BAB 2
LANDASAN TEORI

2.1. Analisis Sentimen Berbasis Aspek

Analisis sentimen level entitas atau aspek disebut juga dengan analisis sentimen berbasis fitur (*feature-based opinion mining*) melakukan analisis sentimen dengan lebih spesifik terhadap target opininya. Analisis sentimen berbasis aspek (*aspect-based sentiment analysis*) ditujukan untuk menentukan polaritas sentimen (misalnya positif, negatif atau netral) dari sebuah kalimat atau teks yang diekspresikan untuk sebuah target yang merupakan aspek dari suatu entitas tertentu. Sistem sentimen analis berbasis aspek menerima masukan berupa serangkaian teks yang membahas entitas tertentu, kemudian berupaya mendeteksi aspek atau fitur utama entitas dan memperkirakan sentimen rata-rata teks per aspek [3].

Entitas dalam analisis sentimen berbasis aspek adalah obyek yang dapat berupa produk, layanan, topik, isu, orang, organisasi atau even sedangkan aspek adalah bagian atau atribut dari entitas. Untuk memahami istilah-istilah dalam analisis sentimen berbasis aspek diberikan contoh opini dengan domain laptop berikut ini: ”*I bought an iPhone a few days ago. It was a nice phone. The touch screen was really cool*”. Dalam opini tersebut, “*iPhone*” adalah entitas dan “*touch screen*” adalah atribut atau aspek dari entitas tersebut. Sedangkan “*nice*” adalah kata opini untuk target entitas “*iPhone*” dan “*really cool*” adalah kata opini untuk aspek “*touch screen*”.

2.2. *Web Scrapping*

Web scrapping merupakan kegiatan pengumpulan data yang bersumber dari internet dengan tujuan mendapatkan data untuk kemudian melakukan ekstraksi informasi yang dimiliki oleh data tersebut. Cara kerja *web scrapping* adalah dengan mengakses halaman *Web*, memilih elemen data yang ada dalam halaman tersebut, melakukan ekstraksi dan transformasi bila diperlukan, dan terakhir menyimpan data tersebut menjadi dataset terstruktur [10].

2.3. Kata Baku

Menurut kaidah penulisan kata bahasa Indonesia, penggunaan kata baku dalam karya tulis ilmiah sangatlah penting. Kata baku biasanya digunakan dalam kalimat formal atau dalam berbagai bahasa standar, termasuk bahasa lisan dan tulisan [21]. Kata baku bahasa Indonesia juga memiliki ciri-ciri sebagai berikut:

1. Baik lisan maupun tulisan, kata baku digunakan dalam acara resmi, seperti surat resmi, undang undang, karya ilmiah, laporan penelitian, dan lainnya. Variasi bahasa yang baku tidak akan terganggu oleh dialek atau aksen tertentu.
2. Baik lisan maupun tulisan, kata baku menggunakan istilah yang berlaku dalam Pedoman Ejaan Umum Bahasa Indonesia (PUEBI).
3. Baik lisan maupun tulisan, varietas standar secara jelas dan lengkap menjalankan fungsi gramatikal subjek, predikat, dan objek.

Sebuah kata dapat disebut kata yang tidak baku jika kata yang digunakan tidak sesuai dengan aturan bahasa Indonesia. Kata tidak baku ini sering muncul dalam kehidupan seharihari kita. Beberapa contoh kesalahan ejaan bahasa Indonesia dalam menulis kata tidak baku sebagai berikut:

1. Kata tidak baku: tehnik, sedangkan kata baku: teknik.
2. Kata tidak baku: sepak bola, sedangkan kata baku: sepakbola.
3. Kata tidak baku: apotik, sedangkan kata baku: apotek.

2.4. *Typo*

Kesalahan penulisan atau disebut *typo* merupakan suatu *human error* (kesalahan manusia) yang cukup sering dilakukan oleh seseorang dalam melakukan sebuah penulisan [23]. *Typo* dapat dikategorikan menjadi dua jenis, yaitu kesalahan *non-word* dan kesalahan *real-word*. Kesalahan *non-word* adalah kesalahan yang tidak memiliki arti dalam kamus, sedangkan kesalahan *real-word* adalah kata yang tertulis benar atau memiliki arti dalam kamus, tetapi tidak dimaksudkan dalam kalimat dan memiliki arti yang berbeda.

