BAB 2
LANDASAN TEORI

2.1. Analisis Sentimen

Analisis sentimen adalah studi komputasi pendapat, sentimen, emosi, dan sikap orang, masalah menarik ini semakin penting dalam bisnis dan masyarakat. Tidak seperti informasi faktual, sentimen dan opini memiliki karakteristik penting, yaitu subjektif. Subjektivitas berasal dari banyak sumber. Sumber subjektivitas yaitu setiap orang memiliki pengalaman, pandangan dan minat yang berbeda sehingga opini tiap-tiap orang memiliki pendapat yang berbeda dengan yang lainnya. Karena setiap orang memiliki pendapat yang berbeda, sehingga penting untuk memeriksa kumpulan pendapat dari banyak orang daripada hanya satu pendapat dari satu orang, karena mendengar pendapat dari satu orang hanya mewakili pandangan subjektif dari satu orang itu, sehingga tidak cukup untuk membantu mengambil tindakan. Dengan banyaknya pendapat yang diambil atau digunakan dapat dibentuk kedalam sebuah ringkasan untuk sebuah bisnis mengambil tindakan seperti pengembangan produk. Secara umum analisis sentimen dibagi menjadi tiga level[2], yaitu:

1. Level dokumen (*Document Level*)

   Pada level dokumen digunakan untuk mengklasifikasikan keseluruhan suatu dokumen kedalam sentimen positif atau negatif.
2. Level kalimat (*Sentence Level*)

   Pada level kalimat digunakan untuk mengklasifikasikan sebuah kalimat ke dalam sentimen positif atau negatif.
3. Level aspek dan entitas (*Aspect and Entity Level*)

   Pada level aspek dan entitas digunakan untuk mengklasifikasikan sebuah kalimat kedalam sebuah aspek dan diberikan sebuah sentimen positif atau negatif.

2.1.1. Analisis Sentimen Berbasis Aspek

Dari segi kelengkapan analisis sentiment level aspek bisa dibilang lebih lengkap bila dibandingkan dengan level dokumen atau pun level kalimat. Hal ini dikarenakan analisis sentimen level kalimat dan level dokumen hanya difokuskan pada penentuan sentimen positif dan negatif saja, tanpa aspek yang terkandung di dalamnya. Seperti kalimat berikut "makanannya mahal tetapi enak dan pelayanannya bagus" dimana pada kalimat tersebut memiliki sentimen yang positif secara keseluruhan kalimat tetapi dalam kalimat tersebut terdapat aspek yang memiliki sentimen yang negatif, sehingga diperlukan analisis yang lengkap untuk mengetahui sentimen positif atau negatif pada masing-masing aspek dengan melakukan analisis sentimen berbasis aspek[2].

Dalam melakukan analisis sentimen berbasis aspek, diperlukan beberapa tugas atau tahap dasar. Di antara tahap dasar yang diperlukan terdapat dua tugas yang paling banyak mendapat perhatian penelitian[2], yaitu:

1. *Aspect extraction*

   Pada tahap *Aspect extraction* untuk mengekstrak aspek yang telah ditentukan sebelumnya.
2. Aspect sentiment classification.

   Pada tahap *Aspect sentiment classification* untuk menentukan sebuah pendapat dari tiap aspek kedalam sebuah sentimen positif atau negatif.

2.2. Multilabel Classification

Multilabel classification adalah sebuah masalah klasifikasi dimana setiap sampel data dapat diklasifikasikan ke dalam lebih dari satu kelas pada saat yang sama. Sebagai contoh, dalam klasifikasi genre film, satu film dapat termasuk dalam beberapa genre seperti komedi, romantis, dan drama sekaligus. Dimana terdapat beberapa cara untuk menyelesaikan masalah *multilabel classification* yaitu *binary relevance*, *classifier chains*, *label powerset* dengan pendekatan yang bisa digunakan yaitu *one-verus-one* dan *one-versus-rest*[7].

