hal ini dapat terjadi pada tingkat yang berbeda. Dokumen dapat dipecah menjadi bab-bab, bagian, paragraf, kalimat, kata, dan bahkan suku kata. Pendekatan yang paling sering ditemukan dalam sistem text mining melibatkan teks menjadi kalimat dan kata-kata, yang disebut tokenization.

4) *Part-of-speech*

(PoS) tagging menghasilkan parse tree untuk tiap-tiap kalimat, dan pembersihan teks yang ambigu.

c) *Text Transformation (Feature Generation)*

Transformasi teks atau pembentukan atribut mengacu pada proses untuk mendapatkan representasi dokumen yang diharapkan. Pendekatan

representasi dokumen yang lazim bag of words. Transformasi teks sekaligus juga melakukan pengubahan kata-kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen.

d) *Feature Selection*

Pemilihan fitur (kata) merupakan tahap lanjut dari pengurangan dimensi pada proses transformasi teks. Operasi feature selection ini meliputi:

1) *Stop words removal*

Walaupun tahap sebelumnya sudah melakukan penghapusan kata-kata yang tidak deskriptif (*stopwords*), namun tidak semua kata-kata di dalam dokumen memiliki arti penting. Oleh karena itu, untuk mengurangi dimensi, pemilihan hanya dilakukan terhadap kata-kata yang relevan yang benar-benar merepresentasikan isi dari suatu dokumen. Langkah preprocessing yang menghilangkan atau menghapus kata-kata yang tidak penting atau tidak relevan disebut fitur seleksi. Banyak sistem, bagaimanapun, melakukan penyaringan jauh lebih agresif, menghilangkan 90 hingga 99 persen dari semua *fitur*

2) *Stemming*

Stemming merupakan suatu proses yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (root word) dengan menggunakan aturan-aturan tertentu. Algoritma Nazief & Adriani sebagai algoritma stemming untuk teks berbahasa Indonesia yang memiliki kemampuan presentase keakuratan (presisi) lebih baik dari algoritma lainnya. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan distem ke root word-nya yaitu “sama”. Proses stemming pada teks ber Bahasa Indonesia berbeda dengan stemming pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilangkan

2.6 Text Preprocessing

Proses preprocesing ini dilakukan bertujuan supaya menghasilkan term (kata) yang akan digunakan sebagai prototype dalam setiap dokumen. Tiap term tersebut dicari bentuk kata dasar-nya berdasarkan kamus kata dasar Bahasa Indonesia. Hal ini untu menghindari tersimpannya term-term (kata) yang memiliki kata dasar yang sama tapi berimbuhan berbeda. Disamping dilakukan penyaringan (filtering) terhadap kata-kata yang tidak penting untuk dijadikan sebagai pembeda. Kelompok kata ini biasanya disebut sebagai stopword. Setelah itu dilakukan tahapan text transformation yaitu penyaringan (filtration). Penyaringan itu dilakukan guna untuk menentukan term-term (kata) mana yang akan digunakan untuk merepresentasikan sebuah dokumen sehingga dapat menjelaskan isi dari dokumen dan dapat membedakan dokemen itut dengan dokumen lain dalam koleksi. Term yang sering dipakai tidak dapat digunakan untuk tujuan ini. Karena ada dua hal yaitu yang pertama, jumlah dokumen yang relevan terhadap suatu query kemungkinan besar merupakan bagian kecil dari koleksi.Kedua, term yang muncul dalam banyak dokumen tidak mencerminkan definisi dan topik atau sub-topik pada dokumen. Karena itu, term yang sering digunakan dianggap sebagai stop-word dan dihapus. Stop-word didefinisikan sebagai term (kata) yang tidak berhubungan (irrelevant) dengan subjek utama dari database meskipun term (kata) tersebut sering kali hadir di dalam dokumen (Cios, 2007). Stopword merupakan kata-kata yang biasanya sering muncul pada sebuah dokumen dengan jumlah besar yang biasanya kata tersebut merupakan kata yang tidak penting. Jadi dengan menghilangkannya dari suatu dokumen maka sistem hanya akan memperhitungkan kata-kata yang dianggap penting. Penghapusan stop-word dari dalam suatu koleksi dokumen pada satu waktu membutuhkan banyak waktu. Jadi solusinya adalah dengan menyusun suatu pustaka stop-word atau stop-list dari term yang akan dihapus

