BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian tentang pengukuran kinerja karyawan sudah banyak dilakukan dan diterapkan pada berbagai kasus di Indonesia. Hal tersebut tentu untuk menunjang perkembangan kualitas karyawan pada suatu industri. Penelitian ini dilakukan dan dikembangkan dari beberapa referensi yang mempunyai keterkaitan dalam metode. Referensi digunakan sebagai acuan untuk pelaksanaan penelitian ini.

Penelitian yang berjudul *Algoritma Klasifikasi Naïve Bayes dan Support Vector Machine Dalam Layanan Komplain Mahasiswa* yang dilakukan oleh Hermanto, Ali Mustopa, dan Antonius Yadi Kuntoro pada tahun 2020 untuk mendapatkan algoritma yang paling akurat dalam klasifikasi komplain mahasiswa dan dapat mengetahui hasil klasifikasi dari metode algoritma SVM dan *Naïve Bayes* yang digunakan dan dibandingkan. Hasil yang diperoleh dari penelitian tersebut menghasilkan bahwa metode SVMmemperoleh akurasi yang lebih tinggi dibandingkan metode *Naïve Bayes*.

Penelitian yang berjudul *Perbandingan Kinerja Metode Naïve Bayes dan K-Nearest Neighbor Untuk Klasifikasi Artikel Berbahasa Indonesia* yang dilakukan oleh Riri Nada Devita, Heru Wahyu Herwanto, dan Aji Prasetya Wibawa pada tahun 2018 ini mengkomparasi atau membandingkan algoritma *Naïve Bayes* dan *K-Nearest Neighbor* untuk mengelompokkan artikel secara otomatis dan akurat. Hasil dari penelitian tersebut menghasilkan bahwa metode *Naïve Bayes* memiliki kinerja yang lebih baik dengan akurasi yang lebih tinggi dari pada *K-Nearest Neighbor*.

Penelitian yang berjudul *Analisis Sentimen Calon Gubernur DKI Jakarta 2017 di Twitter* yang dilakukan oleh Ghulam Asrofi Buntoro pada tahun 2017 ini bertujuan untuk membantu masyarakat menentukan sentimen yang terdapat pada

tweet masyarakat pengguna *Twitter* terhadap calon gubernur DKI Jakarta pada tahun 2017. Penelitian tersebut menggunakan algoritma *Naïve Bayes Classifier* dengan *pre-processing* data menggunakan *tokenisasi*, *cleansing*, dan *filtering*. Untuk menentukan *class* sentimen dengan metode *Lexicon Based*. Untuk proses klasifikasinya menggunakan metode *Naïve Bayes Classifier* danSVM. Data yag digunakan pada penelitian tersebut adalah *tweet* dalam Bahasa Indonesia dengan kata kunci AHY, Ahok, Anies, dengan jumlah dataset sebanyak 300 *tweet* dengan akurasi tertinggi diperoleh dengan metode *Naïve Bayes Classifier* yaitu nilai akurasi mencapai 95%, presisi 95%, *recall* 95%, *TP rate* 96,8%, dan *TN rate* 84,6%.

Penelitian yang berjudul *Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Tidak Baku* yang ditulis oleh Prananda Antinasari, Rizal Setya Perdana, dan M. Ali Fauzi pada tahun 2017 ini bertujuan agar algoritma klasifikasi *Naïve Bayes* dengan perbaikan kata tidak baku normalisasi *levenshtein distance* dapat digunakan dalam analisis klasifikasi sentimen *tweets* mengenai opini film di media sosial *Twitter* ke dalam kategori positif, dan negatif. Pada penelitian ini digunakan kamus kata tidak baku dan normalisasi *Levenshtein Distance* untuk memperbaiki kata yang tidak baku menjadi kata baku dengan pengklasifikasian *Naïve Bayes* pada tweet opini film dengan menghasilkan akurasi tertinggi dengan nilai akurasi 98.33%, presisi 96.77%, *recall* 100%, dan *f-measure* sebesar 98.36%.

Penelitian yang berjudul *Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features* yang dilakukan oleh Umi Rofiqoh, Rizal Setya Perdana, dan M. Ali Fauzi pada tahun 2017 untuk mengklasifikasi opini masyarakat mengenai penyedia layanan telekomunikasi seluler. Metode yang digunakan adalah SVM dengan menggunakan *Lexicon Based Features* sebagai pembaharuan fiturnya. Data yang digunakan pada penelitian ini merupakan data dari *Twitter* yang terdiri dari 300 data yang dibagi menjadi dua jenis yaitu 70% sebagai data latih dan 30% data uji yang kemudian menghasilkan akurasi 79% pada

metode SVM dan *Lexicon Based Features*, sedangkan tanpa *Lexicon Based Features* menghasilkan akurasi sebesar 84%.

