BAB II

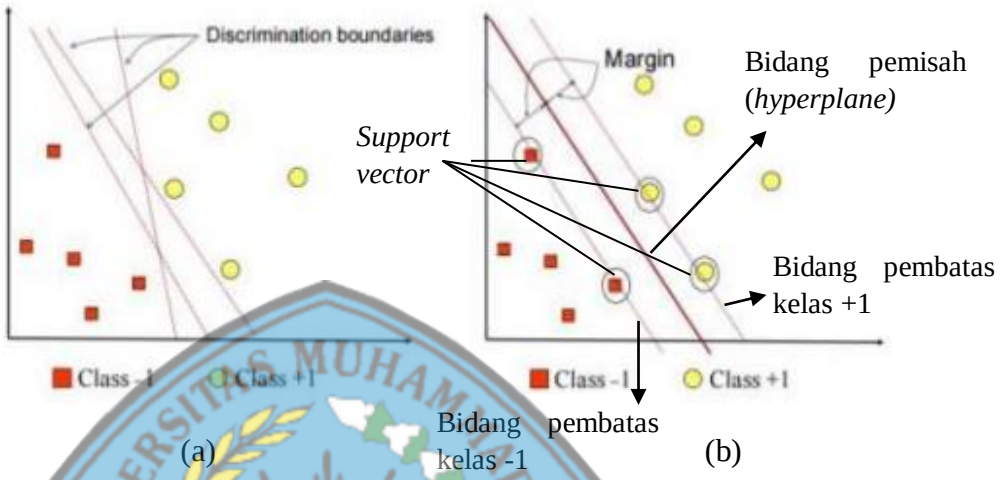
TINJAUAN PUSTAKA

2.1 *Support Vector Machine*

Support Vector Machine (SVM) diperkenalkan oleh Vapnik pada tahun 1992 sebagai suatu teknik klasifikasi yang efisien untuk masalah nonlinear. SVM juga dikenal sebagai teknik pembelajaran mesin (*machine learning*) paling mutakhir setelah pembelajaran mesin sebelumnya yang dikenal sebagai *Neural Network* (NN). Baik SVM maupun NN tersebut telah berhasil digunakan dalam pengenalan pola. Pembelajaran dilakukan menggunakan pasangan data input dan data output berupa sasaran yang diinginkan. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari h*yperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space*. SVM berusaha menemukan fungsi pemisah (*hyperplane*) dengan memaksimalkan jarak antar kelas. Dengan cara ini, SVM dapat menjamin kemampuan generalisasi yang tinggi untuk data-data yang akan datang (Suyanto, 2017). Kelebihan *Support Vector Machine* yaitu metode yang paling akurat untuk teks klasifikasi (Moraes, dkk, 2013).

2.1.1 SVM pada Data Terpisah secara Linear

Ilustrasi SVM pada data terpisah secara linear dapat dilihat pada Gambar 2.1 di bawah ini :



Gambar 2.1 Ilustrasi SVM menemukan *hyperline* terbaik untuk memisahkan kelas

(Sumber: Nugroho, Witarto & Handoko, 2003)

Gambar 2.1 di atas memperlihatkan konsep dasar *Support Vector Machine* (SVM) dimana penyebaran data memiliki dua kelas yang ditunjukkan oleh kotak warna merah (kelas negatif yang dinotasikan dengan -1) dan lingkaran warna kuning (kelas positif yang dinotasikan dengan +1). Masalah utama dalam klasifikasi dari gambar di atas adalah mencari *hyperplane* pemisah antara kedua kelas. Dari gambar 2.1a terlihat terdapat beberapa alternatif garis pemisah (*discrimination boundaries)* antara kedua kelas. *Hyperplane* pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing kelas. Subset data *training set* yang paling

dekat ini disebut dengan *support vector*. Bidang pemisah terbaik adalah bidang yang memisahkan data dan memiliki margin yang besar (Sembiring, 2007).

Garis solid warna merah pada Gambar 2.1 sebelah kanan menunjukkan *hyperplane* pemisah terbaik, yaitu yang terletak tepat di tengah-tengah kedua kelas, sedangkan titik kotak dan lingkaran yang berada dalam lingkaran hitam pada bidang pembatas adalah *support vector*. Upaya mencari lokasi *hyperplane* optimal ini merupakan inti dari proses pembelajaran pada SVM. Bidang pembatas pertama membatasi kelas pertama sedangkan pada pembatas bidang kedua membatasi kelas kedua sehingga diperoleh persamaan :

$$x_i . w + b = 0, untuk\ y_i = 0\ (hyperplane)$$

$$x_i . w + b \geq +1, untuk\ y_i = +1\ (kelas\ positif)$$

$$x_i . w + b \leq -1, untuk\ y_i = -1\ (kelas\ negatif) \qquad 2.1$$

Keterangan :

w = bobot

x = data (*input)*

b = bias

Nilai margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak dengan titik terdekatnya dihitung menggunakan rumus $\frac{1-b-(1-b)}{w} = \frac{1}{|w|}$. Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) *problem*, yaitu mencari titik minimal persamaan 2.2

dengan memperlihatkan *constraint* persamaan 2.3 dimana memaksimalkan $\frac{1}{|w|}$ sama dengan meminimumkan $|w|^2$,

$$min_w = \frac{1}{2}|w|^2 \qquad 2.2$$

$$y_i(x_i.w + b) - 1 \geq 0 \qquad 2.3$$

dengan $|w|$ yang merupakan vektor normal.