2.5. *Text Pre-Processing*

Text preprocessing merupakan tahap persiapan data berupa teks sebelum diproses ke tahap selanjutnya, proses ini bertujuan untuk membuat data menjadi lebih terstruktur dan siap diolah ada beberapa tahap para proses *preprocessing* ini sendiri [9]. Pada penelitian ini tahapan yang dilakukan meliputi *Case Folding, Cleaning, Tokenizing, Normalization, Stemming, Convert Negation, Stopword Removal*

2.5.1 *Case Folding*

Pada tahapan *Case Folding*, sistem akan menyeragamkan huruf ke dalam bentuk yang sama. Semua huruf akan diubah menjadi *lowercase* atau huruf kecil [10]. Dalam hal ini hanya menerima huruf latin dari a hingga z. Dimana jika ditemukan dalam suatu dokumen akan terdapat beberapa huruf saja memiliki huruf kapital seperti awalan kalimat, nama orang, nama kota, dll.

2.5.2 *Cleaning*

Proses *Cleaning* dilakukan untuk mengurangi *noise*, maka dilakukan pembersihan kata, tanda baca atau simbol yang tidak diperlukan pada proses hasil klasifikasi sentimen berbasis aspek [10]. Karena dalam opini pariwisata pantai Malang Selatan pasti terdapat beberapa atribut yang tidak terpakai. Proses yang dilakukan yaitu terjadi pengubahan menjadi spasi pada angka, tanda baca, serta simbol-simbol seperti ‘@’ untuk username, hastag (#), alamat website, mention, link, emoticon, spasi berlebih.

2.5.3 *Tokenizing*

Tokenize atau Tokenisasi merupakan tahapan untuk mengubah struktur teks ke dalam bentuk token [10]. Data yang awalnya berbentuk kalimat diubah menjadi bentuk kata perkata. Pada tahap ini kalimat akan dipecah berdasarkan spasi menjadi kata-kata tunggal.

2.5.4 *Normalization*

Normalization merupakan proses yang dilakukan untuk mengubah kata-kata yang tidak baku menjadi kata-kata yang baku [12]. Pada tahap ini menggunakan

normalisasi kata dengan kamus *slang words* dan singkatan serta normalisasi kata dengan *Peter Norvig.*

2.5.4.1. Kamus *Slang Words* dan Singkatan (SS)

Normalisasi kata menggunakan pencocokan dengan kamus SS bertujuan untuk memperbaiki kata yang tidak baku menjadi baku dan memperbaiki kata yang disingkat [14]. Kamus yang digunakan akan disimpan ke dalam format .csv lalu diberi nama *slang words* dan singkatan. Kamus tersebut berisi kata-kata tidak baku dan kata singkatan beserta kata. Normalisasi ini dilakukan dengan cara tiap kata yang ada akan diperiksa dengan kamus, jika terdapat kata yang sama pada kamus maka kata tersebut akan digantikan dengan kata perbaikannya.

2.5.4.2. Peter Norvig Spelling Corrector

Peter Norvig Spelling Corrector merupakan sebuah algoritma yang mampu mengubah kata yang mengandung kesalahan ejaan, lalu dicarikan kandidat-kandidat kata yang benar melalui proses pemisahan kata menjadi dua bagian dan sejumlah suntingan yang mengubah bentuk kata ke suatu bentuk ke bentuk yang lain. Pendekatan kata pada metode *Peter Norvig* dapat menghasilkan semua kemungkinan kata dengan semua operasi *edit-distance* yaitu operasi penambahan (*insert*), operasi penggantian (*replace*), operasi penukaran (*transpose*), operasi penghapusan (*delete*) dari kata yang terdeteksi *typo* dan mencarinya dalam kamus [8]. Proses operasi diterapkan untuk semua huruf pada kata yang salah ejaan secara bergantian.

Setelah semua operasi mendapatkan kandidat kata, selanjutnya menghitung peluang masing-masing kandidat kata yang cocok dengan korpus. *Peter Norvig Spelling Corrector* memiliki fungsi korektor kata untuk memilih koreksi ejaan terdekat c untuk kata w. Tidak ada kandidat yang dipilih secara mutlak karena hanya berupa saran perbaikan kata menggunakan probabilitas. Algoritma ini mencoba menemukan koreksi c dari semua kemungkinan koreksi kandidat yang memaksimalkan probabilitas bahwa c adalah koreksi yang ditujukan dengan kata

asli w [21]. Rumus untuk perhitungan *Peter Norvig* dapat dilihat pada persamaan (2.1).

$$correction(w) = \text{argmax}_{c \in candidates} P(c|w) \quad (2.1)$$

Dimana:

argmax = pemilihan kandidat yang memiliki probabilitas tertinggi.

c∈candidates = kandidat kata c dari kumpulan kandidat.

P(c) = probabilitas kandidat c muncul pada sebuah corpus dokumen.

P(w|c) = menunjukkan probabilitas bahwa kata w adalah teks yang dimaksud pada kandidat c.

