BAB 2
TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis Sentimen atau opinion mining merupakan proses memahami, mengekstrak, dan mengolah data tekstual secara otomatis untuk mendapatkan suatu informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dan opinion mining berfokus kepada opini yang mengekspresikan sentimen positif dan negatif[12]. Analisis Sentimen juga merupakan studi komputasi mengenai perilaku, opini-opini, dan emosi berdasarkan entitas tertentu. Entitas ini bisa berupa peristiwa atau topik tertentu berupa ulasan/review.

Secara umum, analisis sentimen sudah dilakukan penelitian pada 3 level:

1. *Document Level*

   Tugas di level ini adalah menentukan secara keseluruhan pendapat dokumen. Analisis sentimen pada tingkat dokumen mengklasifikasi bahwa setiap dokumen dapat mengekspresikan sentimen positif atau negatif.
2. *Sentence Level*

   Tugas di tingkat ini adalah menentukan masing-masing kalimat memiliki pendapat. Level ini membedakan kalimat objektif informasi faktual dan kalimat subjektif dalam memberikan pendapat positif, negatif atau netral.
3. *Entity and Aspect Level*

   Level ini melakukan analisis dan kebutuhan yang lebih baik menggunakan pemrosesan bahasa alami. Di level ini,opini menentukan sentimen berdasarkan target (aspek) pendapat dalam suatu kalimat. Seringkali pada analisis Level Dokumen (*Document Level*) atau Level Kalimat (*Sentence Level)* tidak menemukan apa yang sebenarnya disukai dan tidak disukai orang-orang.

2.2 Analisis Sentimen Berdasarkan Aspek

Analisis Sentimen berdasarkan aspek merupakan salah satu domain kasus opinion mining yang bertujuan untuk mengklasifikasikan teks berupa opini berdasarkan sentimen[13]. Kemudian para peneliti menemukan Analisis Sentimen berdasarkan aspek yang mana dapat melakukan analisis sentimen secara lebih dalam suatu ulasan atau opini. Tujuan Analisis sentimen berdasarkan aspek yaitu untuk mengidentifikasi aspek-aspek dari suatu entitas, dan sentimen yang diungkapkan oleh penulis komentar tentang aspek tersebut. Seperti saat melihat ulasan mengenai Restoran Bakso, opini yang ada bukan hanya sentimen tetapi terdapat aspek yang spesifik seperti aspek makanan, aspek layanan, dan aspek atmosfir[14].

Secara otomatis *Aspect-Based Sentimen Analysis* (ABSA) dapat mengekstraksi aspek pada ulasan atau opini, kemudian menentukan sentimen berdasarkan aspek tersebut[15]. Tahap pada penelitian ini dilakukan sebanyak dua kali klasifikasi, yaitu klasifikasi aspek dan klasifikasi sentimen.