Tabel 2.1 Kajian Penelitian Terdahulu

No.	Judul	*Comparing*	*Contrasting*	*Critisize*	*Synthesize*	*Summarize*
1.	Algoritma Klasifikasi Naïve Bayes dan *Support Vector Machine* Dalam Layanan Komplain Mahasiswa (Hermanto, Ali Mustopa, Antonius Yadi Kuntoro, 2020).	Penelitian untuk mendapatkan algoritma yang paling akurat dalam klasifikasi komplain mahasiswa dan dapat mengetahui hasil klasifikasi dari metode algoritma *Support Vector Machine* dan naïve bayes yang digunakan dan dibandingkan.	Membahas mengenai pengujian model menggunakan metode *Support Vector Machine* dan naïve bayes untuk mengetahui hasil klasifikasi komplain mahasiswa.	Pada penelitian ini tidak dijelaskan metode ekstraksi fitur yang digunakan	Pada penelitian ini, penulis melakukan pemahaman terhadap objek penelitian, kemudian mengumpulkan data dan mempersiapkan data, tahap pemodelan lalu melakukan tahapan pengujian pada tiap algoritma	Hasil penelitian yang telah dilakukan menunjukkan bahwa algoritma *Support Vector Machine* menghasilkan tingkat akurasi yang lebih tinggi daripada naïve bayes. Dengan tingkat akurasi 84,45% untuk svm, sedangkan untuk naïve bayes sebesar 69,75%.

No.	Judul	*Comparing*	*Contrasting*	*Criticize*	*Synthesize*	*Summarize*
2.	Perbandingan Kinerja Metode Naïve Bayes dan K-Nearest Neighbor Untuk Klasifikasi Artikel Berbahasa Indonesia (Riri Nada Devita, Heru Wahyu Herwanto, Aji Prasetya Wibawa, 2018).	Penelitian yang dilakukan untuk menentukan kecocokan isi artikel dengan tema jurnal menggunakan perbandingan metode naïve bayes dna k-nearest neighbor	Membandingkan metode naïve bayes dengan K-nearest Neighbor untuk klasifikasi artikel berbahasa indonesia	Kurangnya data set dan kelengkapan proses pre-processing.	Penulis melakukan pengumpulan data untuk klasifikasi yang merupakan abstrak jurnal sebanyak 40 dokumen kemudian dilakukan pre-processing yang dilanjutkan dengan proses klasifikasi dengan metode naïve bayes dan k-nearest neighbor.	Pada penelitian ini kinerja metode naïve bayes dinilai lebih unggul daripada metode k-nearest neighbor dibuktikan dengan dari 40 data uji, metode naïve bayes mampu mengklasifikasikan 28 dokumen dengan akurasi 70% sementara k-nearest neighbor sebanyak 16 dokumen dengan akurasi 40%

No.	**Judul**	***Comparing***	***Contrasting***	***Criticize***	***Synthesize***	***Summarize***
3.	Analisis Sentimen Calon Gubernur DKI Jakarta 2017 di *Twitter* (Ghulam Asrofi Buntoro, 2017).	Penelitian ini dilakukan untuk melakukan riset terhadap opini masyarakat pengguna *Twitter* terhadap calon gubernur DKI Jakarta pada tahun 2017 yang mengandung sentimen negatif positif dan netral dengan metode naïve bayes classifier dan SVM dengan menggunakan lexicon based untuk penentuan class sentiment.	Melakukan klasifikasi tweet dengan metode naïve bayes classifier dan SVM untuk mengetahui sentiment masyarakat *Twitter* terhadap calon gubernur DKI Jakarta 2017	Perlu data yang lebih banyak dan real time dan perlu dikembangkannya *stop word list* dan stemming untuk meningkatkan akurasi	Penulis melakukan pengumpulan data dari *Twitter*. Data yang diambil hanya tweet berbahasa Indonesia kemudian melakukan pre-processing dilanjutkan dengan klasifikasi untuk menguji metode lexicon based dalam menentukan opini tweet dengan menggunakan metode naïve bayes classifier dan SVM	Pada penelitian ini nilai akurasi tertinggi didapat pada metode naïve bayes classifier untuk data klasifikasi AHY dengan akurasi mencapai 95%, presisi 95%, recall 95%, TP rate 96,8%, dan TN rate 84,6%. Dapat diketahui metode naïve bayes classifier lebih tinggi akurasinya untuk klasifikasi sentimen tweet berbahasa Indonesia daripada metode SVM