Dengan Teorema Bayes, ini setara dengan persamaan (2.2).

$$correction(w) = \text{argmax}_{c \in candidates} \frac{P(c)P(c|w)}{P(w)} \quad (2.2)$$

Karena P(w) adalah sama untuk setiap kandidat c yang mungkin, maka dapat memfaktorkannya ke persamaan (2.3)

$$correction(w) = \text{argmax}_{c \in candidates} P(c)\ P(w|c) \quad (2.3)$$

Misalnya ada salah ketik “Sya” (kata yang benar: “Saya”), metode akan mencari kandidat kata yang mendekati kata sebenarnya dengan menggunakan model kandidat seperti *splits, deletion, transpostion, substitution* dan *insertion* yang akan menghasilkan: “Say”, “Sy”, “Sya”, “ayS”, “Saya”, “eSy”, dan “aSya”. Bahkan kata tersebut akan digabungkan dengan karakter “a” sampai “z”. Ketika kata kandidat sudah menghitung probabilitas dalam korpus, maka kata salah ketik “Sya” paling dekat dengan kata “Saya” dalam korpus. Proses komputasi pada metode ini akan terus berjalan hingga ditemukan koreksi kata terdekat.

2.5.4.2.1. Korpus

Korpus merupakan kumpulan teks autentik, baik tulis maupun transkrip yang disimpan di komputer, dan dianalisis menggunakan perangkat lunak yang dirancang untuk analisis korpus [17]. Korpus dapat digunakan untuk mengumpulkan data kebahasaan di internet khususnya pada media massa daring kemudian data tersebut dapat dianalisis. Contoh kumpulan teks untuk diolah oleh korpus adalah buku, jurnal internasional, majalah, koran, artikel, dll.

2.5.4.2.2. Metode N-Gram

Penggunaan N-gram telah banyak digunakan untuk berbagai masalah. Seperti prediksi kata, koreksi ejan, pengenalan suara, koreksi kataterjemahan dan pencarian string. Metode N-gram mengambil potongan karakter kata dengan jumlah n dalam sebuah kalimat [8]. N-gram dapat dibedakan berdasarkan sejumlah n, nilai n = 1 adalah unigrams, n = 2 adalah bigrams.

Contoh pemenggalan kalimat metode N-gram dengan contoh kalimat “harus tetap waspada” sebagai berikut:

Unigrams : harus, tetap, waspada

Bigrams : harus tetap, tetap waspada

Probabilitas N-Gram menggunakan *Maximum Likehood Estimation* (MLE) [24] merupakan pengambilan asumsi kemunculan kata bergantung pada kemunculan kata sebelum dan kemunculan sesudahnya dalam suatu kalimat dengan menghitung jumlah kemunculan N-Gram pada korpus lalu membaginya dengan nilai total agar nilainya berada di antara 0 dan 1. Dapat dilihat pada persamaan (2.4).

$$P_1\left(W^i|W^{i-1}\right) = \frac{count\left(W^{i-1}W^i\right)}{\sum_m count\left(W^{i-1}W\right)} \tag{2.4}$$

Dengan menggunakan *Maximum Likehood Estimation* (MLE) diatas, maka dihasilkan rumus-rumus yang dapat dilihat pada persamaan (2.5), dan (2.6).

Probabilitas dari unigram c_j^i

$$P_1\left(c_j^i\right) = \frac{count\left(c_j^i\right)}{\sum_r^{k_i} count\left(c_r^i\right)} \quad (2.5)$$

$$P_2\left(c_j^i | W^{i-1}\right) = \frac{count\left(W^{i-1}\, c_j^i\right)}{\sum_r^{k_i} count\left(W^{i-1}\, c_r^i\right)} \quad (2.6)$$

Dimana:

P_1 = Probabilitas untuk unigram
P_2 = Probabilitas untuk bigram

2.5.5 *Stemming*

Stemming merupakan tahapan untuk mengubah kata menjadi kata dasar, dengan menggunakan aturan tertentu [10]. Proses stemming bahasa Indonesia dilakukan dengan menghilangkan *sufix*, *prefix*, dan *konfiks* pada dokumen.

2.5.6 *Convert Negation*

Convert Negation merupakan tahapan yang dilakukan jika terdapat sebuah kata negasi seperti ”tak”, ”tidak”, ”bukan”, ”tanpa” [25]. Tahapan ini dilakukan karena kata negasi dapat mengubah makna. Jika ditemukan kata negasi maka akan digabungkan dengan kata selanjutnya.

2.5.7 *Stopword Removal*

Stopword Removal merupakan tahapan untuk menghilangkan kata yang tidak sesuai dengan topik dokumen dengan menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting) [10].

2.6. Pembobotan TF-IDF

Data yang sudah melalui *preprocessing* perlu diubah menjadi bentuk numerik, sehingga perlu dilakukannya pembobotan kata [9]. TF-IDF (*Term frequency–inverse document frequency*) adalah proses untuk mentransformasi data tekstual menjadi numerik untuk pembobotan setiap kata. TF-IDF juga merupakan stastistik numerik untuk mencerminkan pentingnya sebuah kata dalam

dokumen. Metode ini menggunakan frekuensi kemunculan kata di dalam dokumen. Konsep metode ini mempunyai kelemahan yaitu kata dianggap selalu setara yang menyebabkan relevansi kata menjadi tinggi ketika sering muncul dalam sebuah dokumen.