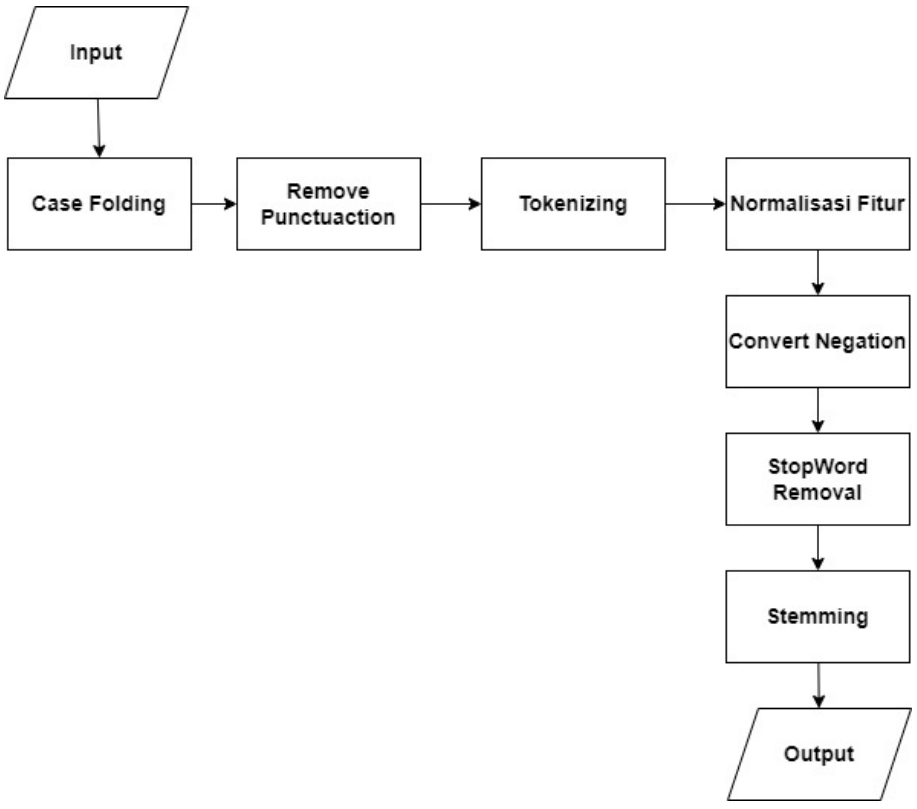
2.3 Text Mining

Text mining merupakan proses menambang data berupa teks dimana sumber data biasanya didapatkan dari dokumen dan tujuannya adalah mencari kata kata yang berguna dari beberapa dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen[16]. Sumber data yang digunakan dalam text mining adalah sekumpulan teks yang memiliki format yang awalnya tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari text mining antara lain yaitu pengkategorisasian teks dan pengelompokkan teks[14]. Text mining mengekstrak informasi berguna dari sumber data melalui identifikasi dan eksplorasi yang tidak dalam bentuk database record, melainkan dalam data teks yang tidak terstruktur. Text mining memberikan solusi dari permasalahan-permasalahan yang ada seperti pemrosesan, pengorganisasian atau pengelompokkan data dalam jumlah yang besar, dalam hal ini data yang digunakan adalah data yang diambil dari *Twitter*[17]. Dalam memberikan solusi, text mining mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti Data Mining, Information Retrieval, Statistik dan Matematik, Machine Learning, Linguistic, Natural

Language Processing dan Visualization. Kegiatan riset untuk text mining antara lain ekstraksi dan penyimpanan teks, preprocessing akan konten teks, pengumpulan data statistik serta indexing dan analisis sentimen [16].

2.4 Pre-processing

Preprocessing adalah proses pengolahan data asli yang sebelumnya tidak terstruktur menjadi terstruktur [12]. Proses ini bertujuan agar data yang akan digunakan dalam proses klasifikasi sentimen bersih dari noise. Oleh karena itu, dibutuhkan proses yang dapat mengubah data yang sebelumnya tidak terstruktur menjadi data yang terstruktur untuk diproses ke tahap selanjutnya. Adapun tahapan dari Preprocessing pada penelitian ini antara lain, *Case Folding, Remove Punctuation, Normalisasi Fitur, Tokenizing, Stopwords Removal, dan Stemming.* Berikut alur preprocessing yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 Alur Pre-Processing

Untuk penjelasan masing-masing bagian dari gambar 2.1 tentang alur pre-processing dapat dilihat dibawah ini:

2.4.1 Case Folding

Pada data Tweet yang digunakan banyaknya penggunaan huruf capital yang tidak konsisten, sehingga hal ini dibutuhkan case folding untuk mengubah semua huruf dalam dokumen menjadi huruf kecil (*lowercase*). Ini merupakan proses dalam text preprocessing yang dilakukan untuk menyeragamkan karakter pada data. Proses case folding adalah proses mengubah seluruh huruf menjadi huruf kecil. Pada proses ini karakter-karakter 'A'-'Z' yang terdapat pada data diubah kedalam karakter 'a'-'z'. Karakter-karakter selain huruf 'a' sampai 'z'[18].