No.	Judul	*Comparing*	*Contrasting*	*Criticize*	*Synthesize*	*Summarize*
4.	Analisis Sentimen Tentang Opini Film Pada Dokumen *Twitter* Berbahasa Indonesia Menggunakan *Naive Bayes* Dengan Perbaikan Kata Tidak Baku (Prananda Antina Sari, Rizal Setya Perdana, M. Ali Fauzi, 2017).	Penelitian ini dilakukan untuk menganalisis opini pengguna *Twitter* mengenai film dengan pengklasifikasian naïve bayes	Melakukan klasifikasi tweet dengan metode naïve bayes dan dengan perbaikan kata tidak baku normalisasi levenshtein distance untuk mengetahui opini film pengguna *Twitter*	Tidak ada pemaparan secara detil mengenai kriteria film yang digunakan	Peneliti melakukan proses pre-processing yang dilanjutkan dengan perbaikan kata yang tidak baku dengan normalisasi levenshtein distance serta proses pengklasifikasian dengan metode naïve bayes.	Pada penelitian ini penggunaan proses pre-processing dan perbaikan kata tidak baku dengan penambahan normalisasi Levenshtein Distance terhadap hasil klasifikasi memberikan pengaruh akurasi yang lebih baik sebesar 98, 33%, presisi 96,77%, recall 100%, dan f-measures sebesar 98,36%

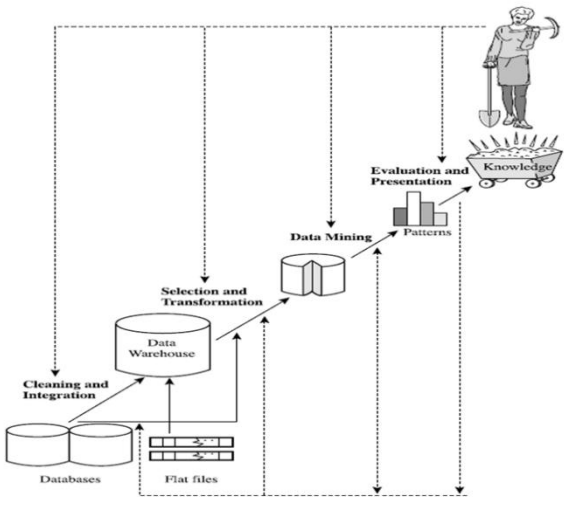
No.	Judul	*Comparing*	*Contrasting*	*Criticize*	*Synthesize*	*Summarize*
5.	Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada *Twitter* Dengan Metode *Support Vector Machine* dan Lexicon Based Features (Umi Rofiqoh, Rizal Setya Perdana, M.Ali Fauzi, 2017).	Penelitian ini dilakukan untuk mengklasifikasi opini masyarakat mengenai penyedia layanan telekomunikasi seluler pada platform *Twitter*	Melakukan klasifikasi tweet opini masyarakat mengenai penyedia layanan telekomunikasi seluler dengan metode *Support Vector Machine* dengan pembobotan lexicon based features	Tingkat akurasi sistem analisis sentimen dengan menggunakan Lexicon Based features lebih rendah daripada yang tanpa menggunakan Lexicon Based Features karena terdapat kata bersentimen negatif dalam data uji yang seharusnya positif ataupun sebaliknya	Dalam penelitian ini penulisan menggunakan data dari *Twitter* sebanyak 300 data dengan perbandingan 70% data uji dan 30% data latih yang kemudian dilakukan tahap pre-processing, serta tahap pembobotan dengan lexicon based features dan proses pengklasifikasian dengan metode *Support Vector Machine*	Hasil penelitian dengan menggunakan pembobotan lexicon based features terhadap proses klasifikasi pada penelitian ini menghasilkan akurasi sebesar 79%, presisi 65%, recal 97%, dan f-measures sebesar 78%

Kesimpulan titik temu antara penelitian peneliti dengan penelitian terdahulu yakni penggunaan metode algoritma *naïve bayes classifer* dan *Support Vector Machine* yang berfokus kepada pemrosesan sebuah data berbentuk teks. Kemudian didapatkan hasil evaluasi berbentuk *confusion matrix* terdiri dari Akurasi, Presisi, Recall untuk mengevaluasi antara kedua algoritma *naïve bayes classifier* dan *support vector machine.*

2.2. Dasar Teori

2.2.1. Data Mining

Data mining merupakan istilah yang digunakan untuk penemuan pengetahuan yang didapatkan dari sebuah basis data. Data Mining juga merupakan proses yang menggunakan statistik, matematika, kecerdasan buatan dan teknik pembelajaran mesin untuk mengekstraksi dan mengidentifikasi informasi data yang berjumlah besar untuk menggambarkan pola-pola data yang belum diidentifikasi dan diperkirakan sebelumnya [2]. Data mining mengubah kumpulan data yang besar menjadi pengetahuan [8]. Data mining biasanya menggunakan proses eksplorasi dan analisis sejumlah besar data untuk menemukan pola dan aturan yang bermakna [7]. Data Mining merupakan analisis data set pengamatan untuk menemukan hubungan yang tidak terduga dan untuk meringkas data dengan cara-cara baru yang dapat dimengerti dan berguna bagi pemilik data [8].



Gambar 2.2.1 Proses Data Mining [9].