Dalam sebuah dokumen umumnya ada beberapa kata yang tidak penting, maka dari itu diperlukan pembobotan *document frequency* untuk menghitung jumlah atau frekuensi dokumen yang didalamnya terdapat term atau kata. Dengan nilai term yang ditemukan pada setiap dokumen, dilakukan kebalikan dari proses *document frequency* yaitu proses *inverse document frequency*. Proses ini memunculkan bobot yang tinggi terhadap kata yang jarang muncul pada setiap dokumen. IDF berfungsi untuk mengurangi bobot suatu term jika kemunculannya dari term tersebut banyak dan tersebar diseluruh dokumen.

Metode yang digunakan untuk pembobotan kata TF-IDF dapat dilihat pada rumus persamaan (2.7) berikut:

$$TF * IDF(d,t) = TF(d,t) * log(\frac{N}{df_t}) \quad (2.7)$$

Keterangan :

$TF * IDF(d,t)$: Pembobotan TF-IDF

$TF(d,t)$: Frekuensi munculnya term t pada dokumen d

N : Jumlah dari semua kumpulan dokumen

df_t : Jumlah dari dokumen yang mengandung term t

2.7. *Feature Selection*

Penerapan metode *feature selection* digunakan untuk mengurangi dimensi dari set fitur dengan menghapus fitur yang tidak relevan [16]. Fitur yang tidak relevan adalah yang tidak memberikan informasi bermanfaat tentang data, dan fitur yang berlebihan merupakan fitur yang memberikan informasi lebih banyak

daripada fitur yang saat ini dipilih. Dengan kata lain, fitur berlebihan memberikan informasi yang berguna tentang kumpulan data, namun informasinya telah disediakan oleh fitur yang sudah dipilih sebelumnya. Fitur seleksi mempunyai keunggulan seperti ukuran dataset yang lebih kecil, menyusutkan ruang pencarian, dan kebutuhan komputasi yang rendah.

2.8. Information Gain

Information Gain (IG) merupakan salah satu metode untuk seleksi fitur yang banyak yang banyak digunakan oleh peneliti untuk menentukan batas dari kepentingan sebuah atribut. Nilai IG diperoleh dari nilai *entropy* sebelum pemisahan dikurang dengan nilai *entropy* setelah pemisahan [26]. Nilai ini digunakan untuk penentuan atribut mana yang akan dibuang atau digunakan. Atribut yang memenuhi kriteria pembobotan nantinya akan digunakan untuk proses klasifikasi.

Dalam pemilihan fitur dengan IG dilakukan dengan 3 tahapan, yaitu:

1. Menghitung nilai IG untuk setiap atribut.
2. Menentukan *threshold* (batas). Hal ini digunakan untuk menentukan atribut yang bobotnya lebih kecil dari *threshold* akan dibuang.
3. Memperbaiki dataset dengan pengurangan atribut.

Rumus untuk mendapatkan perhitungan dari *entropy* dapat dilihat pada persamaan (2.8)

$$Entropy(S) = \sum_{i=1}^{m} p_i log_2(pi) \quad (2.8)$$

dimana:

m = jumlah kelas klasifikasi

p_i = jumlah proporsi sampel (peluang) untuk kelas i

Sedangkan rumus untuk *Information Gain* dari suatu atribut A, ditunjukkan pada persamaan (2.9)

$$Gain(S,A) = Entropy(S) - \sum_v \frac{|S_v|}{|S|} Entropy(S_v)$$

(2.9)

dimana:

A = variabel

v = nilai yang mungkin untuk variabel A

$|S_v|$ = jumlah sampel untuk nilai v

$|S|$ = jumlah sampel untuk nilai v

2.9. Klasifikasi

Klasifikasi merupakan cara pengelompokkan data sesuai dengan ciri-ciri atau karakteristik data ke kelas yang sesuai [15]. Vektor fitur pelatihan tersedia dan telah diketahui kelaskelasnya, kemudian vektor fitur pelatihan tersebut dimanfaatkan untuk merancang pemilah. Pengenalan pola ini disebut terbimbing, *supervised*. klasifikasi memiliki beberapa algoritma, diantaranya *Naïve Bayes*, *Support Vector Machine*, *Decission Tree*, *Fuzzy* dan Jaringan Saraf Tiruan.