2.4.2 Remove Punctuation

Remove Punctuation adalah proses menghapus semua karakter non alphabet misalnya simbol, spasi dan lain-lain [19]. Dimana sistem akan menghilangkan tanda baca atau simbol yang ada dalam dataset yang tidak akan digunakan. Tanda baca atau simbol dihapus dikarenakan tidak berpengaruh pada hasil sentiment analisis.

2.4.3 Tokenizing

Teks yang merupakan representasi sebuah baris data, dimana data tersebut harus disegmentasi menjadi kata-kata[18]. Tahap tokenizing adalah tahap pemotongan string masukan berdasarkan kata-kata yang menyusunnya atau dengan kata lain pemecahan kalimat menjadi kata. Strategi umum yang dilakukan pada tahap tokenizing adalah memotong kata pada white space atau spasi[18]. Pemecahan data menjadi kata-kata tunggal dilakukan dengan memindai data dan setiap kata teridentifikasi atau terpisahkan dengan kata yang lain oleh pemisah spasi.

2.4.4 Normalisasi Kalimat/Fitur

Normalisasi atau convert word adalah untuk mengkonversi kalimat yang tidak baku, saat ini penggunaan kalimat alay atau bahasa gaul mengakibatkan

penggunaan Bahasa Indonesia tidak baku [19]. Dengan ini semua kalimat yang diproses dianggap sudah siap untuk dikategorikan kalimat yang telah melalui proses normalisasi.

2.4.5 Convert Negation

Convert negasi merupakan proses konversi kata-kata negasi yang terdapat pada suatu tweet, karena kata negasi mempunyai pengaruh dalam mengubah nilai sentimen pada suatu tweet. Kata negasi yang terdapat pada suatu tweet akan dihilangkan, dan diberikan penanda [27]. Seperti halnya ilmu matematika, dalam bahasa terdapat kata yang dapat membalikan arti dari kata tersebut atau bersifat negasi. Kata-kata yang bersifat negasi adalah “kurang”, “tidak”, “enggak”, “ga”, “nggak”, “tak”, dan “gak”[28].

2.4.6 StopWord Removal

Tweet yang terdapat pada Twitter memiliki beberapa macam komponen tweet yang khas seperti “@” yang diidentifikasi sebagai komponen username, URL yang dikenal melalui operasi regular, hashtag yang menandakan kata sebagai topik yang sedang dibicarakan, dan “RT” yang diidentifikasi sebagai mengulang kembali tweet yang telah diposting. Komponen-komponen tersebut tidak memiliki pengaruh apapun terhadap sentimen, maka akan dibuang [7]. Stopword diproses pada sebuah kalimat jika mengandung kata-kata yang sering keluar dan dianggap tidak penting seperti waktu, penghubung dan lain sebagainya. Untuk itu perlu dilakukan penghapusan. Untuk melakukan proses penghapusan kata ini diperlukan sebuah data atau daftar kata yang diinginkan untuk dihapus[20].

2.4.7 Stemming

Stemming merupakan tahapan pada text preprocessing yang bertujuan untuk mengubah term ke bentuk akar katanya. Stem (akar kata) adalah bagian dari kata yang tersisa setelah dihilangkan imbuhannya (awalan dan akhiran)[18]. Proses stemming dilakukan dengan menghilangkan semua imbuhan baik yang terdiri dari awalan 15 (prefix), akhiran (surfix), sisipan (infix), bentuk

perulangan dan kombinasi antara awalan dan akhiran (confix). Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama[21].ini bertujuan unutk membersihkan suatu kata dari pengejaan yang kurang tepat.

