BAB 2
LANDASAN TEORI

2.1.Analisis Sentimen

Sentimen analisis adalah proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini [1]**.** Tugas dasar dalam analisis sentimen adalah untuk mengklasifikasikan teks yang terdapat dalam dokumen, atau kalimat apakah bersifat positif, negatif atau netral. Manfaat lain dari analisis sentimen adalah dapat mengklasifikasikan ungkapan emosional seperti sedih, gembira, atau marah**.**

2.2.Tahapan Pra Proses

Tahapan pra proses merupakan tahapan untuk mengubah struktur isi dari suatu data menjadi format yang sesuai, berupa kumpulan term atau kata, agar dapat diproses oleh algoritma kalsifikasi yang digunakan [8]. Ada berbagai macam proses dalam tahapan pra proses seperti *Case Folding, Tokenizing, Stopword Removal,* Normalisasi Fitur, *Convert* Negasi, *Convert Emoticon, Remove Punctuation, Stemming,* dan Normalisasi Kalimat.

2.2.1. Case Folding

Pada proses *case folding*, sistem akan mengubah semua huruf menjadi *case* yang sama [9]. Dalam penelitian ini semua huruf akan diubah menjadi *lowercase* atau huruf kecil.

2.2.2. Tokenizing

Proses tokenisasi merupakan proses memecahkan data menggunakan spasi untuk dijadikan token-token [4]. Pemecahan data menjadi kata-kata tunggal dilakukan dengan memindai data dan setiap kata terindentifikasi atau terpisahkan dengan kata yang lain oleh pemisah spasi.

2.2.3. Stopword Removing

Sistem akan menghilangkan kata-kata yang dianggap tidak dapat memberikan pengaruh dalam menentukan suatu kategori sentimen [4]. Kata-kata stopword biasanya berupa kata ganti orang, kata ganti penghubung, pronomial penunjuk, dan lain sebagainya.

2.2.4. Normalisasi Fitur

Tweet yang terdapat pada twitter memiliki berbagai komponen atau karakteristik 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 [4].

2.2.5. Convert Negasi

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 [4].

2.2.6. Convert Emoticon

Convert emoticon adalah proses mengkonversikan emoticon ke dalam string yang sesuai dengan ekspresi emoticon itu sendiri [4]. Setelah diubah menjadi string itulah emoticon dapat diproses karena sudah sesuai dengan format yang dapat diproses.

2.2.7. Remove Punctuation

Remove Punctuation adalah proses dimana sistem akan menghilangkan tanda baca atau simbol yang ada dalam dataset [10]. Tanda baca atau simbol ini dihapus karena tidak berpengaruh pada hasil sentimen analisis. Remove punctuation sebaiknya tidak dilakukan sebelum convert emoticon karena remove punctuation dapat menghapus emoticon yang mungkin saja berpengaruh pada hasil analisis sentimen.

2.2.8. Stemming

Sistem akan mereduksi setiap kata dalam dataset untuk mendapatkan kata dasar dari setiap kata [6]. Pada umumnya kata dasar pada bahasa Indonesia terdiri dari kombinasi:

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1 (2.1)

Pada penelitian ini Algoritma yang digunakan dalam proses ini yaitu nazief dan andriani. Algoritma Stemming Bahasa Indonesia M. Adriani dan B Nazief ini mempunyai aturan imbuhan sendiri dengan model, seperti:

[[[AW+]AW+]AW+] Kata-Dasar [[+AK][+KK][+P]] (2.2)

AW : Awalan

AK : Akhiran

KK : Kata Ganti kepunyaan

P : Partikel

Tanda kurung besar menandakan bahwa imbuhan adalah opsional.

2.2.9. Normalisasi Kalimat

Normalisasi kalimat merupakan proses untuk mengubah data tweet yang tidak baku menjadi kalimat baku. Hal ini dilakukan karena ditemukan banyak tweet yang menggunakan kalimat tidak baku sehingga akan sulit dilakukan pengujian data [12].

2.2.10. Filtering

Filtering adalah tahapan dimana kata-kata yang dianggap tidak berpengaruh pada analisis sentimen akan dihilangkan [2]. Kata-kata yang dianggap tidak berpengaruh adalah kata-kata yang dianggap tidak akan mengubah hasil klasifikasi sentimen dari sebuah data.

2.2.11. Data Cleaning

Data cleaning adalah menghilangkan hal-hal yang tidak berhubungan dengan analisis sentimen [13]. Hal-hal tersebut dihilangkan karena dianggap akan mengganggu hasil analisis sentimen.