Text Preprocessing merupakan tahapan dalam text mining. Tahapan ini dilakukan untuk mengolah suatu teks di dalam dokumen agar dapat digunakan dalam proses klasifikasi. Pada dasarnya suatu dokumen yang akan digunakan memiliki teks yang tidak jelas dan tidak terstruktur, sehingga perlu adanya pemrosesan data sebelum dilakukan proses klasifikasi secara langsung. Tidak

semua kata yang akan diproses di dalam suatu dokumen dapat mencerminkan isi dari dokumen tersebut. (Feldman & Sanger, 2007)

Tahapan dari *Text Preprocessing* dilakukan dalam beberapa tahap yaitu sebagai berikut:

1. *Case Folding*

   Case Folding merupakan salah satu bentuk teknik text preprocessing. Tujuan proses ini adalah mengubah semua huruf dalam dokumen menjadi huruf kecil (Rahman, Wiranto, & Doewes, 2017). Pada proses ini juga dilakukan penghilangan tanda baca, angka dan karakter lain selain huruf alphabet. Hal ini dikarenakan karakter-karakter tersebut dianggap sebagai pemisah kata atau delimeter dan tidak memiliki pengaruh terhadap pemrosesan suatu teks (Hudin, 2018). Lalu pada tahap ini juga akan dilakukan penghapusan spasi di awal dan akhir, teknik ini biasa 9 disebut whitespace removal (Hudin, 2018).
2. *Tokenizing* merupakan proses memisahkan teks pada suatu kalimat maupun paragraf menjadi potongan atau bagian-bagian yang disebut token yang nantinya akan dianalisis. Proses ini bertujuan untuk mempermudah dalam pembobotan kata.
3. *Text Filtering*

   Filtering merupakan proses pemilihan kata-kata penting dari hasil tokenisasi, yaitu kata-kata yang bisa digunakan untuk mewakili isi dari sebuah teks atau dokumen. Proses filtering juga biasa disebut sebagai stopword removal. Pada proses ini terdapat dua teknik, yaitu stop list dan word list. Stop list merupakan proses membuang kata yang tidak deskriptif atau tidak penting. Sedangkan word list merupakan proses menyimpan kata yang dianggap penting (Hudin, 2018)
4. Word Nomalization

   Word Nomalization merupakan proses pengubahan bentuk kata menjadi kata dasar atau sebuah proses mencari akar kata dari setiap kata hasil filtering. Dengan proses ini, setiap kata yang berimbuhan akan berubah

menjadi kata dasar dan dapat lebih mengoptimalkan proses text mining (Hudin, 2018).

Salah satu library yang dapat digunakan dalam melakukan proses stemming bahasa Indonesia adalah menggunakan Library Python Sastrawi. Library ini menerapkan algoritma Algoritma Nazief dan Adriani .