Sistem akan mereduksi setiap kata dalam dataset untuk mendapatkan kata dasar dari setiap kata. Pada umumnya kata dasar pada bahasa Indonesia terdiri dari kombinasi persamaan berikut[22].

$$\textbf{Prefiks 1} + \textbf{Prefiks 2} + \textbf{Kata dasar} + \textbf{Sufiks 3} + \textbf{Sufiks 2} + \textbf{Sufiks 1}$$

2.5 TF-IDF

Term Weighting *Term frequenc-Inverse document frequency* (TF-IDF) adalah salah satu pembobotan yang sering digunakan dan merupakan gabungan dari Term Frequency dan Inverse Document Frequency. *Term frequenc-Inverse document frequency* (TF-IDF) terdiri dari frekuensi term dan inverse dokumen yang didapatkan dari membagi seluruh jumlah dokumen terhadap jumlah dokumen yang memiliki term tersebut[23]. *Term frequenc-Inverse document frequency* (TF-IDF) bertujuan untuk memberi bobot pada kata *t* dalam dokumen *d* sesuai dengan persamaan (2.2) berikut:

$$weight(t,d) = tf(\,t,d) \times idf(t,D) \qquad \textbf{(2.2)}$$

Dimana makna atau definisi dari *t, d, D, tf (t,d), idf (t, D)* berturut-turut adalah kata(*t*), dokumen(*d*), kumpulan dokumen/ *corpus(D), frequency t di d(tf (t,d),* dan *inverse document frequency* dari *t* di *D(idf(t,D)).*

Nilai tf-idf yang tertinggi adalah saat suatu kata *t* muncul berkali-kali dalam jumlah dokumen yang sedikit sedangkan nilai tf-idf menjadi lebih rendah apabila suatu kata *t* muncul lebih sedikit dalam satu dokumen, atau dalam banyak dokumen. Nilai tf-idf yang terendah adalah ketika kata muncul hampir semua dokumen[1].

Term frequency(tf) akan menunjukkan seberapa banyak kata yang muncul dalam setiap dokumen. Hal ini menunjukkan tentang seberapa penting kata tersebut

dalam suatu dokumen. Semakin tingginya bobot *ttf*menunjukkan bahwa semakin banyak kemunculan suatu kata dalam dokumen. Rumus tf adalah seperti persamaan (2.3) berikut:

$$tf = f_{t,d} \qquad \textbf{(2.3)}$$

Inverse document frequency (idf) menunujukkan tentang jarangnya suatu kata muncul. Kata yang jarang muncul berfungsi untuk membedakan satu dokumen dengan yang lainnya. Perhitungan dari idf adalah kebalikkan dari df. Rumus df adalah seperti persamaan (2.4) berikut:

$$idf = log_{10}\left(\frac{N}{df_t}\right) \qquad \textbf{(2.4)}$$

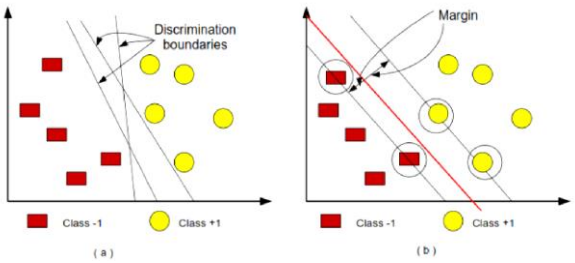
Dimana *N* menunjukkan jumlah dari dokumen, df_tmenunjukkan jumlah dari dokumen dalam corpus yang memuat kata *t.* Nilai idf yang tinggi menunjukkan jarangnya kata tersebut muncul, sedangkan nilai idf yang rendag menunjukkan kata tersebut sering muncul[1].

2.6 SVM

Algoritma SVM bertujuan untuk mencari Maximum Marginal Hyperplane (MMH) menggunakan support vector dan margin. MMH merupakan sebuah hyperplane terbaik dengan jarak margin terbesar yang digunakan untuk memisahkan data secara maksimal dan akurat terhadap setiap kelas. Margin bisa di definisikan sebagai jarak terpendek dari sebuah hyperplane terhadap satu sisi dari margin itu sama dengan jarak hyperplane dengan sisi margin lainnya, dengan catatan kedua margin tersebut dalam posisi paralel dengan hyperplane[24]. Algoritma SVM pada dasarnya digunakan untuk proses klasifikasi antara dua kelas atau *binary classification*, sesuai dengan perkembangannya SVM digunakan untuk klasifikasi multi-class yaitu dengan cara kombinasi antara beberapa binary classifier.