2.2.1. *One-versus-rest*

One-versus-Rest (OvR) digunakan untuk menyelesaikan masalah multilabel classification. Pendekatan ini juga dikenal sebagai *One-versus-All* (OvA). Pada pendekatan OvR, setiap kelas atau label dianggap sebagai label positif, dan label-label lainnya dianggap sebagai label negatif. Sebuah model klasifikasi biner dipelajari untuk setiap label positif dengan membedakan sampel yang termasuk dalam kelas positif tersebut dari sampel yang termasuk dalam kelas negatif[7].

2.2.2. Binary Relevance

Binary Relevance merupakan salah satu cara dalam klasifikasi multilabel. Cara kerja dari metode *binary relevance*, yaitu memisahkan setiap label untuk per kelas atau per aspek, lalu pada masing-masing aspek tersebut akan dijadikan sebuah model menggunakan metode yang akan digunakan, setelah proses training dilakukan akan dilakukan pengujian dengan data baru untuk masing-masing aspek menggunakan metode yang akan digunakan, lalu akan didapatkan hasil prediksi [7].

2.3. Preprocessing

Tahap preprocessing dilakukan sebelum dilakukannya proses klasifikasi, tujuan dari tahap Preprocessing ini dilakukan untuk membersihkan data dari kata, simbol dan huruf yang tidak diperlukan agar menyeragamkan bentuk kata sehingga dapat mengurangi volume kata. Karena sebelum dilakukan tahap Preprocessing dataset memungkinkan memiliki noise yang tinggi yang dapat mempengaruhi proses klasifikasi[3][4].

2.2.1. *Case Folding*

Case Folding merupakan tahapan praproses dimana setiap kata pada dataset diubah atau diseragamkan menjadi huruf kecil semua[4].

2.2.2. *Cleaning*

Cleaning merupakan tahapan pada praproses dimana menghapus tanda baca, angka, dan simbol-simbol pada dataset[4].

2.2.3. *Normalization*

Normalization atau normalisasi merupakan proses perbaikan kata-kata yang salah eja atau disingkat dalam bentuk tertentu dan juga mengubah kata tidak baku menjadi kata baku berdasarkan kamus[4].

2.2.4. *Stopword Removal*

Stopword Removal merupakan sebuah proses menghapus kata-kata yang tidak penting. Proses *Stopword Removal* dapat dilakukan dengan dua cara yaitu *wordlist* atau *stoplist*. Proses Stoplist berisi sekumpulan kata yang tidak memiliki makna atau biasa disebut dengan stopword[4].

2.2.5. *Tokenization*

Tokenization atau tokenisasi adalah tahap memisahkan atau memotong sebuah kalimat menjadi kata per kata dengan sebuah pemisah yaitu *blank space* atau spasi[4].

2.2.6. *Stemming*

Stemming, merupakan tahapan praproses yang bertujuan untuk menghapus imbuhan yang terdapat pada kata atau mengubah kata menjadi bentuk dasarnya[4].

2.2.7. *Convert Negation*

Convert Negation adalah merupakan tahapan menggabungkan kata negasi dengan kata selanjutnya, karena kata negasi dapat mengubah makna sebuah sentimen dalam sebuah ulasan[8].

2.4. Pembobotan TF-IDF

Metode TF-IDF digunakan untuk memberikan bobot terhadap suatu kata berdasarkan tingkat kepentingan terhadap dokumen atau kategori dalam suatu kumpulan dokumen, penentuan nilai bobot dilakukan dengan cara menghitung frekuensi kemunculan kata dalam dokumen[9]. Perhitungan TF (*Term Frequency*) menunjukan jumlah munculnya suatu kata pada suatu dokumen/teks, sedangkan DF (Docum*ent Frequency*) merupakan jumlah dari dokumen/teks yang terdapat suatu kata/term[10].

Persamaan TF[11] dapat dirumuskan dengan persamaan 2.1. sebagai berikut:

$$TF_{ij} = \frac{f_i(d_j)}{\sum_{i=1}^{k} f_i(d_j)} \quad (2.1)$$

$\mathrm{TF}(i,j)$: *Term Frequency i* pada dokumen j

$f_i(d_j)$: Frekuensi kemunculan term i pada dokumen j

$\sum_{i=1}^{k} f_i(d_j)$: Total term pada dokumen j

Persamaan IDF[12] dapat dirumuskan dengan persamaan 2.2. sebagai berikut:

$$IDF = log\left(\frac{1+D}{1+DF_j}\right) + 1 \quad (2.2)$$

Dimana :

IDF : *Inverse Document Frequency*

D : Jumlah Dokumen

DF_j : jumlah dokumen yang berisi term j.