QP *problem* ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya *lagrange multiplier* yang dinyatakan dalam persamaan 2.4 berikut:

$$L(w, b, \alpha) = \frac{1}{2}|w|^2 - \sum_i^l \alpha_i(y_i(x_i.w + b) - 1)) \qquad 2.4$$

dengan $i = 1,2, \ldots\ldots, l$.

Dimana α merupakan *lagrange multiplier* yang bernilai 0 atau positif $\alpha_i \geq 0$. Nilai optimal dari persamaan 2.4 di atas dapat dihitung dengan meminimalkan L terhadap w dan b, serta memaksimalkan L terhadap α_i. Selain itu dengan memperlihatkan sifat bahwa pada titik optimal *gradient* L = 0 persamaan 2.4 dapat dimodifikasi sebagai maksimalisasi *problem* yang hanya mengandung α_i , sebagaimana terlihat pada persamaan 2.5 dan 2.6 berikut:

$$L = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j x_i x_j \qquad 2.5$$

Dimana $\alpha_i \geq 0\ (i = 1,2, \ldots\ldots, l) \sum_{i=1}^{l} \alpha_i y_i = 0$ 2.6

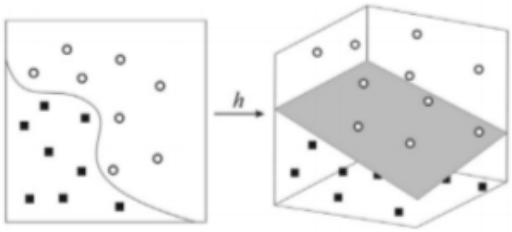
Dengan demikian akan diperoleh α_i dengan mayoritas bernilai positif yang disebut sebagai *support vector* dan juga memperoleh persamaan 2.7 dan 2.8 sebagai solusi dalam bidang pemisah (*hyperplane*).

$$w = \sum \alpha_i y_i x_i \qquad 2.7$$

$$b = y_k - w^T x_k \qquad 2.8$$

2.1.2 SVM pada Data Tidak Terpisah secara Linear

Dalam beberapa kasus, dapat ditemukan bahwa himpunan data tidak dapat dipisahkan secara linear. SVM mampu menyelesaikan permasalahan tidak linear dengan menggunakan teknik kernel (Cortes dan Vapnik, 1995). Pada dasarnya, penggunaan kernel ini memetakan vektor masukan pada ruang berdimensi rendah ke ruang berdimensi lebih tinggi. Gambar 3.2 menunjukkan bahwa data masukan yang tidak dapat dipisahkan secara linear kemudian ditransformasikan ke dalam ruang berdimensi lebih tinggi *(feature space)*. Jika pada data linear, *hyperplane* berbentuk sebuah garis yang memisahkan antar kelas, maka pada data non linear, *hyperplane* akan berbentuk sebuah bidang yang memisahkan antar kelas.



Gambar 2.2 Transformasi dari *input space* ke *feature space*

(Sumber: Moraes dkk, 2013)

Kasus data yang tidak terpisah secara linear diasumsikan bahwa kelas pada *input space* tidak dapat terpisah secara sempurna. Selain itu, *feature*

space pada kenyataannya memiliki dimensi yang lebih tinggi dari pada vektor input (*input space*). Hal ini menyebabkan komputasi pada *feature space* sangat besar sehingga terdapat kemungkinan *feature space* memiliki jumlah *feature* tak terhingga. Maka untuk mengatasi masalah ini SVM digunakan "*kernel trick"*.

2.1.3 *Kernel Trick* dan *Non Linear Classification* pada SVM

Untuk menyelesaikan masalah *non linear* dalam pemisahan data, SVM dimodifikasi dengan memasukkan fungsi *Kernel*. Keistimewaan dari *Support Vector Machine* berasal dari kemampuan untuk menerapkan pemisahan linear pada input data non linear berdimensi tinggi, dan ini diperoleh menggunakan fungsi *kernel* yang diperlukan. Efektivitas *Support Vector Machines* sangat dipengaruhi oleh jenis fungsi kernel yang dipilih dan diterapkan berdasarkan karakteristik data (Haddi, dkk, 2013).

$$\phi : \Re^2 \rightarrow \Re^2 \quad d < q \qquad 2.9$$

Data SVM *non linear*, data $\vec{x}$ dipetakan oleh fungsi $\phi(\vec{x})$ ke ruang vektor yang berdimensi lebih tinggi. Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*, sebaliknya jika dua data yang berjarak jauh pada *input space* maka akan berjarak jauh juga pada *feature space*. Kemudian, proses pembelajaran pada SVM hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi yaitu $\phi(x_i).\phi(x_j)$. karena

transformasi ϕ tidak diketahui, maka perhitungan *dot product* dapat digantikan dengan fungsi *kernel* $K(x_i, x_j)$ yang secara implisit mendefinisikan fungsi transformasi ϕ tersebut. Inilah yang disebut *Kernel Trick*, dimana formulasi fungsi tersebut sebagai berikut:

$$K(x_i, x_j) = \phi_i(x_i).\phi_j(x_j) \quad 2.10$$

Beberapa *kernel* yang umum digunakan pada SVM di antaranya:

2.1.3.1 *Polynomial*

Kernel trick polynomial diformulasikan untuk digunakan dalam menyelesaikan masalah klasifikasi, dimana dataset pelatihan yang digunakan sudah normal. Berikut persamaan:

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i}, \overrightarrow{x_j} + 1)^d \quad 2.11$$

2.1.3.2 *Radial Basis Function (RBF)* atau *Gaussian*

Kernel Gaussian ini merupakan *kernel* yang paling banyak digunakan dalam penyelesaian masalah klasifikasi untuk dataset yang tidak terpisah secara *linear,* dikarenakan pada *kernel* ini memiliki akurasi prediksi yang sangat baik. Persamaan yang dimiliki sebagai berikut:

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = exp(-\left\|\overrightarrow{x_i} - \overrightarrow{x_j}\right\|^2)\gamma \quad 2.12$$

2.1.3.3 *Sigmoid Kernel*

Sigmoid merupakan *kernel trick* SVM yang merupakan pengembangan dari jaringan saraf tiruan, dimana *kernel* ini dinyatakan dengan persamaan berikut:

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \tanh(\overrightarrow{ax_i} + \overrightarrow{x_j} + \beta) \quad 2.13$$

Kernel trick memberikan beberapa kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support vector*, pengguna hanya cukup mengetahui fungsi *kernel trick* yang dipakai, tanpa perlu mengetahui wujud dari fungsi *non linear*. Dari keseluruhan *kernel trick* tersebut, *kernel trick radial basis function* merupakan *kernel trick* yang memberikan hasil terbaik pada proses klasifikasi khususnya untuk data yang tidak dapat dipisahkan secara *linear*. Untuk klasifikasi sebuah objek data x dapat diformulasikan sebagai berikut :

$$f(x) = \sum_{i=1,\overrightarrow{x_i},\epsilon SV}^{n} \alpha_i y_i K(\overrightarrow{x_i}, \overrightarrow{x_j}) + b \quad 2.14$$

2.2 *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 memberikan sebuah set metodologi dan *tool* untuk menemukan (*discovering*), memvisualisasikan (*presenting*), mengevaluasi pengetahuan dari kumpulan besar dari teks dokumen. Pengolahan *text mining* ini untuk memperoleh informasi berkualitas tinggi dari teks, biasanya diperoleh karena memperhatikan pola dan tren melalui cara seperti mempelajari pola statistik. *Text mining* biasanya termasuk kategorisasi teks,

teks *clustering*, ekstraksi konsep, analisis sentiment, merangkum dokumen dan pemodelan hubungan entitas (misalnya mempelajari hubungan antar entitas) (Purbo, 2017).

Text mining adalah teknologi yang mampu menganalisa data teks semi terstruktur maupun tidak terstruktur, sedangkan *data mining* mengolah data yang terstruktur (Han & Kamber, 2012). Itulah perbedaan mendasar antara *text mining* dan *data mining* yang terletak pada sumber data. Data yang diolah dalam *data mining* mudah diproses oleh mesin atau komputer. Sedangkan dalam *text mining* proses analisis lebih sulit dilakukan karena bukan digunakan untuk mesin atau komputer tetapi sebagai konsumsi manusia secara langsung. Selain itu, struktur teks yang kompleks dan tidak lengkap, bahasa yang berbeda dan arti yang tidak standar. Maka untuk mengubah teks menjadi data untuk dianalisis, pada umumnya menggunakan *Natural Language Processing* (NLP) (Purbo, 2017). Tahapan-tahapan dalam *text mining* secara umum adalah *text preprocessing* dan *feature selection* (Feldman & Sanger, 2007). Penjelasan tahap-tahap tersebut sebagai berikut:

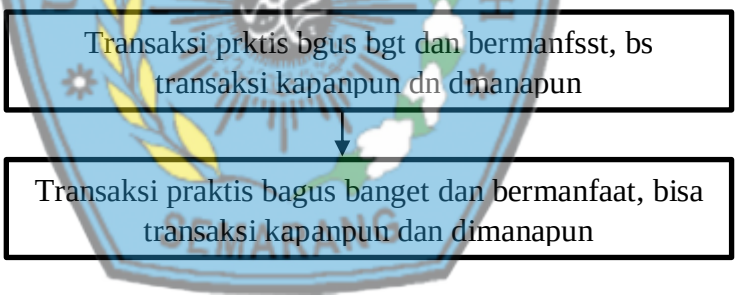
2.2.1 *Text Preprocessing*

Untuk mendapatkan data yang siap diproses oleh *data mining*, maka harus melakukan *text preprocessing* (mempersiapkan teks dokumen atau dataset mentah) terlebih dahulu dengan cara penyeleksian kata. Setiap kata dipecah menjadi bagian yang lebih kecil sehingga mempunyai arti yang lebih sempit dan jelas. *Text preprocessing* berfungsi untuk mengubah data teks yang tidak terstruktur menjadi data yang terstruktur. Secara umum

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

2.2.1.1 *Spelling Normalization*

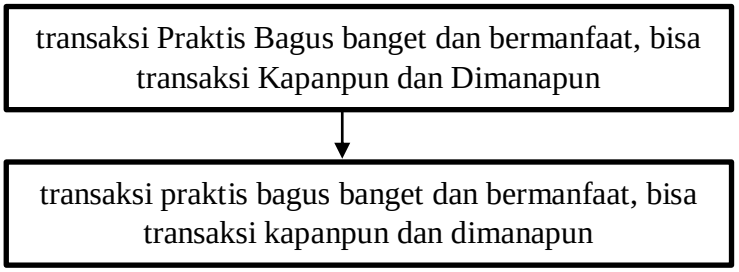
Merupakan proses perbaikan atau subtitusi kata-kata yang salah eja atau disingkat dalam bentuk tertentu. Subtitusi kata dilakukan untuk menghindari jumlah perhitungan dimensi kata yang melebar. Perhitungan dimensi kata akan melebar jika kata yang salah eja atau disingkat tidak diubah karena kata tersebut sebenarnya mempunyai maksud dan arti yang sama tetapi akan dianggap sebagai entitas yang berbeda pada saat proses penyusunan matriks. Contoh dapat dilihat pada Gambar berikut:



Gambar 2.3 *Spelling Normalization*
(Sumber: Firmansyah, dkk, 2017)

2.2.1.2 *Case Folding*

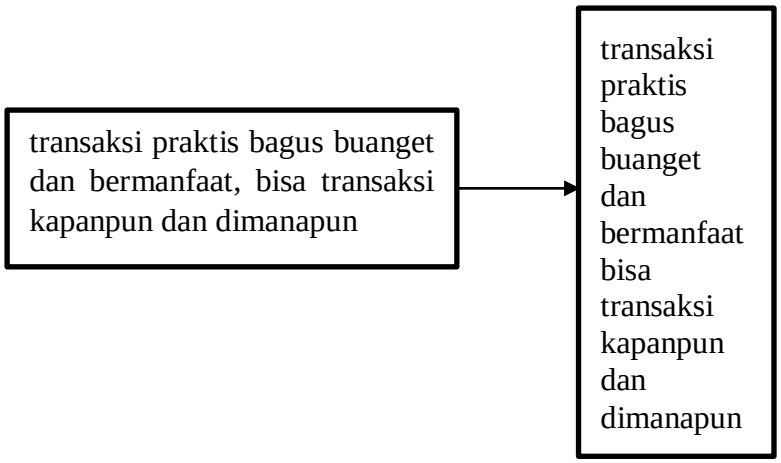
Merupakan tahapan untuk merubah semua huruf dalam dokumen menjadi huruf kecil (*lowercase)*. Hal ini dilakukan untuk mempermudah pencarian. Karena tidak semua dokumen teks atau *database* konsisten dalam penggunaan huruf kapital. Berikut ilustrasi *case folding* dapat dilihat pada Gambar 2.4:



Gambar 2.4 *Case Folding*
(Sumber: Firmansyah, dkk, 2017)

2.2.1.3 *Tokenizing*

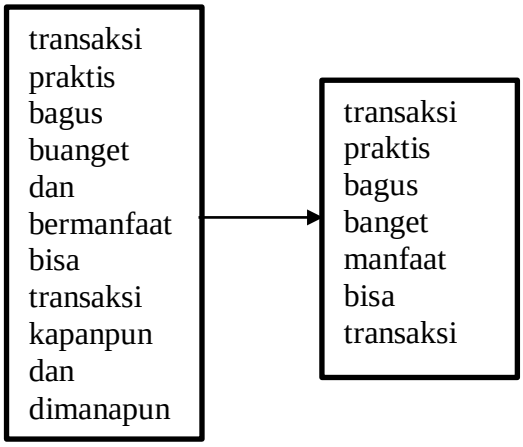
Merupakan tahapan untuk memotong dokumen menjadi bagian-bagian kecil berdasarkan tiap kata yang menyusunnya atau disebut dengan *token* dan disertai tahapan untuk membuang karakter tertentu seperti tanda baca (Manning, Raghavan, dan Schutze, 2009). Berfungsi untuk menjadikan sebuah kalimat menjadi lebih bermakna. Tahap yang dilakukan pertama kali adalah normalisasi kata dengan mengubah semua karakter huruf menjadi huruf kecil (*lowercase)*. Proses tokenisasi diawali dengan menghilangkan simbol dan tanda baca yang ada pada teks tersebut seperti @, $, &, tanda titik (.), koma (,) tanda tanya (?), tanda seru (!). Tahap selanjutnya yaitu proses penguraian teks yang semula berupa kalimat-kalimat yang berisi kata-kata. Proses pemotongan string berdasarkan tiap kata yang menyusunnya, umumnya setiap kata akan terpisahkan dengan karakter spasi, proses tokenisasi mengandalkan karakter spasi pada dokumen teks untuk melakukan pemisahan. Hasil dari proses ini adalah kumpulan kata saja (Putri, 2016). Untuk gambaran tahap *tokenizing*, dapat dilihat pada Gambar 2.5 berikut:



Gambar 2.5 *Tokenizing*
(Firmansyah, dkk, 2017)

2.2.1.4 *Filtering*

Merupakan tahapan untuk mengambil kata-kata penting dari hasil *tokenizing* menggunakan algoritma *stopword removal* (membuang kata yang kurang penting). Kata penghubung (*Stopword)* didefinisikan sebagai sebuah kata yang sangat sering muncul dalam suatu dokumen teks yang kurang memberikan arti penting terhadap isi dokumen (Patel & Shah, 2013). Sehingga kata-kata tersebut dapat dibuang dan hanya menyisakan kata-kata penting untuk dapat memiliki arti yang akan diproses ke tahap berikutnya. Contohnya adalah “yang”, “dan”, “di”, “dari” dan sebagainya (Putri, 2016). Gambaran proses *filtering* dapat dilihat pada Gambar 2.6 berikut:



Gambar 2.6 *Filtering*
(Firmansyah, dkk, 2017)