2.10. *Support Vector Machine*

Support Vector Machine adalah metode klasifikasi untuk mencari *Maximum Marginal Hyperplane* (MMH) atau pemisah terbaik untuk menciptakan pemisahan maksimal untuk semua kelas [10]. Margin bisa didefinisikan 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* [18]. Berikut merupakan konsep SVM yang dimulai dSengan persamaan garis lurus. Dengan rumus persamaan (2.10) berikut:

$$y = ax + b \tag{2.10}$$

Persamaan garis lurus tersebut dituliskan pada persamaan (2.11) berikut:

$$y - ax - b = 0$$

(2.11)

Dimana:

a = kemiringan atau gradien (m)

b = bilangan konstanta, a dan b merupakan bilangan real dan a tidak nol.

Dalam SVM, secara general sebuah *hyperplane* dinyatakan pada persamaan (2.12) berikut:

$$w \cdot x + b = 0 \tag{2.12}$$

Keterangan:

w = nilai dari bidang normal

x = data input

b = posisi bidang relatif terhadap pusat koordinat, dimana skalar b bisa bernilai negatif, nol, maupun positif.

Pencarian *hyperplane optimum* yang memaksimasi margin dapat dipandang sebagai sebuah masalah *Quadratic Programming* (QP), yaitu mencari titik minimal dari persamaan (2.13) dengan memperhatikan batasan pada persamaan (2.14) berikut:

$$min\ \tau(w) = \frac{1}{2}||w||^2 \tag{2.13}$$

$$yi\left(x_i \cdot w + b\right) - 1 \geq 0,\ \forall i$$

(2.14)

Masalah Lagragian dijelaskan pada persamaan (2.15) untuk pengklasifikasian

$$L(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i \left(y_i \left(x_i \cdot w + b\right) - 1\right), i = 1,2, \ldots, l$$

(2.15)

Kemudian untuk memaksimalkan L terhadap α_i dijelaskan pada persamaan (2.16) dengan memperhatikan batasan pada persamaan (2.17).

$$\max_{\alpha} L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.16)$$

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

(2.17)

Setelah mendapatkan *Support Vector* dengan α_i bernilai positif, maka selanjutnya menghitung w yang dapat dilihat pada persamaan (2.18) dan menghitung b yang dapat dilihat pada persamaan (2.19).

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

(2.18)

$$b = y_k - w^T x_k$$

(2.19)

Berikutnya untuk menghitung kelas $\bar{x}$, dapat dilihat pada persamaan (2.20)

$$F(\bar{x}) = W^T x + b \quad (2.20)$$

2.10.1. *Kernel Trick*

Dengan menggunakan *kernel trick*, hanya perlu diketahui fungsi kernel yang dipakai untuk menentukan *support vector*. Sehingga tidak perlu mengetahui wujud dari fungsi nonlinier ϕ. Pada umumnya terdapat empat jenis fungsi kernel yang dapat digunakan, yaitu:

1. Kernel linier, dapat dilihat pada persamaan (2.21).

$$K\left(x, x_k\right) = X_k^T x \tag{2.21}$$

2. Kernel polynomial, dapat dilihat pada persamaan (2.22).

$$K\left(x, x_k\right) = X_k^T x + 1)^d \tag{2.22}$$

3. Kernel Gaussian (radial basis function, RBF), dapat dilihat pada persamaan (2.23).

$$K\left(x, x_k\right) = exp \quad \left\{-|\left|x - x_k\right||_2^2 / \sigma^2\right\} \tag{2.23}$$

4. Kernel sigmoid, dapat dilihat pada persamaan (2.24).

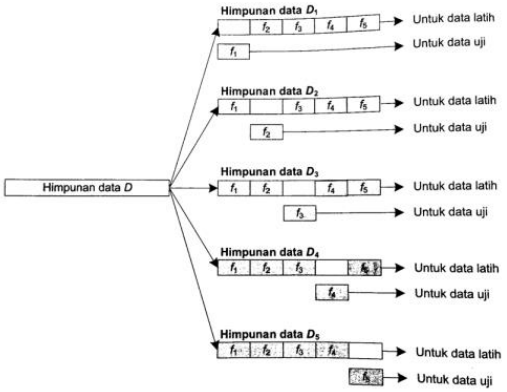
$$K\left(x, x_k\right) = \tanh[k x_k^T x + \theta] \tag{2.24}$$

2.11. Metode Pengujian

Dalam penelitian ini metode pengujian yang digunakan yaitu *K-Fold Cross Validation* dan *Confusion Matrix*.