Penjelasan proses data mining pada Gambar 2.2.1 sebagai berikut,

1. Data cleaning : untuk menghilangkan noise dan data yang tidak konsisten.
2. Data integration : mengkombinasikan atau mengintegrasi kan beberapa sumber data.
3. Data selection : mengambil data-data yang relevan dari database untuk dianalisis.
4. Data transformation : mentransformasikan data summary ataupun operasi agregasi.
5. Data mining : merupakan proses yang esensial dimana metode digunakan untuk mengekstrak pola data yang tersembunyi.
6. Patternevaluation : untuk mengidentifikasi pola sehingga mereperesentasikan pengetahuan berdasarkan nilai-nilai yang menarik.
7. Knowledge presentation : dimana teknik representasi dan visualisai data digunakan untuk mempresentasikan pengetahuan yang didapat kepada user.

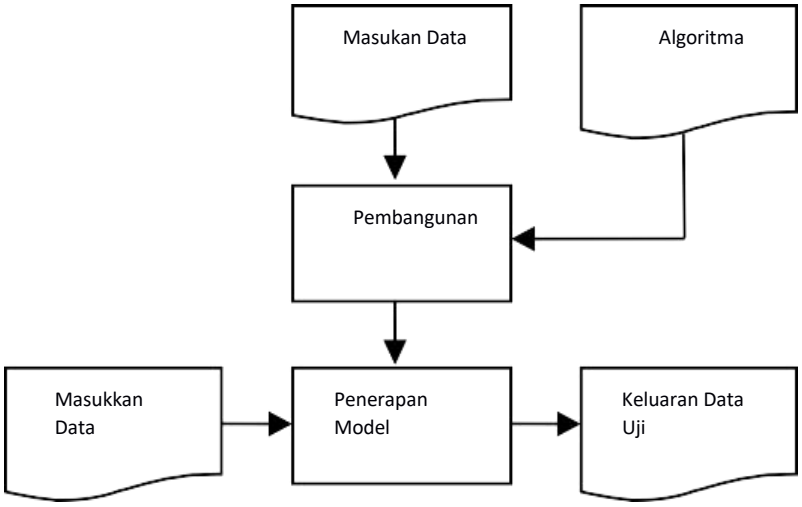
2.2.2. Classification

Klasifikasi merupakan proses yang dapat menyimpulkan karakteristik untuk mendefinisikan kelompok tertentu. Metode ini melibatkan penyemaian seperangkat data dengan seperangkat kelas yang sudah diketahui, kemudian memetakan setiap atribut kesalah satu kelas yang telah didefinisikan sebelumnya. Classification termasuk kedalam kategori supervised learning. Algoritma yang digunakan biasanya *Naïve Bayes Classifier, Support Vector Machine, Logistic Reggresion, k-Nearest Neighbors* [2].

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah *class* yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu : Pembangunan model sebagai *prototype* untuk disimpan sebagai memori, dan penggunaan model tersebut untuk melakukan pengenalan klasifikasi atau prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang mudah disimpan. Contoh aplikasi yang sering ditemui adalah pengklasifikasian jenis hewan, yang mempunyai sejumlah atribut. Dengan atribut tersebut, jika ada hewan baru, kelas hewannya bisa langsung diketahui. Contoh lain adalah bagaimana melakukan diagnosis penyakit kulit kanker melanoma yaitu dengan melakukan pembangunan model berdasarkan data latih yang ada, kemudian menggunakan model tersebut untuk mengidentifikasi penyakit pasien baru sehingga diketahui apakah pasien tersebut menderita kanker atau tidak [10].

Model Klasifikasi

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. Kerangka kerja (framework) klasifikasi disediakan sejumlah data latih (x,y) untuk digunakan sebagai data pembangunan model. Model tersebut kemudian dipakai untuk memprediksi kelas dari data uji (x,y) sehingga diketahui kelas y yang sesungguhnya [38].



Gambar 2.2.2 Contoh Proses Klasifikasi [38].

Gambar 2.2.2 menunjukkan model yang dibangun pada saat pelatihan kemudian dapat digunakan untuk memprediksi label *class* baru yang belum diketahui. Dalam pembangunan model selama proses pelatihan tersebut diperlukan suatu algoritma untuk membangunnya, yang disebut algoritma pelatihan (*Learning Algorithm*). Setiap algoritma mempunyai kelebihan dan kekurangan, tetapi semua algoritma berprinsip sama, yaitu melakukan suatu pelatihan sehingga di akhir pelatihan, model dapat memetakan (memprediksi) setiap vektor masukan ke label kelas keluaran dengan benar [38].

2.2.3. Sentiment Analysis

Analisis sentimen adalah metode untuk menganalisis sebagian data untuk mengetahui emosi manusia. Analisis sentimen dapat dikategorikan kedalam tiga *task*, yaitu *informative text detection, information extraction* dan

sentiment interestingness classification (emotional, polarity identification). *Sentiment classification* (negatif atau positif) digunakan untuk memprediksi *sentiment polarity* berdasarkan data sentimen dari pengguna.