2.7 Klasifikasi

Klasifikasi adalah pekerjaan yang menilai suatu objek data agar masuk kedalam kelas tertentu dari sejumlah kelas yang sudah ada. Klasifikasi dibagi menjadi dua pekerjaan utama yaitu membangun model sebagai prototype yang disimpan sebagai memori dan menggunakan model untuk melakukan pengklasifikasian prediksi pada objek data lain agar diketahui terdapat dikelas mana objek data yang disimpan (Putri, Suparti, & Rahmawati, 2014). Klasifikasi memiliki contoh aplikasi yang sering dijumpai yaitu pengklasifikasian jenis binatang yang memiliki jumlah atribut. Jika terdapat binatang baru, kelas binatang langsung dapat diketahui karena adanya atribut tersebut. Contoh lainnya adalah diagnosa penyakit kulit kanker melanoma (Prasetyo, Eko. 2012), yaitu dengan membangun model berdasarkan data latih yang ada, selanjutnya menggunakan model tersebut untuk identifikasi penyakit pasien sehingga dapat diketahui pasien mengidap penyakit kanker atau tidak. Klasifikasi merupakan metode datamining yang digunakan untuk proses pencarian sekumpulan model yang dapat membedakan kelas data atau konsep. Metode klasifikasi bertujuan untuk melakukan pemetaan data ke dalam kelas yang sudah didefinisikan sebelumnya berdasarkan nilai atribut data (Kaku, 2014).

2.8 Machine learning

Machine learning merupakan disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris. Sistem pembelajar dapat memanfaatkan contoh (data) untuk menangkap ciri yang diperlukan dari probabilitas yang mendasarinya (yang tidak diketahui). Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara variabel yang diamati. Fokus besar

penelitian pembelajaran mesin adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Kesukarannya terjadi karena himpunan semua perilaku yang mungkin, dari semua masukan yang dimungkinkan, terlalu besar untuk diliput oleh himpunan contoh pengamatan (data pelatihan). Karena itu pembelajar harus merampatkan (generalisasi) perilaku dari contoh yang ada untuk menghasilkan keluaran yang berguna dalam kasus-kasus baru. Jenis-jenis dari *machine learning* yaitu sebagai berikut:

1. Pembelajaran terarah (*supervised learning*) membuat fungsi yang memetakan masukan ke keluaran yang dikehendaki, misalnya pada pengelompokan (klasifikasi). Merupakan algoritma yang belajar berdasarkan sekumpulan contoh pasangan masukan-keluaran yang diinginkan dalam jumlah yang cukup besar. Algoritma ini mengamati contoh-contoh tersebut dan kemudian menghasilkan sebuah model yang mampu memetakan masukan yang baru menjadi keluaran yang tepat.
2. Pembelajaran tak terarah (*unsupervised learning*) memodelkan himpunan masukan, seperti penggolongan (*clustering*). Algoritma ini mempunyai tujuan untuk mempelajari dan mencari pola-pola menarik pada masukan yang diberikan. Meskipun tidak disediakan keluaran yang tepat secara eksplisit..
3. Pembelajaran semi terarah (*semi-supervised learning*), yakni tipe yang menggabungkan antara *supervised* dan *unsupervised* untuk menghasilkan suatu fungsi. Algoritma pembelajaran semi terarah menggabungkan kedua tipe algoritma di atas, di mana diberikan contoh masukan-keluaran yang tepat dalam jumlah sedikit dan sekumpulan masukan yang keluarannya belum diketahui. Algoritma ini harus membuat sebuah rangkaian kesatuan antara dua tipe algoritma di atas untuk dapat menutupi kelemahan pada masing-masing algoritma.

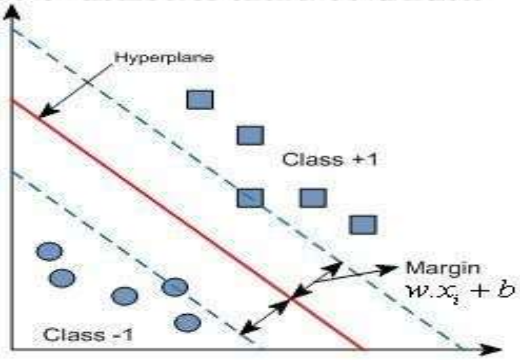
2.9 Support Vector Machine (SVM)

Dalam penelitian ini, penulis menggunakan salah satu pembelajaran dari *machine learning* yaitu pembelajaran terarah (*supervised learning)*. Metode