2.2.2 *Feature Selection*

Tahap ini merupakan tahapan lanjutan dari tahapan *filtering* bertujuan untuk mengurangi dimensi dari suatu kumpulan teks, atau dengan kata lain menghapus kata-kata yang dianggap tidak penting atau tidak menggambarkan isi dokumen sehingga proses pengklasifikasian menjadi lebih efektif dan akurat contohnya adalah proses *stemming* (Feldman & Sanger, 2007., Berry & Kogan 2010). *Stemming* adalah proses pemetaan dan penguraian berbagai bentuk dari suatu kata menjadi bentuk kata dasarnya (*stem*) (Tala, 2003). Tujuan *stemming* adalah menghilangkan imbuhan-imbuhan baik yang ada pada setiap kata sehingga akan menyisakan kata dasar. Jika imbuhan tersebut tidak dihilangkan maka setiap satu kata dasar akan disimpan dengan berbagai macam bentuk yang berbeda sesuai dengan imbuhan yang melekatinya sehingga hal tersebut akan menambah beban *database*. Kata-kata yang dinilai penting dilihat

berdasarkan intensitas kemunculan dan yang paling informatif dari keseluruhan.

2.2.2.1 Pembobotan Kata (*Term Weighting)*

Tahap selanjutnya yaitu pembobotan. Hal yang perlu diperhatikan dalam pencarian informasi dari dokumen yang heterogen (bervariasi) adalah pembobotan *term*. Tahap ini bertujuan untuk memberi nilai frekuensi suatu kata sebagai bobot. *Term* dapat berupa kata, frase atau unit hasil *indexing* lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui konteks dari dokumen tersebut. Karena setiap kata dalam dokumen memiliki tingkat kepentingan masing-masing, maka untuk setiap kata tersebut diberikan sebuah indikator, yaitu *term weight* (Zafikri, 2008). Menurut Zafikri (2008) *term weighting* atau pembobotan kata dipengaruhi oleh hal-hal di antaranya:

1. ***Term Frequency (TF)***

   Merupakan frekuensi kemunculan sebuah kata (*term)* dalam sebuah dokumen. Semakin besar jumlah *term* yang muncul (TF tinggi) maka semakin besar bobot dokumen atau memberikan nilai kesesuain yang semakin besar (Informatikalogi, 2016). Menurut (Zafikri, 2008) terdapat beberapa jenis formula yang dapat digunakan pada *Term Frequency* (TF), yaitu:

a. TF biner (*binary TF*)

Hanya memperhatikan apakah suatu kata ada atau tidak dalam dokumen. Jika ada akan diberi nilai satu, tetapi jika tidak diberi nilai nol.

b. TF murni (*raw TF*)

Nilai TF yang diberikan berdasarkan jumlah kemunculan suatu kata dalam dokumen. Contohnya, jika muncul lima kali maka kata tersebut akan bernilai lima.

c. TF logaritmik

TF ini untuk menghindari dominansi dokumen yang mengandung sedikit kata dalam *query*, namun mempunyai frekuensi yang tinggi.

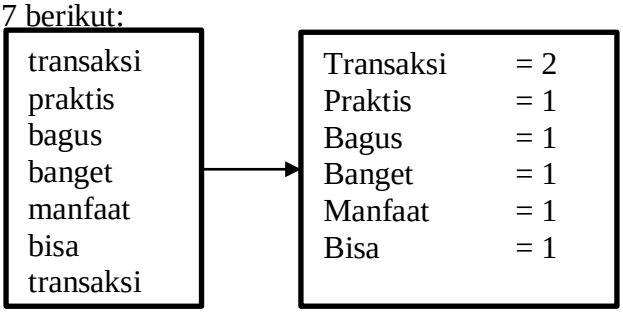
$$tf = 1 + \log(tf) \qquad 2.15$$

d. TF normalisasi

Menggunakan perbandingan antara frekuensi sebuah kata dengan jumlah keseluruhan kata pada dokumen.

$$tf = 0{,}5 + 0{,}5x(\frac{tf}{\max tf}) \qquad 2.16$$

Contoh proses pembobotan TF murni (*raw TF)* dapat dilihat pada Gambar 2.7 berikut:



Gambar 2.7 TF *review* bagus

Gambar 2.7 di atas merupakan contoh kalimat *review* yang mengalami proses perhitungan *term frequency* dimana *review* berikut memberikan emosi senang (puas).

2. *Inverse Document Frequency* (IDF)

Inverse Document Frequency (IDF) merupakan metode statistik numerik yang menghitung seberapa pentingnya kata dalam sebuah dokumen dimana dalam konteks ini yaitu pengurangan dominansi *term* yang sering muncul di berbagai dokumen. Metode ini digunakan sebagai bobot dalam pencarian informasi dalam *text mining* (Fanani, 2017). Hal ini diperlukan karena *term* yang banyak muncul di berbagai dokumen, dapat dianggap sebagai *term* umum, sehingga menjadi tidak penting nilainya. Sebaliknya kata yang jarang muncul dalam dokumen harus diperhatikan dalam pemberian bobot. Menurut (Witten, 1999) Kata yang muncul pada sedikit dokumen harus dipandang sebagai kata yang lebih penting (*uncommon terms*) daripada kata yang muncul pada banyak dokumen. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kata (*Inverse Document Frequency*). Formula IDF dapat dituliskan sebagai berikut:

$$idf_j = log\left(\frac{D}{df_j}\right) \qquad 2.17$$

Keterangan:

D : jumlah semua dokumen dalam koleksi

df_j : jumlah dokumen yang mengandung *term* t_j

Maka, rumus umum untuk TF-IDF adalah penggabungan dari formula perhitungan *raw TF* dan formula IDF dengan cara mengalikan nilai keduanya, seperti berikut:

$$w_{ij} = tf_{ij} \times idf_j \quad 2.18$$

$$w_{ij} = tf_{jj} \times log\left(\frac{D}{df_j}\right) \quad 2.19$$

Keterangan:

w_{ij} : Bobot term t_j terhadap dokumen d_i

tf_{ij} : Jumlah kemunculan term t_j dalam dokumen d_i

D : Jumlah semua dokumen dalam koleksi

df_i : Jumlah dokumen yang mengandung term t_j (minimal ada satu kata yaitu term t_j)

2.3 Analisis Sentimen

Analisis sentimen atau sering disebut dengan *opinion mining* adalah bidang ilmu data mining yang bertujuan untuk menganalisis, memahami, mengolah dan mengekstrak data teks yang berupa opini, sentimen, evaluasi, sikap, dan emosi terhadap suatu entitas seperti produk, servis, organisasi, individu, dan topik-topik tertentu. Klasifikasi sentimen digunakan untuk menyelesaikan masalah klasifikasi dua kelas, positif dan negatif. Data pengujian yang digunakan biasanya adalah *reviews*/*review* produk secara *online*. Karena *review online* memiliki nilai penilaian yang ditetapkan oleh

para peneliti. Sebagian besar penelitian tidak menggunakan kelas netral, dengan tujuan agar memudahkan masalah klasifikasi, namun juga tidak jarang peneliti yang menggunakan kelas netral, misalnya penggunaan peringkat bintang tiga sebagai kelas netral (Liu, 2012).

Pengaruh dan manfaat analisis sentimen menyebabkan penelitian mengenai analisis sentimen semakin berkembang pesat. Di Amerika sekitar 20 s.d. 30 perusahaan memfokuskan pada layanan analisis sentimen (Liu, 2012). Manfaat analisis sentimen dalam dunia usaha di antaranya dapat melakukan pemantauan terhadap suatu produk. Dimana secara cepat dapat digunakan sebagai alat bantu untuk melihat respon dan *review* masyarakat terhadap suatu produk tertentu apakah positif atau negatif di web, sehingga dapat segera diambil langkah-langkah maupun kebijakan strategis lainnya untuk memajukan perusahaan atau menghindari masalah. Hal ini memungkinkan bisnis untuk melacak:

a. Deteksi Flame (rants buruk)

b. Persepsi produk baru.

c. Persepsi Merek.

2.4 Evaluasi Sistem Klasifikasi

Evaluasi terhadap suatu klasifikasi umumnya dilakukan menggunakan sebuah himpunan data uji, yang tidak digunakan dalam pelatihan klasifikasi tersebut, dengan suatu ukuran tertentu. Sebuah sistem klasifikasi harus dinilai performanya agar dapat mengukur tingkat akurasi dari prediksi

klasifikasi yang dihasilkan. Terdapat beberapa metode perhitungan yang digunakan untuk menilai performa sebuah klasifikasi misalnya *Accuracy*, *Precision, Recall,* dan lain-lain.

Terdapat empat kemungkinan yang dapat terjadi dari hasil klasifikasi suatu data pada evaluasi klasifikasi. Jika data positif dan diprediksi positif, maka akan dihitung sebagai *true positive*. Jika data positif diprediksi negatif, maka akan dihitung sebagai *false negative.* Kemudian pada data negatif jika diprediksi negatif akan dihitung sebagai *true negative* dan jika diprediksi positif maka akan dihitung sebagai *false positive* (Fawcett, 2006). Untuk lebih jelasnya dapat dilihat pada Tabel *Confusion Matrix* berikut:

Tabel 2.1 *Confusion Matrix*

Prediction	*Actual*	
Class	***Positive***	***Negative***
Positive	*True Positive (TP)*	*False Positive (FP)*
Negative	*False Negative (FN)*	*True Negative (TN)*

2.4.1 *Accuracy*

Accuracy adalah jumlah proporsi prediksi data yang benar, serta merupakan tingkat kedekatan antara nilai prediksi dengan nilai akurasi. Jika nilai akurasi tinggi maka sebuah sistem akan semakin baik dalam melakukan prediksi. *Accuracy* dapat dihitung dengan persamaan 2.20 berikut (Lim, dkk, 2006) :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad 2.20$$

2.4.2 *Precision*

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban sistem. Rumus *precision* dapat dilihat pada persamaan 2.21 berikut (Lim, dkk, 2006):

$$Precision = \frac{TP}{TP+FP} \quad 2.21$$

2.4.3 *Recall*

Recall adalah salah satu perhitungan keakuratan prediksi yang digunakan sebagai ukuran tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Recall* dapat dihitung melalui persamaan 2.22 berikut (Lim, dkk, 2006):

$$Recall = \frac{TP}{TP+FN} \quad 2.22$$

2.5 Akurasi Terbaik

Menurut (Sasongko, 2016), standar tabel kategori pengklasifikasian berdasarkan nilai akurasi dapat dilihat pada Tabel berikut:

Tabel 2.2 Kategori Klasifikasi berdasarkan Nilai Akurasi

Nilai Akurasi	Kategori Klasifikasi
91-100%	Excellent (Sangat baik)
81-90%	Good (Baik)
71-80%	Cukup Baik
61-70%	Poor (Buruk)
51-60%	Fail (Sangat buruk)