Sejak pertama kali dikembangkan oleh Boser, Guyon & Vapnik pada tahun 1992, untuk saat ini konsep SVM pada dasarnya adalah upaya pencarian nilai

hyperline yang terbaik pemisah antara dua buah class dalam input space, gambaran proses pemisahan tersebut seperti digambarkan pada Gambar 2.2.



Gambar 2.2 Hyperlane Class

Jika terdapat sebuah dataset dalam bentuk $(X_1, y_1), (X_2, y_2), (X_3, y_3), \ldots., (X_i, y_i)$ dimana X_i adalah training tuple dan y_i adalah label kelas dengan $i = 1 \ldots N, X_i \in Rd$ dan $yi \in \{-1,1\}$. Setiap y_i dapat memilih salah satu dari dua nilai baik +1 atau -1 SVM akan membentuk *classifier* seperti persamaan (2.5) berikut:

$$f(x_i) = \begin{cases} \geq 0, y_{i=} & +1 \\ < 0, y_{i=} & -1 \end{cases} \qquad \textbf{(2.5)}$$

Dalam SVM sebuah hyperplane akan digambarkan dalam persamaan (2.6) berikut:

$$\boldsymbol{W}.\boldsymbol{X} + b = 0 \qquad \textbf{(2.6)}$$

Keterangan

- W = Bobot Scalar

- n = banyak atribut

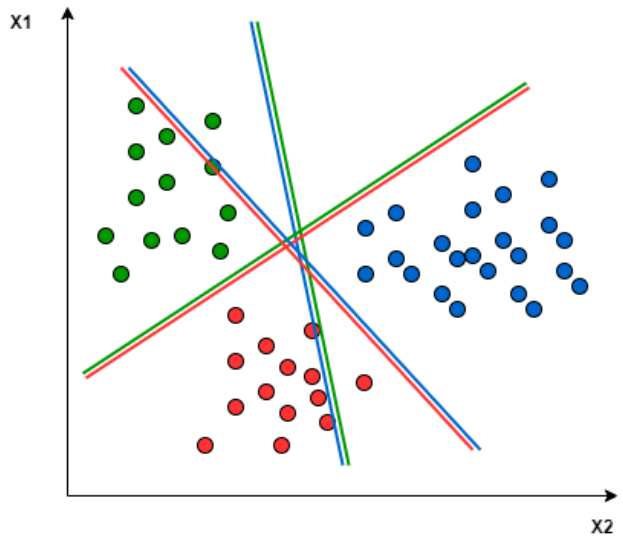
- b = Nilai scalar atau disebut bias

- X = set data latih atau training tuples

Contoh di atas menunjukkan bagaimana algoritma SVM digunakan ketika *class* nya berbentuk *binary classification* atau hanya dua kelas saja. Ketika menggunakan *multiclass* artinya lebih dari dua kelas maka SVM akan

menggunakan konsep seperti ini, yaitu untuk klasifikasi multikelas, prinsip yang sama digunakan adalah memecah masalah multiklasifikasi menjadi beberapa masalah klasifikasi biner. Cara yang paling umum dikenal ada dua cara yaitu *One-to-One Approach* dan *One-to-Rest Approach.* Ini disebut pendekatan *One-to-One*, yang memecah masalah multiclass menjadi beberapa masalah klasifikasi biner. Pengklasifikasi biner per setiap pasangan kelas. Pendekatan lain yang dapat digunakan adalah One-to-Rest. Dalam pendekatan itu, perincian diatur ke pengklasifikasi biner per setiap kelas[26].

