BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen atau disebut juga *opinion mining* adalah salah satu area penelitian yang populer dalam *natural language processing, data mining, information retrieval,* dan *web mining* yang digunakan untuk menganalisis atau mengidentifikasi opini, sentimen, dan emosi seseorang dalam menyikapi suatu topik tertentu. Analisis sentimen mengklasifikasi polaritas menjadi 3 yaitu positif, negatif, dan netral [1].

Analisis sentimen dapat dibedakan berdasarkan tugasnya, seperti *polarity classification* merupakan dasar penelitian analisis sentimen yang tugasnya mengklasifikasi polaritas pengguna, *the level of valence or arousal at specific scale* merupakan analisis sentimen yang membagi polaritas menjadi beberapa skala, dan *aspect-based sentimen analysis* yang merupakan analisis sentimen yang klasifikasi polaritas berdasarkan aspek atau fitur yang berbeda dalam suatu entitas. Secara umum, analisis sentimen dapat dilakukan pada tiga level, yaitu *document level* merupakan analisis tingkat dokumen yang mengklasifikasi seluruh dokumen dari subjektif atau objektif dan positif atau negatif, *sentence level* merupakan analisis tingkat kalimat yang lebih efektif dari pada analisis tingkat dokumen karena berfokus pada klasifikasi kalimat subjektif apakah menyatakan sentimen positif, negatif, atau netral, dan *wssord level* yang merupakan analisis tingkat kata dengan mengklasifikasi polaritas tiap kata yang terkait dengan subjektivitas kalimat atau dokumen [7].

2.1.1 Analisis Sentimen Berdasarkan Aspek

Analisis sentimen berdasarkan aspek merupakan analisis sentimen yang dilakukan berdasarkan istilah-istilah aspek yang diberi label dalam kalimat. Secara umum analisis sentimen berdasarkan aspek memanfaatkan hubungan atau posisi antar kata-kata target dan kata-kata konteks dengan menggunakan struktur pohon atau dengan menghitung jumlah kata [8].

Analisis sentimen berdasarkan aspek bertujuan untuk mengidentifikasi polaritas pada aspek tertentu dengan mengklasifikasi tiap kata yang memiliki kategori yang sama. Setiap kata pada suatu kalimat akan memiliki aspek dan polaritas sentimen yang berbeda-beda dan membuat proses analisis sentimen menjadi lebih akurat [2].

Analisis sentimen berdasarkan aspek memiliki dua tugas utama yaitu ekstraksi aspek dan klasifikasi aspek. Pada ekstraksi aspek, sentimen suatu data ditentukan berdasarkan aspek apa yang terdapat pada data tersebut. Sedangkan pada klasifikasi sentimen aspek, sentimen diklasifikasi polaritas menjadi positif, netral, dan negatif untuk suatu aspek [9].

2.2 Preprocessing

Preprocessing merupakan tahap persiapan data sebelum dianalisis. Preprocessing diperlukan untuk membersihkan data yang mengandung noise, mengurangi inkonsisten data, dan mengubah data yang sebelumnya tidak terstruktur menjadi data terstruktur sehingga data yang telah dibersihkan dapat lebih efektif digunakan dalam melakukan analisis sentimen [10]. Ada beberapa tahapan preprocessing pada penelitian ini yaitu case folding, filtering, tokenisasi, normalisasi dan stopword removal.

2.2.1 Case Folding

Case folding merupakan proses penyeragaman bentuk huruf dengan mengubah semua huruf pada dokumen menjadi huruf kecil (*lowercase)*. Proses penyeragaman bentuk huruf dilakukan untuk meningkatkan akurasi dalam membedakan kata-kata yang mirip [11].

2.2.2 Filtering

Filtering merupakan proses membersihkan tanda baca, angka, dan karakter yang tidak diperlukan. Proses filtering dilakukan untuk mengubah teks dari kalimat sebelum dipisah menjadi token yang berurutan [11].

2.2.3 Tokenisasi

Tokenisasi merupakan proses pemotongan setiap kata dalam sebuah dokumen. Proses pemotongan kata dilakukan berdasarkan spasi antar kata dalam

sebuah dokumen. Tokenisasi dilakukan agar kata-kata dapat dianalisis secara efektif [12].