2.6 *E-Commerce*

2.6.1 Definisi *E-commerce*

E-commerce atau elektronik *commerce* (perdagangan secara elektronik) menurut (Turban, dkk, 2012) merupakan transaksi bisnis yang terjadi dalam jaringan elektronik seperti internet. Siapapun yang dapat mengakses komputer, memiliki sambungan ke internet dan memiliki cara untuk membayar produk-produk atau jasa yang dibeli, dapat berpartisipasi dalam *e-commerce*. Sedangkan menurut Berkatulloh dan Prasetyo (2005) menjelaskan bahwa *e-commerce* adalah kegiatan-kegiatan bisnis yang menyangkut konsumen (*consumers*), manufaktur (*manufaktures*), *service providers* dan pedagang perantara (*intermediaries*), dengan menggunakan jaringan-jaringan komputer (*computer networks*) yaitu internet.

2.6.2 Komponen *E-commerce*

E-commerce memiliki mekanisme-mekanisme tertentu yang unik dan berbeda dibanding dengan *traditional commerce.* Menurut Turban & King (2002), *e-commerce* memiliki komponen-komponen di antaranya:

2.6.2.1 *Customer*

Customer atau konsumen merupakan para pengguna Internet yang dapat dijadikan sebagai target pasar yang potensial untuk diberikan penawaran berupa produk, jasa, atau informasi oleh para penjual.

2.6.2.2 *Seller*

Seller atau penjual merupakan pihak yang menawarkan produk, jasa, atau informasi kepada para konsumen baik individu maupun organisasi. Proses penjualan dapat dilakukan secara langsung melalui *website* yang dimiliki oleh penjual tersebut melalui *marketplace*.

2.6.2.3 *Product*

Salah satu perbedaan antara *e-commerce* dengan *traditional commerce* terletak pada produk yang dijual. Pada dunia maya, penjual dapat menjual produk digital yang dapat dikirimkan secara langsung melalui Internet.

2.6.2.4 Infrastruktur

Infrastruktur pasar yang menggunakan media elektronik meliputi perangkat keras, perangkat lunak dan juga sistem jaringannya. Seperti *smartphone, personal computer* maupun laptop.

2.6.2.5 *Front end*

Front end merupakan aplikasi web yang dapat berinteraksi dengan pengguna secara langsung. Beberapa proses bisnis pada *front end* antara lain portal penjual, katalog elektronik, *shopping cart*, mesin pencari dan *payment gateway*.

2.6.2.6 *Back end*

Back end merupakan aplikasi yang secara tidak langsung mendukung aplikasi *front end.* Semua aktivitas yang berkaitan dengan

pemesanan barang, manajemen inventori, proses pembayaran, *packaging* dan pengiriman barang termasuk dalam bisnis proses *back end.*

2.6.2.7 *Intermediary*

Intermediary merupakan pihak ketiga yang menjembatani antara produsen dengan konsumen. *Online intermediary* membantu mempertemukan pembeli dengan penjual, menyediakan infrastruktur, serta membantu penjual dan pembeli dalam menyelesaikan proses transaksi. *Intermediary* tidak hanya perusahaan atau organisasi tetapi dapat juga individu. Contoh *intermediary* misalnya *broker* dan distributor.

2.6.3 Jenis-jenis *E-commerce*

Berdasarkan karakteristiknya, menurut id.techinasia.com (dikutip dalam Aprilia, 2017) *e-commerce* dapat dibedakan menjadi beberapa jenis, di antaranya:

2.6.3.1 *Business to Business* (B2B)

Karakteristik yang dimiliki oleh *business to business* sebagai berikut:

1. Model penjualan yang terjadi antara pelaku bisnis dengan pelaku bisnis lainnya.
2. Pertukaran data dapat dilakukan secara berulang-ulang dan berkala sesuai dengan format data yang telah disepakati bersama.

3. Salah satu pihak tidak harus menunggu rekan lainnya untuk mengirimkan data.

Contohnya adalah IndoTrading.com, Indonetwork, MBiz, web *hosting* ke web *agency* (Softwareseni, 2018).

2.6.3.2 *Business to Consumer* (B2C)

Business to Consumer memiliki karakteristik:

1. Model penjualan antara pelaku bisnis dengan konsumen.
2. Terbuka untuk umum dimana informasi disebarkan secara umum dan dapat diakses secara bebas.
3. Servis yang digunakan bersifat umum, sehingga dapat digunakan oleh banyak orang.
4. Servis yang digunakan berdasarkan permintaan. Produsen harus siap memberikan respon sesuai dengan permintaan konsumen.
5. Sering dilakukan sistem pendekatan *client-server*.
6. Terjadi pada pelanggan, perusahaan penjual jasa, perusahaan retail online

Contohnya adalah berjualan makanan, jasa laundry, ojek bahkan salon (Softwareseni, 2018).

2.6.3.3 *Consumer to Consumer* (C2C)

Interaksi antara konsumen satu dengan lainnya dapat terjadi secara langsung dan mudah. Dalam C2C konsumen dapat menjual maupun membeli produk secara langsung kepada konsumen

lainnya. Contohnya adalah ketika ada seseorang yang melakukan penjualan di *classified ads* (misalnya, www.classified2000.com) dan menjual properti rumah hunian, mobil, dan sebagainya, mengiklankan jasa pribadi di internet serta menjual pengetahuan dan keahlian merupakan contoh lain C2C. Contoh lain yang sangat terkenal yaitu eBay.com yang merupakan perusahaan lelang. Selain melalui *marketplace*, kegiatan jual beli juga dapat dilakukan secara langsung antar individu tanpa adanya pihak ketiga.