2.11.1. *K-Fold Cross Validation*

K-fold Cross Validation adalah sebuah metode proses validasi untuk memperkirakan kinerja dari model pembelajaran mesin atau *machine learning*. Secara umum, *K-fold Cross Validation* digunakan untuk memperkirakan akurasi karena biasnya yang relatif rendah [19]. *K-fold Cross Validation* digunakan juga sebagai penghilang *noise* kata untuk meningkatkan akurasi dengan cara mengerjakan perulangan dengan masukan atribut acak. Untuk ilustrasi *K-Fold Cross Validation* ditunjukkan pada Gambar 2.1



Gambar 2. 1 Ilustrasi *K-Fold Cross Validation*

Diawali dengan pembagian data sejumlah *n-fold*. *Fold* ke-1 berarti data uji berada pada bagian ke-1, dan sisanya berupa data latih, *Fold* ke-2 berarti data uji berada pada bagian ke-2, dan sisanya berupa data latih, begitu seterusnya hingga *fold* ke-n. Berikut gambar dari metode *K-fold Cross Validation* dengan k = 10

2.11.2. *Confusion Matrix*

Confusion Matrix merupakan sebuah tabel yang berisi klasifikasi jumlah testing data yang benar atau sesuai dan jumlah testing data yang salah [28]. Metode ini digunakan untuk mengetahui kinerja dari sebuah sistem yang akan dievaluasi. *Confusion Matrix* berguna dalam menganalisis kualitas suatu *classifier* mengenali tuple. Dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Confusion Matrix

Confusion Matrix		**Kelas Prediksi**		
		Positif	**Negatif**	**Netral**
Kelas Sebenarnya	**Positif**	TPP	PFNeg	PFNet
	Negatif	NegFP	TNegNeg	NegFNet
	Netral	NetFP	NetFNeg	TNetNet

Keterangan:

1. **TPP (*True Possitive Possitive*)**, merupakan jumlah dokumen dari kelas positif yang benar diklasifikasikan sebagai kelas positif.

2. **TNegNeg (*True Negative Negative*)**, merupakan jumlah dokumen dari kelas negatif yang benar diklasifikasikan sebagai kelas negatif.
3. **TNetNet (*True Netral Netral*)**, merupakan jumlah dari kelas netral yang benar diklasifikasikan sebagai kelas netral.
4. **PFNeg (*Positive False Negatif*)**, merupakan jumlah dokumen dari kelas positif yang salah diklasifikasikan sebagai kelas negatif.
5. **NegFP (*Negatif False Positive*)**, merupakan jumlah dokumen dari kelas negatif yang salah diklasifikasikan sebagai kelas positif.
6. **PFNet (*Positive False Netral*)**, merupakan jumlah dokumen dari kelas positif yang salah diklasifikasikan sebagai kelas netral.
7. **NetFP (*Netral False Positive*)**, merupakan jumlah dokumen dari kelas netral yang salah diklasifikasikan sebagai kelas positif.
8. **NetFNeg (*Netral False Negatif*)**, merupakan jumlah dokumen dari kelas netral yang salah diklasifikasikan sebagai kelas negatif.
9. **NegFNet (*Negatif False Netral*)**, merupakan jumlah dokumen dari kelas negatif yang salah diklasifikasikan sebagai kelas netral.

Confusion Matrix menunjukkan tingkat nilai akurasi dari suatu proses klasifikasi yang telah dilakukan. Diantaranya terdiri dari *Accuracy*, *Precision*, *Recall*, dan *F*1 *Score*. Tingkat akurasi yaitu menunjukkan proporsi jumlah prediksi benar. Dapat dilihat pada persamaan (2.24).

$$Accuracy = \frac{\text{TPP} + \text{TNegNeg} + \text{TNetNet}}{\text{Total}} \tag{2.24}$$

Precision merupakan proporsi dari pelabelan yang teridentifikasi dengan benar. Dapat dilihat pada persamaan (2.25).

$$Precision = \frac{\text{TPP}}{\text{TPP} + \text{NegFP} + \text{NetFP}} \tag{2.25}$$

Recall merupakan proporsi dari informasi yang dapat ditemukan dari label. Dapat dilihat pada persamaan (2.26).

$$Recall = \frac{\text{TPP}}{\text{TPP + PFNeg +PFNet}} \quad (2.26)$$

Kemudian dari nilai *Precision* dan *Recall* dapat digunakan untuk mendapatkan proporsi pengukuran lain yaitu *F*1-*Score*. *F*1-*Score* merupakan harmonic mean dari perhitungan Precision dan Recall. Dapat dilihat pada persamaan (2.27).