2.2.4. Naïve Bayes

Naïve Bayes merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. *Naïve Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Teorema tersebut dikombinasikan dengan *Naive (*Kuat) dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi *Naïve Bayes* diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya.

Naïve Bayes untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi objek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari *master* tabel keputusan [11].

2.2.4.1. Bayes Theorem

Bayes merupakan teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes (aturan Bayes) dengan asumsi independensi (ketidaktergantungan) yang kuat (naif), model yang digunakan adalah model fitur independen [12].

Bayes terutama Naïve Bayes memiliki independensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama [12].

Formulasi *Naïve Bayes* untuk klasifikasi :

$$P(X) = \frac{P(H)P(H)}{P(X)} \qquad (2.1)$$

Keterangan :

X : data dengan *class* yang belum diketahui

H : hipotesis data X merupakan suatu *class* spesifik

P(H|X) : probabilitas hipotesis H berdasar kondisi X (*posteriori probability*)

P(H) : probabilitas hipotesis H (*prior probability*)

P(X|H) : probabilitas X berdasar kondisi hipotesis H

P(X) : probabilitas dari X

2.2.4.2. Naïve Bayes Classifier

Naïve Bayes Classifier termasuk model "pengklasifikasi probabilistik" yang didasarkan pada teorema *Bayesian*. Estimasi parameter untuk model *Naïve Bayes* menggunakan metode kemungkinan maksimum. Meskipun asumsi terlalu disederhanakan, sering berkinerja lebih baik dalam banyak situasi dunia nyata yang kompleks. Memiliki kelebihan membutuhkan sejumlah kecil data pelatihan untuk memperkirakan parameter [19].

Naïve Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. *Naïve Bayes Classifier* memiliki tingkat akurasi yg lebih baik dibanding model classifier lainnya [40].

Formulasi *Naïve Bayes Classifier* :

$$P(C_i) = \prod_{k=1}^{n} P(x_k|C_i) \qquad (2.2)$$

$$P(C_i) = P(C_i) * P(C_i) * \ldots * P(xn|C_i) \qquad (2.3)$$

Keterangan :

Karena asumsi atribut tidak saling terkait (conditionally independent).

Bila $P(Ci)$ dapat diketahui melaui perhitungan, maka *class* C_{new} dari data sampel X adalah *class* (label) yang memiliki $P(Ci) * P(Ci)\ max$

$$C_{new} \leftarrow\ argmax\ Ci\ P(C = Ci \prod_i P(X_i^{new}|C = C_i)) \tag{2.4}$$

Formula nilai *mean* :

$$\boldsymbol{\mu} = (\frac{\boldsymbol{x_1+x_2+x_3+...+x_n}}{\boldsymbol{n}}) \tag{2.5}$$

Keterangan :

Xi : Nilai x ke-i

$\boldsymbol{\mu}$: Rata-rata hitung

n : Jumlah sampel

Rumus *Standart Deviasi* :

$$\boldsymbol{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(\boldsymbol{x_i-\mu})^2}{\boldsymbol{n-1}}} \tag{2.6}$$

Keterangan :

σ : *Standart Deviasi*

$\boldsymbol{x_i}$: Nilai x ke –i

$\boldsymbol{\mu}$: Nilai *Mean*

n : Jumlah sampel

Karakteristik *Naïve Bayes* sebagai berikut :

1. *Naïve Bayes* bersifat *independent* (robust) terhadap data-data yang terisolasi yang biasanya merupakan data dengan karakteristik berbeda (outliner). *Naïve Bayes* bisa menangani nilai atribut yang salah dengan mengabaikan data latih selama proses pembangunan model dan prediksi.
2. Tangguh menghadapi atribut yang tidak relevan.
3. Atribut yang mempunyai kolerasi bisa mendegradasi kinerja klasifikasi *Naïve Bayes* karena asumsi independensi atribut tersebut tidak ada.

2.2.5. Support Vector Machine

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian konsep – konsep dalam berbagai bidang pattern recognition. *Support Vector Machine* (SVM) adalah seperangkat metode pembelajaran terbimbing yang menganalisis data dan mengenali pola, digunakan untuk klasifikasi dan analisis regresi. Berbeda neural network yang berusaha mencari *hyperplane* pemisah antar class, *Support Vector Machine* berusaha menemukan *hyperplane* yang terbaik pada input space. Prinsip dasar *Support Vector Machine* adalah linear classifier, dan selanjutnya dikembangkan agar dapat bekerja pada non-linear. Konsep dasar klasifikasi dengan SVM adalah mencari *hyperplane* terbaik yang digunakan sebagai pemisah antara dua kelas data. SVM hanya menggunakan beberapa titik data terpilih (support vector) yang berkontribusi untuk