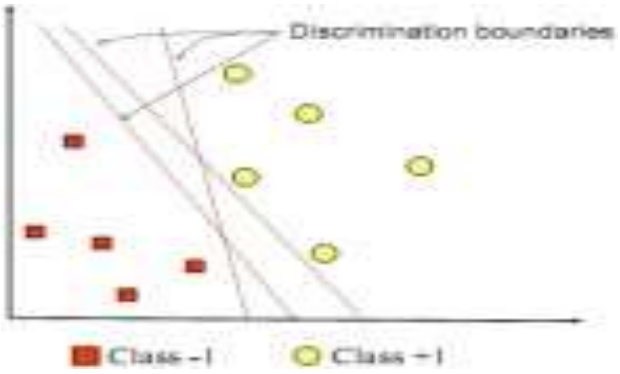
klasifikasi yang digunakan dalam *supervised learning* ini adalah *support vector machine. Support Vector Machine* merupakan salah satu dari sepuluh algoritma terbaik dalam *data mining*. (Wu dan Kumar, 2009)

Boser, Guyon, dan Vapnik pada tahun 1992 untuk pertama kali mengembangkan dan mempresentasikan teori dari algoritma SVM di *Annual Workshop on Computational Learning Theory*, meskipun dasar untuk SVM sendiri telah ada sejak 1960-an (Suyanto, 2017). Menurut Han dan Kamber (2011) metode SVM menjadi sebuah metode baru yang menjanjikan untuk mengklasifikasi data, baik data *linear* maupun *nonlinear*. SVM adalah metode yang cepat dan efektif untuk klasifikasi teks (Feldman & Sanger, 2007). Dalam istilah geometris, SVM *classifier* adalah sebuah *hyperplane* pada ruang *feature* yang memisahkan titik yang merepresentasikan *instance* kelas positif dan negatif. Pang dan Lee (2004), menyatakan bahwa metode SVM telah terbukti sangat efektif untuk kategorisasi teks tradisional mengalahkan metode *Naive Bayes.*

Teknik SVM menarik digunakan oleh para peneliti dalam bidang *data mining/text mining* maupun *machine learning* karena performa yang meyakinkan dalam memprediksi kelas suatu data baru. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space*. *Hyperplane* yang baik didapatkan dengan memaksimalkan nilai *margin. Margin* adalah jarak antara *hyperplane* dengan *support vector* (titik yang terdekat dengan *hyperplane*).



Gambar 2. 2 *Struktur Support Vector Machine*

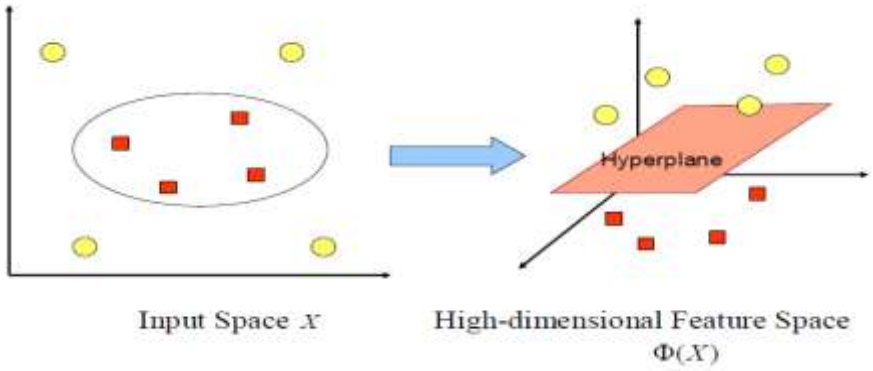


Gambar 2. 3 *Hyperplane terbentuk diantara kelas -1 dan +1*

Pada gambar 2.1 menunjukkan bahwa struktur SVM terdiri dari dua kelas data yaitu kelas +1 dan kelas -1. Kedua kelas pada struktur SVM tersebut dipisahkan oleh *hyperplane.* Data terdekat dengan garis *hyperplane* dibatasi oleh *margin* dan data yang terdapat pada *margin* disebut sebagai *support vector*. Garis merah menunjukkan *hyperplane* yang terbaik, yaitu terletak tepat pada tengah-tengah kedua kelas. Jarak *margin* dengan garis *hyperplane* ditentukan oleh nilai pembobot w dan bias b. (Feldman & Sanger, 2007)