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.27)$$

2.12. Penelitian-Penelitian Terkait

Tabel 2. 2. Penelitian-Penelitian Terkait

Review Literature [10]	
Judul Artikel	Analisis Sentimen Opini Pelanggan Terhadap Aspek Pariwisata Pantai Malang Selatan Menggunakan TF-IDF dan Support Vector Machine
Penulis	Yoga Tika Pratama, Fitra Abdurrachman Bachtiar, Nanang Yudi Setiawan
Nama Jurnal/Proceeding	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Tahun Penerbitan	2018
Tujuan atau Masalah Utama Yang Diangkat	Untuk mendapatkan perspektif pelanggan terhadap aspek pariwisata pantai Malang Selatan bisa dengan melakukan Analisis Sentimen pada tingkat aspek. Salah satu metode klasifikasi yaitu Support Vector Machine yang dapat digunakan untuk melakukan klasifikasi sentimen dalam proses Analisis Sentimen tersebut.
Metode	*Support Vector Machine*
Hasil Penelitian Kesimpulan	a. Hasil Penelitian: rata-rata Accuracy sebesar 85%, Precision sebesar 85%, Recall sebesar 87%, dan F1-Score sebesar 85%. b. Kesimpulan: Klasifikasi sentimen menggunakan Support Vector Machine mampu

	mengklasifikasikan sentimen untuk aspek Umum dengan tingkat akurasi sebesar 85%. Klasifikasi sentimen untuk aspek Kebersihan dengan tingkat akurasi 87%. Klasifikasi sentimen untuk aspek Keramaian dengan tingkat akurasi sebesar 92%. Klasifikasi sentimen untuk aspek Akses Jalan dengan tingkat akurasi sebesar 85%. Klasifikasi sentimen untuk aspek Ombak dengan tingkat akurasi sebesar 87%
Saran Penelitian	1. Menambahkan sumber data opini pelanggan dari sumber data lain seperti Media Sosial, Google Reviews, dan platform lain sejenisnya yang memuat data opini pelanggan pariwisata Pantai Malang Selatan. 2. Penggunaan metode untuk memperbaiki penggunaan kata yang tidak baku dan translasi frasa bahasa asing dalam tahap Text Preprocessing. Hal ini bertujuan untuk mengoptimalkan jumlah feature yang akan digunakan untuk tahap klasifikasi menggunakan algoritme klasifikasi. 3. Pemilihan aspek yang akan dianalisis dalam proses Analisis Sentimen dapat menggunakan pendekatan ekstraksi aspek dengan metode Part-Of-Speech Tagger dengan tujuan mendapatkan aspek yang memang benar-benar ada dalam dataset.
Review Literature [11]	
Judul Artikel	Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking
Penulis	Shima Fanissa, M. Ali Fauzi, Sigit Adinugroho
Nama Jurnal/Proceeding	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Tahun Penerbitan	2018
Tujuan atau Masalah Utama Yang Diangkat	Menganalisis ulasan dari masyarakat tentang pariwisata Kota Malang melalui analisis sentimen dan diklasifikasikan menjadi dua kelas yaitu positif dan negatif.
Metode	Naive Bayes dengan seleksi fitur Query Expansion Ranking

Hasil Penelitian Kesimpulan	3.1. Hasil Penelitian: Pengujian pada penelitian ini adalah uji akurasi dengan menggunakan variasi rasio seleksi fitur, hasilnya seleksi fitur 75% memiliki akurasi terbaik sebesar 86.6%. 3.2. Kesimpulan: Metode Multinomial Naive Bayes dapat diterapkan pada proses analisis sentimen pariwisata Malang
Saran Penelitian	Dalam penyempuranaan penelitian ini maka penelitian selanjutnya disarankan untuk memperhatikan singkatan, gabungan dua kata atau lebih (bigram, trigram, n-gram), kata-kata ambigu, dan kalimat sarkastik supaya hasil klasifikasinya lebih sempurna.
Review Literature [12]	
Judul Artikel	Klasifikasi Hate Speech Berbahasa Indonesia di Twitter Menggunakan Naive Bayes dan Seleksi Fitur Information Gain dengan Normalisasi Kata
Penulis	Ivan, Yuita Arum Sari, Putra Pandu Adikara
Nama Jurnal/Proceeding	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Tahun Penerbitan	2019
Tujuan atau Masalah Utama Yang Diangkat	Membuat sistem yang bisa mengklasifikasikan sebuah tweet pada Twitter ke dalam kelas hate speech (HS) ataupun kelas non hate speech (NONHS)
Metode	Naïve Bayes dan seleksi fitur Information Gain dengan normalisasi kata
Hasil Penelitian Kesimpulan	**3.1.1.** Hasil Penelitian: Hasil akurasi terbaik diperoleh dengan menggunakan normalisasi kata pada tahap pre-processing dan menggunakan seleksi fitur Information Gain dengan threshold 80%. Hasil akurasi terbaik adalah sebesar 98%, nilai precision sebesar 100%, nilai recall sebesar 96,15%, dan nilai f-measure sebesar 98,03%. **3.1.2.** Kesimpulan: Pada saat melakukan klasifikasi hate speech berbahasa Indonesia di Twitter menggunakan Naïve Bayes dan seleksi fitur Information Gain dengan normalisasi kata mampu meningkatkan hasil akurasi menjadi lebih baik
Saran Penelitian	Pada penelitian selanjutnya juga bisa memakai metode lain seperti neural network, Support Vector Machine