Persamaan TF-IDF[12] dapat dirumuskan dengan persamaan 2.3 berikut.

$$\mathrm{W}(i,j) = tf_{(i,j)} * IDF \quad (2.3)$$

Dimana:

$\mathrm{W}(i,j)$: bobot kata j terhadap dokumen i

$tf_{(i,j)}$: banyaknya kemunculan kata j pada dokumen i

IDF : *Inverse Document Frequency*

Pada penelitian ini menggunakan *library* Python yaitu Sklearn dimana TF-IDF telah dinormalisasi, adapaun TF-IDF yang dinormalisasi[12] dirumuskan dengan persamaan 2.4. sebagai berikut:

$$W(i,j) = \frac{tf_{(i,j)} * IDF}{\sqrt{\sum_{s=1}^{k}(tf_{(i,j)} * IDF)^2}} \quad (2.4)$$

2.5. *Neighbor Weighted K-Nearest Neighbor*

Metode *Neighbor Weighted K-Nearest Neighbor*(NWKNN) merupakan pengembangan dari algoritma *K-Nearest Neighbor*(KNN), dimana metode NWKNN memiliki kelebihan dibandingan metode KNN yaitu dapat menangani data yang tidak seimbang[1], dengan penambahan beberapa tahapan yaitu pada tahap penghitungan skor dimana pada metode NWKNN dilakukan penghitungan bobot terlebih dahulu serta perbedaan dalam penghitungan skor dimana dikalikan dengan penghitungan bobot.

Perhitungan bobot[1] dapat dihitung dengan menggunakan persamaan 2.5. sebagai berikut:

$$Weighti = \frac{1}{\left(\frac{num(C_i^d)}{min\{Num(c_j^d)|j = 1,..,k *\}}\right)^{1/E}} \quad (2.5)$$

Dimana:

$num(C_i^d)$: banyaknya data latih d pada kelas i

$num(C_j^d)$: banyaknya data latih d pada kelas j, dimana j terdapat dalam himpunan k tetangga terdekat

E : bilangan lebih dari 1

I : Kelas Polaritas

J : Kelas

Perhitungan nilai skor[13] dapat dihitung dengan menggunakan persamaan 2.6. sebagai berikut:

$$Score\,(q, C_i) = Weight_i * \left(\sum_{d_j KNN(X)} ((\sqrt{\sum_{i=1}^{n} (xi - yi)^2}) * \delta(d_j, C_i)) \right) \qquad (2.6)$$

Dimana:

$Weight_i$: Bobot kelas *i*

$\sqrt{\sum_{i=1}^{n} (xi - yi)^2}$: Jarak antara data latih dan data uji

$\delta(d_j, C_i)$: $\begin{cases} akan\ bernilai\ 1\ jika\ nilai\ jarak \in Ci \\ akan\ bernilai\ 0\ jika\ nilai\ jarak \notin Ci \end{cases}$

C_i : Kelas / Kategori I

2.6. Euclidean Distance

Euclidean Distance merupakan salah satu cara untuk menghitung jarak skalar jauh atau dekatnya suatu data dengan data yang lain. Euclidean Distance melakukan perhitungan jarak dari 2 (dua) buah titik yang berada dalam Euclidean space (baik bidang euclidean dua dimensi, tiga dimensi, dan seterusnya)[14]. Semakin kecil jarak euclidean antara dua buah data maka dapat disimpulkan semakin mirip pula kedua data tersebut[15].