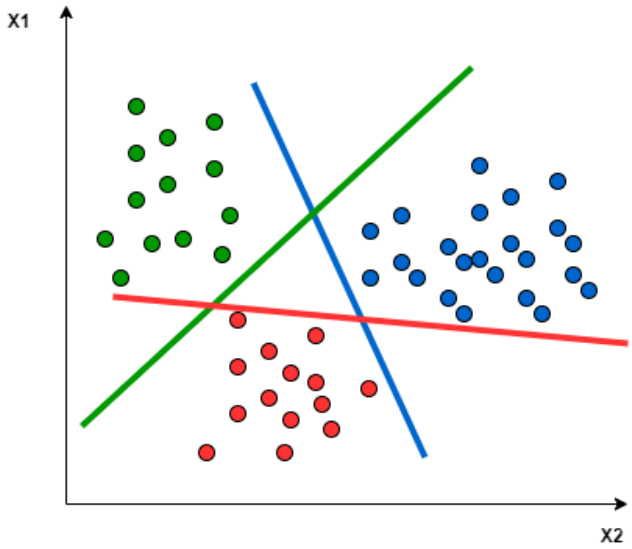
Dalam pendekatan *One-to-One*, kita membutuhkan hyperplane untuk memisahkan antara setiap dua kelas, mengabaikan poin dari kelas ketiga atau lebih. Ini berarti pemisahan hanya memperhitungkan poin dari dua kelas dalam pemisahan saat ini [26]. Jika diberikan contoh problem yang dimiliki ada 3 kelas seperti pada gambar 2.3 dibawah maka, garis merah-biru mencoba memaksimalkan pemisahan hanya antara titik biru dan merah. Ini tidak ada hubungannya dengan poin hijau



Gambar 2. 3 Contoh One-to-One

Sedangkan konsep pendekatan *One-to-Rest*, dengan asumsi kita menggunakan tiga kelas, kita membutuhkan hyperplane untuk memisahkan antara kelas dan yang lainnya sekaligus. Ini berarti pemisahan memperhitungkan semua

poin, membaginya menjadi dua kelompok; grup untuk poin kelas dan grup untuk semua poin lainnya [26]. Misalnya, garis hijau mencoba memaksimalkan pemisahan antara titik hijau dan semua titik lainnya sekaligus dengan contoh gambar dibawah ini:



Gambar 2. 4 Contoh One-to-Rest

2.7 Kernel

Kernel pada SVM digunakan untuk mentransformasi data ke ruang dengan dimensi yang lebih tinggi yang disebut sebagai ruang kernel [2x]. Berikut ini akan ditunjukkan contoh ilustrasi dari pemisahan data dengan menggunakan kernel. Misal dua buah data dinyatakan sebagai $x_i = (u_1, z_1)$ dan $x_j = (u_2, z_2)$. Diasumsikan fungsi kernel akan dibuat dengan menggunakan kedua data tersebut maka terlihat pada persamaan (2.12).

$$K(x_i, x_j) = (x_i, x_j^T)^2$$

$$K(x_i, x_j) = (u_i, u_2 + z_i, z_2)^2$$

$$K(x_i, x_j) = (u_i, u_2 + z_i, z_2)^2$$

$$K(x_i, x_j) = (u_1, \sqrt{2}\, u_1 z_1, z_1)(u_2, \sqrt{2}\, u_2 z_2, z_2)$$

$$K(x_i, x_j) = \phi(x_i)\,\phi(x_j)^T \qquad \textbf{(2.12)}$$

Nilai K yang telah disebutkan diatas telah mendefinisikan pemetaan ke ruang dengan dimensi yang lebih seperti persamaan (2.12) berikut :

$$\phi(x_i) = \{u_1, \sqrt{2}\,u_1 z_1, z_1\} \qquad \textbf{(2.13)}$$

Contoh numerik dari kernel adalah sebagai berikut. Misal $x_i = (5,3)$ dan $x_j = (2,5)$, maka seperti contoh pada persamaan pemisalan (2.13) berikut:

$$K(x_i, x_j) = (x_i \,.\, x_j^T)^2$$

$$= (5 \,.\, 2 + 3 \,.\, 5)^2$$

$$= (10 + 15)^2$$

$$= 25^2$$

$$K(x_i, x_j) = 625 \qquad \textbf{(2.14)}$$

Berikut ini merupakan beberapa fungsi kernel yang ada:

1. Kernel Linear

   Kernel linear merupakan fungsi kernel yang paling sederhana. Kernel ini cocok ketika terdapat banyak fitus dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar-benar meningkatkan kinerja, contohnya adalah pada klasifikasi teks. Dalam klasifikasi teks, baik jumlah dokumen maupun jumlah fitur (kata) sama sama besar. Berikut merupakan persamaan (2.15) dari kernel linear.

   $$K(x_i, x_j) = (x_i . x_j^T) \qquad \textbf{(2.15)}$$

2. Kernel *Radial Basis Function*

   Kernel *Radial Basis Function* (RBF) merupakan fungsi kernel yang digunakan ketika data tidak dapat dipisahkan secara linear. RBF kernel memiliki dua parameter yaitu Gamma dan Cost. Tugas dari parameter Cost atau C ini

adalah sebagai pengoptimalan SVM untuk menghindari terjadinya misklasifikasi di setiap sampel dalam training dataset. Sedangkan tugas dari parameter Gamma adalah untuk menentukan pengaruh dari satu sampel training datasetp pada garis pemisahnya. Berikut merupakan persamaan (2.16) dari kernel RBF:

$$K(x_i, x_j) = \exp[-y\|x - z\|^2] \qquad \textbf{(2.16)}$$

3. Kernel *Polynomial*

Kernel polinomial merupakan fungsi kernel yang digunakan ketika data tidak dapat dipisahkan secara linear. Kernel ini cocok digunakan untuk permasalahan dimana semua training dataset dinormalisasi. Berikut merupakan persamaan (2.17) dari kernel polynomial:

$$K(x_i, x_j) = (x_i, x_j^T)^d \qquad \textbf{(2.17)}$$

2.8 Evaluasi Model

Evaluasi model perlu dilakukan untuk mengetahui seberapa baik suatu model dapat mengklasifikasi suatu kelas dan untuk mengetahui tingkat akurasi dari hasil penggunaan metode *Support Vector Machine* (SVM) dengan cara menghitung jumlah data uji yang kelasya diprediksi dengan benar. Salah satu cara untuk melakukan hal tersebut adalah menggunakan confusion matrix. Confusion matrix adalah sebuah tabel yang menyatakan berapa banyak data uji yang benar dan salah diklasifikasikan. Parameter yang digunakan pada uji yaitu TP *(true positive),* FN *(false negative),* TN *(true negative),* FP *(false positive).*

Tabel 2.1 Confusion Matrix

Aktual	Prediksi	
	Negatif	Positif
Negatif	TN	FN
Positif	FP	TP

Dari *Confusion Matrix* tersebut dapat dihasilkan nilai akurasi. Nilai akurasi digunakan untuk mengukur seberapa akurat suatu model dalam mengklasifikasikan suatu kelas dengan benar. Formula untuk menghitung nilai akurasi adalah pada persamaan (2.17) berikut:

$$\text{Akurasi} = \frac{TN+TP}{TN+FN+TP+FN} \qquad \textbf{(2.17)}$$

Keterangan :

1. TP : True positives, merupakan jumlah data dengan kelas positif yang diklasifikasi kan positif.
2. TN : True negatives, merupakan jumlah data dengan kelas negatif yang diklasifika sikan negatif.
3. FP : False positives, merupakan jumlah data dengan kelas positif diklasifikasikan negatif.
4. FN : False negatives, merupakan jumlah data dengan kelas negatif diklasifikasikan positif.