membentuk model yang akan digunakan dalam proses klasifikasi. Pernyataan hyperplane dapat ditulis menggunakan persamaan 2.7 :

$$w + x + b = 0 \quad (2.7)$$

Keterangan :

w = nilai bobot

x = atribut ke i (i=1, 2, … n)

b = bias

Nilai bias mengikuti dari nilai kelas asli pada data. Apabila b dikatakan sebagai bobot tambahan W0, maka dapat dituliskan persamaan 2.8 :

$$w1x1 + w2x2 + b = 0 \quad (2.8)$$

Kernel linier digunakan ketika data yang akan diklasifikasi dapat terpisah dengan sebuah garis atau hyperplane, sedangkan kernel non-linier digunakan ketika data hanya dapat dipisahkan dengan garis lengkung atau sebuah bidang pada ruang dimensi tinggi. Fungsi kernel Linier yaitu : *K(x,y) = x.y*. Untuk memaksimalkan fungsi, digunakan persamaan 2.9 :

$$Ld = \sum Ni = 1 \propto_i \sum Ni = 0 \sum Ni = 0 \propto_i \propto_j y_i y_j K(x_i, x_j), syarat 0 \leq \propto_i \leq C \; dan \sum Ni = 0 \propto_i y_i = 0 \quad (2.9)$$

Hitung nilai w dan b, maka dapat dituliskan persamaan 2.10 :

$$W = \sum Ni = 1 \propto_i y_i x_i b = -\frac{1}{2}(wx^+ + wx^-) \quad (2.10)$$

Fungsi keputusan klasifikasi sign(f(x)), maka dapat dituliskan persamaan 2.11 :

$$f(x) = wx + b \; atau \; f(x) = \sum Mi = 1 \propto_i y_i K(x, x_i) + b \quad (2.11)$$

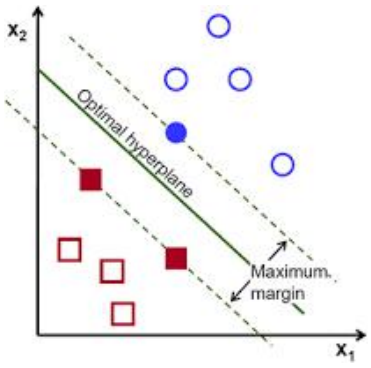
Keterangan :

N : banyaknya data

n : dimensi atau banyaknya fitur

K(x,xi) : fungsi kernel

αi : nilai bobot setiap titik data.



Gambar 2.2.3 Cara Kerja Algoritma Support Vector Machine (SVM) [21].

Berikut gambaran cara kerja algoritma *Support Vector Machine* :

Karakteristik *Support Vector Machine* (SVM) sebagai berikut :

1. Secara prinsip *SVM* adalah *linear classifier.*
2. *Pattern recognition* dilakukan dengan mentransformasikan data pada *input space* ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vector yang baru.
3. Menerapkan strategi *Structural Risk Minimization* (SRM).
4. Prinsip kerja *Support Vector Machine* hanya mampu menangani klasifikasi dua *class*.

2.2.6. Twitter

Twitter adalah sebuah situs web yang menawarkan jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirimkan dan membaca pesan yang disebut *tweets* atau kicauan, yang bebas mengekspresikan sesuatu seperti curhat atau kritik terhadap sesuatu hal [41]. *Tweets* atau kicauan adalah berupa teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil penggunanya [42]. Menurut penelitian dari We Are Social dan Hootsuite pada tahun 2019 media sosial yang populer saat ini adalah twitter dan penggunaan media sosial twitter berada pada urutan ke-5 pengguna terbanyak di seluruh dunia. Kelebihan twitter dibanding dengan media sosial lainnya adalah jangkauannya luas, tidak hanya teman, tetapi juga mampu menjangkau publik figur, potensi periklanan di masa mendatang lebih besar, komunikasi terjadi sangat cepat (*uptodate*), *multi link* (terhubung dengan banyak jaringan) dan lebih terukur dari facebook [43]. Twitter membantu penyebaran informasi secara lebih cepat yang kemudian akan menjadi sebuah topik yang dibahas oleh para penggunanya [41].

2.2.7. Crawling Data

Crawling merupakan teknik mengumpulkan data pada sebuah website dengan memasukkan Uniform Resource Locator (URL). URL ini menjadi acuan untuk mencari semua hyperlink yang ada pada website. Kemudian dilakukan indexing untuk mencari kata dalam dokumen pada setiap link yang ada. Ketika crawler melakukan pengarsipan pada situs web, crawler akan menyalin dan menyimpan informasi selagi berjalan. Data tersebut biasanya disimpan dengan metode tertentu untuk dapat dilihat, dibaca dan diatur seperti pada web asli, tetapi disimpan dengan format lain [13].