Pada gambar 2.2 menunjukkan beberapa *pattern* yang merupakan anggota dari dua buah kelas yaitu +1 dan -1 *pattern* pada kelas +1 disimbolkan dengan warna kuning (lingkaran), sedangkan *pattern* pada kelas -1 disimbolkan dengan warna merah (kotak). Masalah pada klasifikasi di atas adalah menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. (Nugroho, 2003)

Sehingga pada prinsipnya SVM merupakan *linear classifier,* namun telah dikembangkan juga untuk menangani klasifikasi data *non-linear* dengan menggunakan konsep *kernel trick* pada ruang berdimensi lebih tinggi. (Nugroho, 2003)



Gambar 2. 4 *Non-linear SVM*

Pada gambar 2.3 menunjukkan bahwa data pada kelas kuning dan data pada kelas merah berada pada *input space* berdimensi dua yang tidak dapat dipisahkan secara *linear*. Kemudian data pada *input space (X),* memetakan fungsi Φ tiap data pada *input space* ke ruang vektor baru yang berdimensi lebih tinggi dimana kedua kelas dapat dipisahkan secara *linear* oleh *hyperplane.* (Feldman & Sanger, 2007)

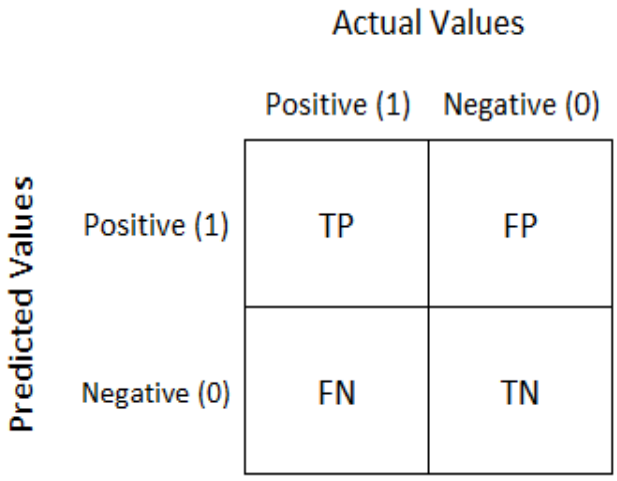
Untuk menyelesaikan problem *non-linear*, SVM dimodifikasi dengan memasukkan fungsi Kernel. Kernel *trick* berguna dalam pembelajaran SVM karena untuk menentukan *support vector* kita hanya cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi *non-linear* Φ. Berikut jenis kernel yang umum digunakan pada Tabel 2.1.

Tabel 2. 1 Jenis kernel SVM yang umum digunakan

Jenis Kernel	Definisi
Polynomial	$K\left(\vec{x_i},\vec{x_j}\right) = \left(\vec{x_i},\vec{x_j}+1\right)^p$
Gaussian RBF	$K\left(\vec{x_i},\vec{x_j}\right) = exp\left(-\frac{\left\|\vec{x_i}-\vec{x_j}\right\|^2}{2\sigma^2}\right)$
Linear	$K\left(\vec{x_i},\vec{x_j}\right) = \vec{x_i}^t\vec{x_j}$