2.2.12. Lemmatization

Lemmatization adalah proses untuk mengubah kata menjadi kata dasar [13]. Kata tersebut diubah menjadi kata dasar untuk menyamakan kata yang memiliki kata dasar yang sama tetapi berbeda dalam penambahan imbuhan pada kata tersebut.

2.3.Klasifikasi

Klasifikasi adalah pengelompokan data menjadi beberapa kelompok menggunakan acuan data yang telah diketahui kelompok atau kelasnya. Data yang belum memiliki kelompok dapat ditentukan kelompoknya melalui proses pembandingan dengan data yang sudah diketahui kelompoknya [14]. Beberapa algoritma yang sering digunakan untuk klasifikasi adalah *Decision Tree, Naive Bayes, K-Nearest Neighbor (KNN)* dan *Support Vector Machine.*

2.4.Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu metode klasifikasi dengan menggunakan machine learning (supervised learning) yang memprediksi kelas berdasarkan model atau pola dari hasil proses training. Dengan melakukan training menggunakan data inputan dalam bentuk numerik dan pembobotan dengan Tf/Idf akan didapatkan sebuah pola yang nantinya akan digunakan dalam proses pelabelan. Nilai atau pola yang dihasilkan dari Metode Support Vector Machine sebenarnya adalah sebuah garis pemisah yang disebut dengan Hyperplane [15]. SVM membagi ruang vektor menjadi 2 bagian yaitu kelas positif dan kelas negatif oleh hyperplan [16]. Dalam linear Support Vector Machine permisah merupakan fungsi linear. Data latih dinyatakan oleh (xi, yi) dan xi = { x1, x2, ... xiq} merupakan atribut (fitur) set untuk data latih kelas ke-i. Untuk yi € {-1,1} menyatakan label kelas. pendefinisian persamaan suatu hyperplane pemisah yang dituliskan dengan:

$$w * xi + b = 0 \qquad (2.3)$$

Data xi yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan (2.4) berikut:

$$w * xi + b < 0 \text{ untuk } yi = -1 \qquad (2.4)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan (2.5) berikut:

$$w * xi + b > 0 \text{ untuk } yi = +1 \qquad (2.5)$$

Dimana:

xi = data input

yi = label yang diberikan

w = nilai dari bidang normal

b = posisi bidang relatif terhadap pusat koordinat

Parameter w dan b adalah parameter yang akan dicari nilainya. Bila label data yi = -1, maka pembatas menjadi persamaan (2.6) berikut:

$$w * xi + b \leq -1 \qquad (2.6)$$

Bila label data yi = +1, maka pembatas menjadi persamaan (2.7) berikut:

$$w * xi + b \geq +1 \qquad (2.7)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu 2/|w|. Hal ini dirumuskan sebagai permasalahan quadratic programming (QP) problem yaitu mencari titik minimal persamaan (2.8) dengan memperhatikan persamaan (2.9) berikut:

$$min\ \tau(w) = \frac{1}{2}\|w\|^2 \qquad (2.8)$$

$$yi(w * xi + b) - 1 \geq 0, (i = 1,..,n) \qquad (2.9)$$

Permasalahan ini ini dapat dipecahkan dengan berbagi teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.8) ke dalam fungsi Lagrangian pada persamaan (2.10), dan menyederhanakannya menjadi persamaan (2.11) berikut:

$$L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i(y_i((w^T x_i + \mathrm{b}) - 1)) \qquad (2.10)$$

$$L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i y_i(w^T x_i + b) + \sum_{i=1}^{n} a_i \qquad (2.11)$$

Dimana ai adalah lagrange multiplier yang bernilai nol atau positif ($ai \geq 0$). Nilai optimal dari persamaan (2.11) dapat dihitung dengan meminimalkan L terhadap w, b dan a. Dapat dilihat pada persamaan (2.12) sampai (2.14) berikut:

$$\frac{\partial \mathrm{L}}{\partial \mathrm{w}} = w - \sum_{i=1}^{n} a_i y_i x_i = 0 \qquad (2.12)$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = \sum_{i=1}^{n} a_i y_i = 0 \quad (2.13)$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i = 0 \quad (2.14)$$