2.2.8. Preprocessing Data

Pre-processing merupakan tahap awal dari text mining untuk mengubah data sesuai dengan format yang dibutuhkan. Proses ini dilakukan untuk menggali, mengolah, dan mengatur informasi dan untuk menganalisis hubungan tekstual dari data terstruktur dan data tidak terstruktur [35]. Tahapan praproses data dilakukan dengan mengolah data mentah yang kita peroleh. Data mentah tersebut diolah dengan mengikuti proses-prosesnya agar dihasilkan data yang siap digunakan dalam penelitian [22]. Tahapan proses yang dilakukan antara lain [38] :

1. **Data Cleansing**

   Proses membersihkan dokumen dari kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah karakter HTML, kata kunci, ikon emosi, hashtag(#), username(@username), url(http://situs.com), dan email(nama@situs.com).

2. **Case folding**

   Penyeragaman bentuk huruf serta penghapusan angka dan tanda baca. Dalam hal ini yang digunakan hanya huruf latin antara a sampai dengan z.

3. ***Spelling Normalization***

   Tahap ini dilakukan normalisasi terhadap kata yang salah eja, terdapat *typo*, penyingkatan atau kata yang tidak baku yang bertujuan mengembalikkan bentuk kata sesuai dengan Kamus Besar Bahasa indonesia

4. **Tokenizing**

   Proses pengubahan dari kalimat menjadi kata-kata.

5. **Stopword removal**

   Merupakan proses pemilahan kata penghubung seperti “dan”, “pada”, dan lain sebagainya.

6. **Stemming**

   Merupakan proses mengambil kata dasar dari sebuah kata yang memiliki imbuhan, misal "mengapresiasi" menjadi "apresiasi".

2.2.9. Term Frequency Inverse Document Frequency (TF-IDF)

Metode *Term Frequency InverseDocument Frequency* (TF-IDF) merupakan metode yang digunakan untuk menentukan seberapa jauh keterhubungan kata (*term*) terhadap dokumen dengan memberikan bobot setiap kata [44]. Metode TF-IDF menggabungkan dua cara dalam perhitungan bobotnya, yaitu dengan menghitung frekuensi kemunculan kata di sebuah dokumen tertentu (TF) dan melakukan perhitungan invers terhadap frekuensi dokumen yang mengandung kata tersebut (IDF) [26].

Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. bobot kata semakin besar jika sering muncul dalam suatu dokumen, dan semakin kecil jika muncul dalam banyak dokumen [45].

Perhitungan TF dan IDF adalah sebagai berikut :

$$TF(d,t) = f(d,t) \quad \textbf{(2.12)}$$

$$IDF(t) = 1 + \boldsymbol{log}\ log(\frac{Nd}{df(t)}) \quad \textbf{(2.13)}$$

Dengan :

$TF(d,t)$: banyaknya *term* yang dicari pada sebuah dokumen

$f(d,t)$: frekuensi ditemukannya *term* t pada dokumen d

$IDF(t)$: Inverse *Document Frequency* pada dokumen ke-t

$df(t)$: jumlah dokumen dimana terdapat *term* t

Nd : jumlah keseluruhan dokumen

Sehingga rumus TF-IDF untuk menghitung bobot (W) masing-masing dokumen adalah sebagai berikut :

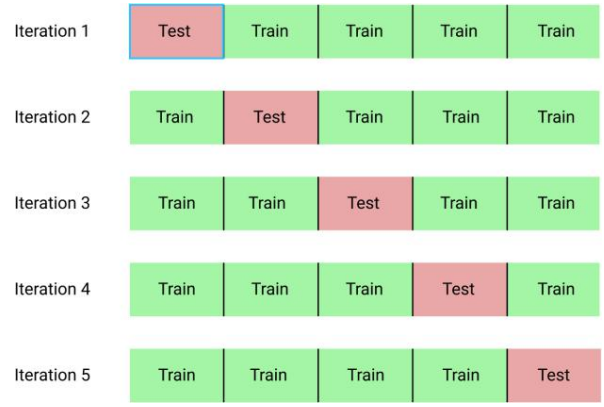
$$Wdt = TF(d,t) * IDF(t) \quad \textbf{(2.14)}$$

Dengan :

Wdt : bobot dokumen ke-d terhadap kata ke-t.

2.2.10. K-Fold Cross Validation

K-fold cross validation merupakan metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut input yang acak. *K-foldcross validation* dimulai dengan membagi data sejumlah *n-fold* yang diinginkan. Dalam proses *cross validation* data akan dibagi dalam n buah partisi dengan ukuran yang sama D1, D2, D3, …Dn selanjutnya proses *testing* dan *training* dilakukan sebanyak n kali. Dalam iterasi ke-i partisi D*i* akan menjadi data testing dan sisanya akan menjadi data training. Untuk penggunaan jumlah fold terbaik untuk uji validitas, dianjurkan menggunakan *10-fold cross validation* dalam model. Penggunaan *10 fold* ini dianjurkan karena merupakan jumlah fold terbaik untuk uji validitas [21]. Cara kerja *K-fold cross validation* dapat dilihat pada gambar dibawah ini :



Gambar 2.2.3 Cara Kerja K-Fold Validation.