Persamaan untuk menghitung jarak dengan Euclidean Distance[16] menggunakan persamaan (2.7) berikut:

$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \qquad (2.7)$$

Dimana:

d : Jarak antara x dan y

x : Data pusat klaster

y : Data pada atribut

i : Setiap data

n : Jumlah data

xi : Data pada pusat klaster ke i

yi : Data pada setiap data ke i

2.7. Confusion Matrix

Salah satu alat yang dapat mengukur performansi yaitu confusion matrix. Confusion matrix berbentuk seperti tabel yang mencakup matrix dua dimensi, dimana dimensi yang pertama diisi oleh kelas sebenarnya dari suatu objek dan di dimensi yang lain diisi oleh kelas yang dihasilkan oleh classifier atau hasil prediksi[17]. Confusion matrix menggunakan *multi-label*[18] dapat dilihat pada Tabel 2.1 berikut.

Tabel 2. 1. Confussion Matrix

Confusion Matrix		Prediksi		
		$C_{Positif}$	$C_{Negatif}$	$C_{NonSentimen}$
Aktual	$C_{Positif}$	$C_{P,P}$	FP	$C_{P,NS}$
	$C_{Negatif}$	FN	TP	FN
	$C_{NonSentimen}$	$C_{NS,P}$	FP	$C_{NS,NS}$

Precision adalah proporsi dari pelabelan yang teridentifikasi dengan benar[19], rumus untuk mencari *precision*[18] adalah:

$$Precision(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \quad (2.8)$$

Recall merupakan proporsi dari informasi yang dapat ditemukan dari label[19], rumus yang dapat digunakan untuk mencari *recall*[18] adalah:

$$Recall(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (2.9)$$

Precision dan *Recall* dapat digunakan untuk mendapatkan proporsi pengukuran lain yaitu *F1-Score*[19], rumus untuk mencari *F1-Score*[18] adalah:

$$F1\ Score(C_i) = 2 \times \frac{Precision(C_i) \times Recall(C_i)}{Precision(C_i) + Recall(C_i)} \quad (2.10)$$

2.8. Penelitian-Penelitian Terkait

Tabel 2. 2. Penelitian-Penelitian Terkait

Review Literature [4]	
Judul Artikel	Klasifikasi Berita Menggunakan Metode *K-Nearest Neighbor*
Penulis	Rahmah Miya Juwita, Elin Haerani, Siska Kurnia Gusti dan Siti Ramadhani
Judul Jurnal/Proceeding	Jurnal Nasional Komputasi dan Teknologi Informasi
Tahun Penerbitan	2022
Tujuan atau Masalah Utama Yang Diangkat	Dalam pengkategorian berita masih tergolong umum sehigga digunakan metode *K-Nearest Neighbor* dalam klasifikasi berita secara detail.
Metode	*K-Nearest Neighbor*
Hasil Penelitian Kesimpulan	a. Hasil Penelitian: Akurasi tertinggi didapatkan sebesar 87% dengan nilai K=3 dan pembagian data latih 80% dan data uji 20%. b. Kesimpulan: Metode *K-Nearest Neighbor* dapat digunakan untuk klasifikasi berita.
Saran Penelitian	-
Review Literature [5]	
Judul Artikel	Klasifikasi Data Tidak Seimbang menggunakan

	Algoritma SMOTE dan *K-Nearest Neighbor*
Penulis	Rimbun Siringoringo
Judul Jurnal/Proceeding	Jurnal ISD
Tahun Penerbitan	2018
Tujuan atau Masalah Utama Yang Diangkat	Metode *K-Nearest Neighbor* memiliki kekurangan dalam penggunaan data yang tidak seimbang
Metode	*K-Nearest Neighbor* dan SMOTE
Hasil Penelitian Kesimpulan	a. Hasil Penelitian: Tanpa penerapan algoritma SMOTE menghasilkan akurasi, F-Measure dan G-Mean masing-masing sebesar 78%, 43%, dan 28%. Sedangkan dengan menerapkan algoritma SMOTE menghasilkan akurasi,F-Measure dan G-Mean masing-masing sebesar 80%,80% dan 81% b. Kesimpulan: Dengan menerapkan algoritma SMOTE untuk menangani data yang tidak seimbang dapat meningkatkan akurasi, F-Measure dan G-Mean.
Saran Penelitian	-
Review Literature [6]	
Judul Artikel	Klasifikasi Penyimpangan Tumbuh Kembang pada Anak Menggunakan Metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN)
Penulis	Afrizal Rivaldi, Putra Pandu Adikara, dan Sigit Adinugroho
Judul Jurnal/Proceeding	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Tahun Penerbitan	2017
Tujuan atau Masalah Utama Yang Diangkat	Metode *K-Nearest Neighbor* memiliki kekurangan dalam penggunaan data yang tidak seimbang
Metode	*Neighbor Weighted K-Nearest Neighbor*
Hasil Penelitian Kesimpulan	a. Hasil Penelitian: Penggunaan metode NWKNN menghasilkan nilai akurasi tertinggi sebesar 95% sedangkan untuk menggunakan metode K-NN