Maka masalah Lagrange untuk klasifikasi dapat dinyatakan pada persamaan (2.15) berikut:

$$Min\ L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i \quad (2.15)$$

Dengan memperhatikan persamaan (2.16) dan (2.17) berikut:

$$w - \sum_{i=1}^{n} a_i y_i x_i = 0 \quad (2.16)$$

$$\sum_{i=1}^{n} a_i y_i x_i = 0 \quad (2.17)$$

Model persamaan (2.15) diatas merupakan model primal Lagrange. Sedangkan dengan memaksimalkan L terhadap ai, persamannya menjadi persamaan (2.18) berikut:

$$Max \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1,j=1}^{n} a_i a_j y_i y_j^T x_i x_j^T \quad (2.18)$$

Dengan memperhatikan persamaan (2.19) berikut:

$$\sum_{i=1}^{n} a_i y_i = 0, a_i \geq 0 (i, j = 1, \dots, n) \quad (2.19)$$

Untuk mencari nilai xi dan yi dapat dilakukan ketika sudah didapatkan nilai tiap kata (term) dari pembobotan tf-idf dan inisialisasi kelas . Hasil dari pembobotan tf/idf diubah ke dalam bentuk format data svm, sedangkan data kelas menjadi label data svm. Untuk mendapatkan nilai ai, langkah pertama adalah mengubah setiap data menjadi nilai vektor (support vector) =$\begin{pmatrix} x \\ y \end{pmatrix}$. Kemudian nilai vektor dari setiap data dimasukkan ke persamaan (2.20) kernel trick phi berikut:

$$\phi\begin{bmatrix} x \\ y \end{bmatrix} = \begin{cases} \sqrt{x\frac{2}{n} + y\frac{2}{n}} > 2\ maka & \begin{bmatrix} \sqrt{x\frac{2}{n} + y\frac{2}{n}} - x + |x-y| \\ \sqrt{x\frac{2}{n} + y\frac{2}{n}} - y + |x-y| \end{bmatrix} \\ \sqrt{x\frac{2}{n} + y\frac{2}{n}} \leq 2\ maka & \begin{bmatrix} x \\ y \end{bmatrix} \end{cases} \quad (2.20)$$

Nilai x didapatkan dari persamaan (2.21) kernel linear untuk x berikut:

$$\sum_{i=1;j=1}^{n} x_i x_j^T, (\mathrm{i,j} = 1, \dots, \mathrm{n}) \quad (2.21)$$

Nilai y didapatkan dari persamaan (2.22) kernel linear untuk y berikut:

$\sum_{i=1;j=1}^{n} y_i y_j^T, (\mathrm{i,j} = 1, \ldots, \mathrm{n})$ (2.22)

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.20) diberi nilai bias = 1 . Kemudian cari parameter *ai*, dengan terlebih dahulu mencari nilai fungsi setiap data menggunakan persamaan (2.23), lalu mencari nilai *ai* pada persamaan linear menggunakan persamaan (2.24) dengan memperhatikan *i,j* = 1,..,*n* berikut:

$\sum_{i=1;j=1}^{n} a_i S_i^T S_j$ (2.23)

$\sum_{i=1;j=1}^{n} a_i S_i^T S_j = y_j$ (2.24)

Setelah parameter *ai* didapatkan, kemudian masukkan ke persamaan (2.25) berikut:

$\hat{\mathrm{W}} = \sum_{i=1}^{n} a_i S_i$ (2.25)

Hasil yang didapatkan menggunakan persamaan (2.25), selanjutnya digunakan persamaan (2.26) untuk mendapatkan nilai w dan b:

$$y = wx + b \qquad (2.26)$$

Sedemikian sehingga didapatkanlah nilai w dan nilai b atau nilai hyperplane untuk mengklasifikasikan kedua kelas. Sebuah fungsi bisa menjadi fungsi kernel jika memenuhi Teorema Mercer, yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat semi positive semi definite. berikut ini adalah beberapa fungsi kernel yang umum digunakan yaitu:

a. Kernel linier $K(xi,x) = xi^T x$

b. Polynomial $K(xi,x) = (\gamma . xi^T x + r)^p$,$\gamma > 0$

c. Radial basis function $K(xi,x) = \exp(-\gamma \left| xi-x \right|^2), \gamma > 0$

d. Sigmoid kernel $K(xi,x) = \tanh(\gamma xi^T + r)$.

2.5.Akurasi

Untuk mengukur akurasi dapat menggunakan rumus [13]:

$$Akurasi = \frac{Jumlah\ yang\ diklasifikasi\ secara\ benar}{Total\ sampel\ testing\ yang\ diuji} \quad (2.26)$$