Gambar 2.2.3 menjelaskan bahwa sebagai contoh setiap iterasi pertama akan dilakukan *testing* dataset secara langsung kemudian baru dilakukan *training* terhadap dataset selama 4 kali, dan iterasi kedua dilakukan *training* terhadap *dataset* terlebih dahulu kemudian baru dilakukan *testing* terhadap dataset yang akan dilakukan dalam tahap penelitian, kemudian iterasi ketiga dilakukannya dua kali *training* terhadap dataset yang akan digunakan baru dilakukan *testing* terhadap dataset yang digunakan. Iterasi keempat dilakukannnya *training* terhadap dataset selama tiga kali kemudian baru dilakukan *testing* terhadap dataset yang telah diberi latihan tadi kepada komputer. Iterasi kelima dilakukannya *training* terhadap dataset yang akan digunakan kepada komputer, kemudian baru dilakukannya *testing* dataset yang digunakan yang telah dipelajari oleh komputer sebelumnnya, hingga N atau berapa kali *Fold Validation* yang akan diterapkan oleh peneliti.

2.2.11. Confusion Matrix

Confusion matrix juga sering disebut *error matrix*. Dasarnya *confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang

dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui [34]. Beberapa *performance matrix* dari *confusion matrix* yang populer adalah *accuracy, precision, dan recall*.

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. A*ccuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. *Accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai *accuracy* yang hasil berupa persentase diperoleh dengan persamaan yaitu pada tabel rumusan **(2.15)**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} x\ 100\% \qquad (2.15)$$

Precision akan menggambarkan keakuratan antar model yang diminta dengan hasil prediksi yang diberikan oleh model dan merupakan rasio prediksi benar positif dari seluruh hasil yang di prediksi positif. Nilai *precision* yang hasil berupa persentase dengan persamaan yaitu **(2.16)**.

$$Precision = \frac{TP}{TP + FP} x\ 100\% \qquad (2.16)$$

Recall ialah menggambarkan keberhasilan model dalam menemukan kembali informasi dan merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai recall dapat diperoleh persamaan. Tabel rumus dari Recall dapat dilihat pada **(2.17)**

$$Recall = \frac{TP}{TP + FN} x\ 100\% \qquad (2.17)$$

Tabel 2.2 *Confussion Matrix*

		Actual	***Values***
		1 (Positive)	***2 (Negative)***
Predicted Values	***1 (Positive)***	***TP***	***FP***
	2 (Negative)	***FN***	***TN***

Keterangan untuk Tabel 2.2 *Confusion matrix* yang menggambarkan nilai:

• *True Positive (TP)*, merupakan data *positive* yang diprediksi memang *positive* (benar). Misalnya *sentiment analysis* pada data tersebut mengandung kata kunci *sentiment positive* serta setelah di prediksi memang benar memiliki tujuan untuk melakukan *sentiment* berbentuk *positive.*

• *True Negative (TN)*, merupakan data *negative* yang diprediksi memang *negative* (benar). Misalnya *sentiment analysis* pada data tidak mengandung kata kunci *negative* lalu setelah di prediksi benar tidak memiliki tujuan untuk melakukan *sentiment* berbentuk *positive*.

• *False Positive (FP),* merupakan data *negative* namun diprediksi sebagai data *positive* (salah atau eror). Misalnya *sentiment analysis* pada data tidak mengandung kata kunci *negative* namun setelah di prediksi memiliki tujuan untuk melakukan *sentiment* berbentuk *negative*.

• *False Negative (FN)*, merupakan data *positive* namun diprediksi sebagai data *negative* (salah atau *error*). Misalnya *sentiment analysis* pada data mengandung kata kunci *negative* namun setelah di prediksi tidak memiliki tujuan untuk melakukan *sentiment* berbentuk *negative.*

2.2.12. Python

Python merupakan salah satu bahasa pemrograman yang sifatnya open source [36]. Selain bersifat *open source*, Bahasa pemrograman *python* juga merupakan salah satu bahasa pemrograman yang dinamis serta memiliki sistem manajemen memori otomatis seperti dapat ditemui pada bahasa pemrograman lain [37]. *Python* telah banyak digunakan untuk pengembangan berbagai jenis perangkat lunak, diantaranya *systems programming, user interfaces, product customization, internet scripting, numeric programming* dan lainnya. *Python* sekarang ini merupakan bahasa pemrograman paling sering digunakan ke 4 atau 5 di seluruh dunia [36].