	(SVM), dan beberapa metode lainnya untuk membandingkan dengan metode Naïve Bayes
Review Literature [13]	
Judul Artikel	Klasifikasi Ulasan Pengguna Aplikasi Mandiri Online di Google Play Store dengan Menggunakan Metode Information Gain dan Naive Bayes Classifier
Penulis	Amalia Elma Sari1 , Sri Widowati2 , Kemas Muslim Lhaksmana
Nama Jurnal/Proceeding	E-Proceeding of Engineering
Tahun Penerbitan	2019
Tujuan atau Masalah Utama Yang Diangkat	Membangun sebuah sistem yang dapat melakukan klasifikasi ulasan pengguna di Google Play Store termasuk kedalam ulasan positif atau negatif, serta mengklasifikasikan berdasarkan faktor kualitas perangkat lunak ISO/IEC 25010
Metode	Naive Bayes dikombinasikan dengan metode seleksi fitur yaitu Information Gain
Hasil Penelitian dan Kesimpulan	**3.2.1.** Hasil Penelitian: Akurasi dan f-measure yang didapat pada klasifikasi dengan seleksi fitur Information Gain yaitu 91,33% dan 89,18%. **3.2.2.** Kesimpulan: Klasifikasi sentimen menggunakan Naive Bayes Classifier dengan feature selection Information Gain akan menghasilkan akurasi dan f-measure lebih baik daripada tanpa feature selection Information Gain.
Saran Penelitian	Melakukan perhitungan feature selection information gain berdasarkan kelas kategori, karena untuk melihat pengaruh dari information gain terhadap klasifikasi kategori dapat dilakukan berdasarkan nilai ketergantungan antara fitur dengan kelas kategori tersebut.
Review Literature [14]	
Judul Artikel	Analisis Sentimen Masyarakat Terhadap Mass Rapid Transit Jakarta Menggunakan Metode Naïve Bayes Dengan Normalisasi Kata
Penulis	Tania Malik Iryana, Indriati, Putra Pandu Adikara

Nama Jurnal/Proceeding	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Tahun Penerbitan	2021
Tujuan atau Masalah Utama Yang Diangkat	Dalam penulisan opini di media sosial sering ditemukan kesalahan seperti kata tidak baku, singkatan, dan salah ketik yang dapat mempersulit proses klasifikasi
Metode	Menggunakan normalisasi kata dengan kamus slang words dan singkatan serta normalisasi kata dengan Peter Norvig dan metode klasifikasi Naïve Bayes.
Hasil Penelitian dan Kesimpulan	**3.2.3.** Hasil Penelitian: Hasil evaluasi rata-rata 5-fold dari klasifikasi Naïve Bayes dengan normalisasi kata menggunakan kamus slang words dan singkatan serta normalisasi kata menggunakan Peter Norvig menghasilkan 0,903 untuk precision, 0,944 untuk recall, 0,922 untuk f-measure, dan 0,903 untuk accuracy. **3.2.4.** Kesimpulan: Penggabungan dari kedua normalisasi kata (kamus SS dan Peter Norvig) mendapatkan hasil evaluasi yang lebih baik dibandingkan hanya memakai salah satu normalisasi kata.
Saran Penelitian	Menambahkan metode normalisasi kata yang bisa memperbaiki lebih dari 2 jarak huruf perubahan untuk mendapatkan hasil evaluasi yang lebih baik.
Review Literature [8]	
Judul Artikel	Penggunaan Spelling Correction dengan Metode Peter Norvig dan N-Gram
Penulis	Ricky Martin, Dali Santun Naga, Viny Christanti
Nama Jurnal/Proceeding	Jurnal Ilmu Komputer dan Sistem Informas
Tahun Penerbitan	2021
Tujuan atau Masalah Utama Yang Diangkat	Membuat aplikasi Spelling Correction berbasis web untuk mengoreksi dan mengubah kata-kata di dalam dokumen ketika terdapat ejaan yang salah dalam Bahasa Indonesia
Metode	Peter Norvig dan N-Gram

Hasil Penelitian dan Kesimpulan	1. Hasil Penelitian: Nilai akurasi hasil Spelling Correction sebesar 69.09% dengan menggunakan 55 dokumen sebagai data pengujiannya. 2. Kesimpulan: Peter Norvig dan N-Gram dapat mengoreksi dokumen berupa text dan aplikasi sudah berhasil mengubah kata-kata yang mengandung sala
Saran Penelitian	1. Aplikasi sangat bergantung pada kualitas kamus daftar kalimat, sehingga kamus kalimat harus mencangkup banyak kalimat- kalimat umum. 2. Menambahkan beberapa fitur untuk mengoreksi Word dan PDF. 3. Mengoreksi kata yang memperhatikan frasa-frasa pada kalimat