2.10 Confusion Matrix

Pengukuran kinerja klasifikasi dilakukan dengan *confusion matrix*. *Confusion matrix* merupakan alat pengukuran yang dapat digunakan untuk menghitung kinerja atau tingkat kebenaran proses klasifikasi. Dengan confusion matrix dapat dianalisa seberapa baik classifier dapat mengenali record dari kelas-kelas yang berbeda. Tabel confusion matrix ditunjukkan pada gambar 2.5 berikut ini



Gambar 2. 5 *Confusion Matrix*

Keterangan :

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

2.11 Python

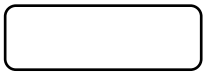
Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (*Object Oriented Programming)* serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan *syntax*. Sebagian lain mengartikan *python* sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Walaupun *python* tergolong bahasa pemrograman dengan level tinggi, nyatanya *python* dirancang sedemikian rupa agar mudah dipelajari dan dipahami.

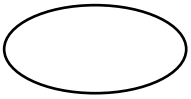
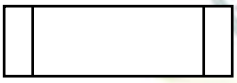
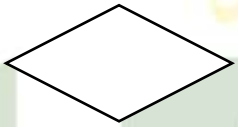
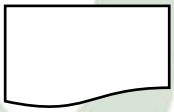
Python sendiri menampilkan fitur-fitur menarik sehingga layak untuk dipelajari. *Python* memiliki tata bahasa dan *script* yang sangat mudah untuk dipelajari. *Python* juga memiliki sistem pengelolaan data dan memori otomatis. Selain itu modul pada *python* selalu diupdate. Ditambah lagi, *Python* juga memiliki banyak fasilitas pendukung. *Python* banyak diaplikasikan pada berbagai sistem operasi seperti *Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm* dan lain-lain.

2.12 Flowchart

Flowchart merupakan gambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Adapun simbol-simbol dari diagram *flowchart* adalah sebagai berikut. (Pressman, 2006)

Tabel 2. 2 Simbol-Simbol Flowchart

No	Simbol	Keterangan
1.		Simbol titik terminal yang digunakan untuk awal dan akhir dari suatu proses

2.		Simbol input/output yang digunakan untuk mewakili data input/output
3.		Simbol proses digunakan untuk menunjukkan pengeluaran yang dilakukan oleh komputer
4.		Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama
5.		Simbol untuk database yang digunakan dalam program
6.		Simbol keputusan yang digunakan untuk suatu penyelesaian kondisi didalam program
7.		Dokumen merupakan simbol untuk data yang berbentuk kertas maupun informasi
8.		Simbol aliran data

2.13 Unified Modelling language (UML)

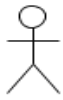
UML(*Unified Modeling Language*) pertama kali diperkenalkan pada tahun 1990-an ketika grady booch dan ivar Jacobson dan james rumbaugh mulai mengadopsi ide-ide serta kemampuan-kemampuan tambahan dari masing-masing metodenya dan berusaha membuat metodologi terpadu yang kemudian dinamakan *Unified Modeling* Language *(UML).* Menurut (Sugiarti & Sulaeman, 2015), *Unified Modeling Language (UML)* adalah gambar pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan *(modeling)*

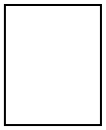
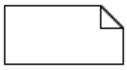
sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.
Berikut ini jenis diagram unified modelling language yang digunakan penulis:

1. *Use Case Diagram*

Use case mendokumentasikan serangkaian interaksi aktor dengan sistem. Interaksi ini dimaksudkan untuk memberikan beberapa hasil nilai yang konkret dan terukur kepada aktor. Use cases menggambarkan apa yang dilakukan suatu sistem, mereka juga menambahkan detail seperti pra dan postkondisi untuk use case, referensi antarmuka pengguna, dan aliran alternatif (Unhelkar, 2018). Notasi utama dalam *Use Case Diagram* dapat dilihat seperti pada Tabel 2.2.

Tabel 2. 3 Notasi Utama Use Case Diagram

No	Simbol	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.

5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System Boundary*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.