	menghasilkan akurasi tertinggi sebesar 90% b. Kesimpulan: Metode NWKNN menghasilkan akurasi yang lebih tinggi dibandingkan metode K-NN.
Saran Penelitian	-
Review Literature [1]	
Judul Artikel	Sentiment Analysis for Review Mobile Applications Using Method *Neighbor Weighted K-Nearest Neighbor* (NWKNN)
Penulis	Indriati, Achmad Ridok
Judul Jurnal/Proceeding	Journal of Environmental Engineering & Sustainable Technology.
Tahun Penerbitan	2016
Tujuan atau Masalah Utama Yang Diangkat	Penggunaan data yang tidak seimbang menjadikan tingkat akurasi metode KNN menurun.
Metode	*Neighbor Weighted K-Nearest Neighbor* (NWKNN)
Hasil Penelitian dan Kesimpulan	a. Hasil Penelitian:Untuk data seimbang dengan data positif 100 dan data negatif 100 menghasilkan akurasi tertinggi 90% untuk kedua metode dengan nilai K=15, sedangkan untuk jumlah data tidak seimbang menggunakan metode K-NN mendapatkan akurasi 48% sedangkan untuk metode NWKNN mendapatkan akurasi 80% untuk K=45. b. Kesimpulan: Jika menggunakan data yang seimbang akurasi untuk kedua metode tidak jauh berbeda, sedangkan jika menggunakan data yang tidak seimbang akurasi NWKNN mendapatkan akurasi yang lebih tinggi.
Saran Penelitian	
Review Literature [3]	
Judul Artikel	Analisis sentimen pada Ulasan Produk Kecantikan Menggunakan K-Nearest Neighbor dan Information Gain
Penulis	Ekky Yulianti Prastika S, Said Al Faraby, Mahendra

	Dwifebri P
Judul Jurnal/Proceeding	*e-Proceeding of Engineering*
Tahun Penerbitan	2021
Tujuan atau Masalah Utama Yang Diangkat	Mencari nilai K yang optimal untuk metode KNN
Metode	K-Nearest Neighbor
Hasil Penelitian dan Kesimpulan	a. Hasil Penelitian: Nilai k optimal yang didapatkan yaitu 23 dengan mendapatkan akurasi rata-rata untuk semua aspek sebesar 74.21%. b. Kesimpulan: Dalam penelitian untuk mendapatkan hasil yang optimal dengan menggunakan k=23 dan untuk IG yaitu 0.5.
Saran Penelitian	
Review Literature [20]	
Judul Artikel	Klasifikasi Status Pembayaran Premi Menggunakan Algoritma Neighbor Weighted K-nearest Neighbor (Nwknn) (Studi Kasus: Pt. Bumiputera Kota Samarinda)
Penulis	Grassella, Ika Purnamasari, Fidia Deny Tisna Amijaya
Judul Jurnal/Proceeding	VARIANCE: Journal of Statistics and Its Applications
Tahun Penerbitan	2019
Tujuan atau Masalah Utama Yang Diangkat	Metode KNN tidak dapat menangani data yang tidak seimbang.
Metode	Neighbor Weighted K-Nearest Neighbor
Hasil Penelitian dan Kesimpulan	a. Hasil Penelitian: Hasil akurasi terbesar didapatkan dengan nilai K=3,4,5 dan nilai E=6 b. Kesimpulan: Untuk memperoleh hasil yang optimal maka nilai K yang digunakan 3 dan nilai E yang digunakan yaitu 6, didapatkan nilai akurasi sebesar 75%.
Saran Penelitian	