Contohnya adalah Bukalapak, Shopee, Tokopedia, OLX, Blibli, Jd.id, Tokopedia, Kaskus hingga Instagram (Softwareseni, 2018).

2.6.3.4 *Consumer to Business* (C2B)

Customer to Business merupakan model bisnis dimana konsumen (individu) menciptakan nilai, dan perusahaan mengonsumsi nilai tersebut. Sebagai contoh, ketika konsumen menulis *review*, atau ketika konsumen memberikan ide yang berguna untuk pengembangan produk baru, maka konsumen (individu) ini sudah menciptakan nilai bagi perusahaan, jika perusahaan tersebut menerima dan mengaplikasikan input tersebut. Sebagai contoh, Priceline.com merupakan situs yang memungkinkan seseorang menjual produk kepada perusahaan yang siap membelinya. Dalam hal ini, internet dapat digunakan sebagai sarana negosiasi (Softwareseni, 2018).

2.7 Tokopedia

2.7.1 Profil Perusahaan

Tokopedia secara resmi diluncurkan ke publik pada 17 Agustus 2009 di bawah naungan PT Tokopedia yang didirikan oleh William Tanuwijaya dan Leontinus Alpha Edison pada 6 Februari 2009 (http://republikpos.com/2016/10/profilperusahaan-tokopedia-dan-kisah-pendirinya). Diakses pada 16 Juni 2020.

Tokopedia merupakan salah satu perusahaan jual beli berbasis digital (*e-commerce)* terbesar di Indonesia dengan pertumbuhan pesat yang mengusung model bisnis *marketplace* dan *mall online*, serta memungkinkan setiap individu, toko kecil dan *brand* untuk membuka dan mengelola toko *online* masing-masing. Visi Tokopedia adalah "Membangun Indonesia lebih baik, lewat internet". Sedangkan misinya adalah selalu positif, memecahkan masalah, menjadi yang terbaik, generasi Indonesia yang lebih baik lagi dan fokus pada pelanggan.

1. Logo dan Maskot

Tokopedia memiliki logo dan maskot yang didominasi dengan warna hijau karena identik dengan bumi yang merupakan lambang dari kerendahan hati dan ketenangan. Logo tokopedia adalah tulisan berwarna hijau yang bertuliskan "tokopedia" seperti Gambar 2.9 berikut:

tokopedia

Gambar 2.8 Logo Tokopedia

Sedangkan, maskot dari Tokopedia adalah "Toped" (burung hantu berwarna hijau) sebagai simbol kecerdasan dan kebijaksanaan, serta

burung hantu memiliki kemampuan untuk melihat ke semua arah. Maskot Tokopedia dapat dilihat pada Gambar 2.10 berikut:

Gambar 2.9 Maskot Tokopedia

2. Layanan Tokopedia

Tokopedia memberikan layanan kepada pelanggannya untuk dapat bertransaksi secara lebih aman dan bebas penipuan karena pembayaran di Tokopedia akan diteruskan kepada pihak penjual setelah barang diterima didukung dengan fasilitas rekening bersama secara gratis.

Disisi lain, saat ini Tokopedia pun sudah terhubung ke berbagai perusahaan logistik terbesar di Indonesia. Sehingga, ongkos kirim dan *tracking* pesanan pun dapat dilakukan secara mudah, otomatis dan *real time*. Tidak hanya pedagang perorangan, Tokopedia pun telah bekerjasama dengan beberapa perusahaan retail besar (*official* store) untuk memasarkan produk mereka secara *online* seperti P&G, Lotte Mart, Nissin, Ace Indonesia, Adidas, Ramayana, Century, Oppo, Samsung, Smartfren, Mustika Ratu serta masih banyak lagi (Tokopedia, 2020).

3. Jenis Produk Tokopedia

Produk pada etalase Tokopedia merupakan produk yang benar-benar akan dijual. Dan hanya menjual barang, bukan berupa jasa. Selain

jual beli produk, Tokopedia juga memberikan layanan untuk dapat melakukan transaksi seperti *top up* donasi, pemesanan dan pembayaran tiket kereta api serta pesawat, angsuran kredit, pembayaran air PDAM, iuran BPJS, pembelian token listrik PLN, pembelian *voucher game*, *gift card*, pulsa maupun paket data, serta masih banyak lagi layanan yang diberikan.

2.8 *Google Play*

Google Play merupakan layanan konten digital toko aplikasi *online* milik *Google* yang menawarkan berbagai macam produk seperti music, buku, dan aplikasi yang dapat diakses melalui web, aplikasi android (*Play Store)* dan *Google TV*. G*oogle play* terdapat fitur *reviews* yang berisi *review* dari para pengguna. Berikut logo *Google Play:*

Gambar 2.10 Logo *Google Play*

2.9 *Review*

Pardiyono (2007) menyatakan bahwa teks *review* adalah teks yang berisi pemberian kritik, evaluasi, atau melakukan *reviews* terhadap karya cipta intelektual yang bertujuan untuk memberikan kritikan hasil evaluasi, atas suatu karya ilmiah, buku, atau karya seni. Fitur *review* yang diberikan oleh aplikasi-aplikasi dapat diberikan dan dilihat secara *online* pada suatu sumber yang terkoneksi pada sebuah jaringan komputer yang bersifat *online*.