2.2.4 Normalisasi

Normalisasi merupakan proses mengubah kata-kata yang tidak baku menjadi kata baku. Proses normalisasi dilakukan untuk mengatasi permasalahan banyaknya penyingkatan kata, penggunaan bahasa gaul atau slang, kelasahan ejaan pada suatu kalimat, dan penggunakan bahasan yang masih belum sesuai dengan kamus [13].

2.2.5 Stopword Removal

Stopword removal merupakan proses menghilangkan kata-kata yang tidak memberikan arti apa pun pada teks. Proses menghilangkan kata-kata dilakukan dengan cara membandingkan dengan stoplist. Stoplist berisi sekumpulan kata yang tidak relevan tetapi sering mucul dalam sebuah dokumen. Stoplist berisi kumpulan kata-kata stopword [12].

2.3 Inset Lexicon

Inset lexicon merupakan lexicon sentimen berbahasa Indonesia. Lexicon adalah sekumpulan kosakata lengkap dari suatu bahasa. Inset lexicon adalah metode berbasis lexicon yang berisi kata-kata sentimen dalam bahasa Indonesia dan memiliki bobot pada setiap kata. Inset lexicon dibangun untuk mengidentifikasi opini dan mengelompokkan menjadi opini positif atau negatif yang dapat digunakan untuk menganalisis sentimen. Inset lexicon menggunakan kumpulan data tweet berbahasa Indonesia dan dibuat dengan menimbang setiap kata secara manual dan disempurnakan dengan menambah kumpulan stemming dan sinonim. Inset lexicon terdapat 3.609 kata positif dan 6.609 kata negatif dengan bobot berkisar antara -5 hingga +5 [5].

Tabel 2.1 Sample data Inset Lexicon

Kata	Bobot
terawat	5
makasih	4
dinamika	3

mencederai	-4
nggak	-3

Langkah-langkah klasifikasi sentimen menggunakan *Inset Lexicon* adalah sebagai berikut [14]:

1. Pengecekan setiap kata dalam dokumen, jika kata tersebut ada dalam kamus *Inset Lexicon* maka kata tersebut diberi bobot.
2. Perhitungan *score* sentimen yaitu dengan menjumlahkan bobot sentimen setiap kata dalam satu dokumen.
3. *Score* sentimen yang sudah didapatkan akan diklasifikasi ke dalam sentimen positif, sentimen negatif, atau sentimen netral.

2.4 Koherensi

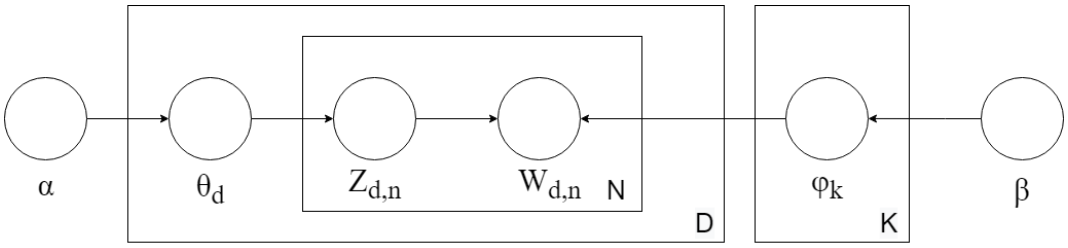
Koherensi merupakan sekumpulan pernyataan atau fakta yang saling mendukung. Koherensi dapat menilai topik terkait dengan pemahaman pada topik tersebut berdasarkan kata-kata pada topik sebagai fakta yang membatasi koherensi. Koherensi topik didasarkan pada vektor konteks untuk setiap kata teratas pada sebuah topik [15]. Koherensi dapat digunakan pada Latent Dirichlet Allocation untuk mengetahui nilai topik atau aspek terbaik. Pengujian koherensi juga dilakukan untuk mempermudah proses interpretasi jumlah aspek.

2.5 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) adalah algoritma *probabilistic topic modeling*. Algoritma LDA digunakan sebagai clustering *unsupervised learning* untuk data yang tidak terstruktur. LDA merepresentasikan topik berdasarkan probabilitas kata dari koleksi data seperti *corpus*. Kata-kata dengan probabilitas tertinggi di setiap topik akan memberikan gambaran yang baik tentang suatu topik [16].

LDA menghitung probabilitas kata-kata yang akan dimasukkan ke dalam setiap topik. LDA mengasumsikan bahwa beberapa kata dapat dikelompokkan

dalam topik yang berbeda dan menghitung probabilitas kata-kata yang akan dimasukkan ke dalam setiap topik berdasarkan probabilitas tertinggi yang sesuai dengan sebuah topik. Skema analsisi LDA dapat dilihat pada Gambar 2.1.