Penggunaan metode evaluasi dengan confusion matrix ini dapat digunakan dengan merepresentasikan setiap kelas ataupun label yang dimiliki setiap dataset yang dimiliki. Pada analisis sentimen, tentu akan menggunakan positif dan negatif, sedangkan pada level aspek evaluasinya juga akan mengarah cukup dengan reprsentasi masing-masing label aspeknya, seperti pendidikan, kesehatan dst. Ketika memiliki multi kelas seperti beberapa dalam aspek maka rumus dari confusion matrix juga akan menyesuaikan, dengan contoh Kesehatan maka cukup mengakomodasi posisi TP,FP,TN dan FN nya.[25]

Tabel 2.2 Contoh *Confusion Matrix MultiClass* Kesehatan

Aktual	Prediksi			
	Kesehatan	Pendidikan	Ekonomi	Teknologi
Kesehatan	TP	FP	FP	FP
Pendidikan	FN	TN	TN	TN
Ekonomi	FN	TN	TN	TN

Teknologi	FN	TN	TN	TN

Dengan tabel di atas kita sudah dapat memperoleh TP,TN,FP dan FN dari Class Kesehatan. Setiap bagian yang sama akan diakumulasikan dan mendapatkan nilai, sehingga rumus presisi,recall dan F1-Measure masih bisa dilakukan.

Selanjutnya dari tersebut didapat kesimpulan berupa :

- *Precision* digunakan untuk mengetahui seberapa banyak persentase prediksi klasifikasi yang di prediksi *true* dan terbukti *true*. (TP) dibandingkan dengan jumlah keseluruhan prediksi bernilai *true*. Persamaan Precision :

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

- *Recall* digunakan untuk mengetahui seberapa banyak persentase prediksi klasifikasi yang di prediksi *true* dan terbukti *true* (TP) dibandingkan dengan keseluruhan data aktual bernilai *true*. Persamaan Recall :

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

- *F-Measure (F1) F1-score* adalah penggabungan nilai precision dan recall. Nilai yang dihasilkan pada perhitungan *F-Measure* ini merupakan nilai yang mewakili keseluruhan kinerja sistem. Semakin besar nilai *F1Score* yang dihasilkan maka semakin baik performansi nya. Persamaan nilai *F-Measure:*

$$F - Measure = 2x\frac{Precision \times Recall}{Precision + Recall}$$

Evaluasi bertujuan untuk mengetahui performansi sejauh mana penggunaan metode SVM dan TF-IDF dilakukan dengan cara menghitung jumlah data uji yang kelasnya diprediksi benar oleh sistem. Perhitungan evaluasi pada klasifikasi menggunakan *Accuracy. Accuracy* digunakan untuk mengevaluasi dan mengetauhi kondisi pada system dengan menghitung berapa banyak data uji yang tepat. Persamaan *Accuracy* :

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

2.9 Cross Validaton

K-fold Cross Validation adalah teknik untuk melakukan validasi pada dataset untuk menemukan akurasi yang baik. Teknik ini membagi dataset sebanyak k subset. Satu dari subset ini akan dijadikan sebagai data uji dan k-1 subset sisanya digunakan untuk proses data latih. Proses ini dilakukan sebanyak k kali sehingga setiap subset akan menjadi data uji dari model. Proses ini akan mendapatkan k buah nilai performa dari proses pembelajaran. Semua nilai performa ini akan dicari rata-ratanya dan nilai dengan rata-rata tertinggi akan dipilih sebagai model. *k-fold cross validation* memiliki kelebihan dapat mengklasifikasi dataset lebih efisien, namun metode ini memiliki kelemahan dalam proses komputasi yang digunakan akan 26 lebih besar karena akan melakukan proses sebanyak k kali [29]. Disini proses untuk membagi data ini menjadi dua bagian belum dilakukan karena semua data yang dimiliki harus masuk ke dalam proses preprocessing, baik data yang akan menjadi data latih (train) maupun data uji (test). Untuk jumlah total dataset yang dimiliki nantinya akan masuk ke tahap pre-processing, setelah itu data akan dibagi dua dengan jumlah yang berbeda dengan komposisi yang telah ditentukan pada saat proses *cross-validation* nanti.

Data latih digunakan untuk melatih algoritma dari model yang digunakan dalam hal ini adalah SVM untuk setiap aspek dan sentimen dengan asumsi proses pembelajaran akan dilakukan secara terpisah, sedangkan data uji digunakan untuk mengetahui kinerja dari yang dilatih sebelumnya. algoritma atau model ketika menemukan data baru yang tidak pernah terlihat sebelumnya.