Gambar 2.1 Skema Model Latent Dirichlet Allocation (LDA)

Pada Gambar 2.1 α dan β mempresentasikan *dirichlet parameter* yang ditentukan oleh pengguna, θ_d merupakan distribusi probabilitas topik pada dokumen, φ_k merupakan distribusi probabilitas kata pada topik, $Z_{d,n}$ merupakan distribusi topik kata dalam dokumen, dan $W_{d,n}$ merupakan kata observasi [17]. Berdasarkan Gambar 3.2 Skema Model LDA dapat dirumuskan sebagai berikut:

$$P(Z_t = j|z_{-t}, w_t, d_t) = \frac{C_{w,j}^{WT} + \beta}{\sum_{w=1}^{W} C_{w,j}^{WT} + W.\beta} \; x \; \frac{C_{d,j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d,j}^{DT} + K.\alpha} \qquad \textbf{(2.1)}$$

Dengan keterangan sebagai berikut:

α = Distribusi topik per dokumen

β = Distribusi kata per topik

K = Jumlah topik

W = Jumlah token dalam dokumen lengkap

$C_{w,j}^{WT}$ = Jumlah kemunculan kata pada topik

$\sum_{w=1}^{W} C_{w,j}^{WT}$ = Jumlah kemunculan topik dalam matriks

$C_{d,j}^{DT}$ = Jumlah kemunculan topik dalam setiap dokumen

$\sum_{t=1}^{T} C_{d,j}^{DT}$ = Total jumlah berapa kali setiap dokumen muncul sebagai topik 0, topik 1, …, topik n

Berikut adalah tahapan ekstraksi topik menggunakan Latent Dirichlet Allocation [17]:

1. Menentukan nilai parameter α, β, dan *K* yang merupakan jumlah aspek. Nilai α dan β adalah rentang dari nilai $0 \leq \alpha, \beta \leq 1$. Kemudian, untuk setiap aspek $k = 1, 2, \ldots, K$ (*K* merupakan jumlah aspek).
2. Menghitung frekuensi setiap kata yang terdapat pada seluruh dokumen dan menentukan aspek secara acak pada setiap kata di setiap dokumen. Hasil sebaran kata untuk setiap aspek dicocokkan pada tiap dokumen, untuk dihitung berapa jumlah kata yang masuk di setiap aspeknya.
3. Menghitung jumlah kata yang masuk di setiap aspek pada masing-masing dokumen.
4. Menghitung peluang setiap aspek pada setiap kata, menggunakan persamaan (2.1).
5. Menentukan sebaran kata yang terdapat pada setiap topik berdasarkan nilai probabilitas tertinggi. Setiap aspek akan diambil 10 kata dengan probabilitas tertinggi. Nama label aspek akan ditentukan berdasarkan probabilitas kata tertinggi dari aspek.
6. Penentuan aspek pada setiap dokumen berdasarkan probabilitas tertinggi dari keseluruhan aspek untuk sebuah kata.

2.6 Confusion Matrix

Confusion matrix adalah metode yang digunakan untuk menghitung tingkat akurasi pada *machine learning*. *Confusion matrix* membandingkan hasil klasifikasi yang diprediksi oleh sistem dengan hasil klasifikasi yang seharusnya. Evaluasi yang dilakukan dengan menggunakan *confusion matrix* memiliki empat istilah untuk merepresentasikan hasil proses klasifikasi yang terdiri dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). *True Positive* (TP) merupakan jumlah kelas positif yang terdeteksi sebagai kelas positif, *False Positive* (FP) merupakan jumlah kelas negatif yang terdeteksi sebgai kelas positif, *True Negative* (TN) merupakan jumlah kelas negatif yang terdeteksi sebagai kelas

negatif, dan *False Negative* (FN) merupakan jumlah kelas positif yang terdeteksi sebagai kelas negatif [18].

Tabel 2.2 Pengkategorian *confusion matrix*

		Actual Class	
		+	-
Predicted Class	+	True Positive (TP)	False Posisitf (FP)
	-	False Negative (FN)	True Negative (TN)

Berdasarkan pengkategorian confusion matrix dapat diperoleh nilai akurasi, presisi, dan recall. Nilai akurasi merupakan persentasi dari seberapa akurat sistem dapat mengklasifikasi data secara benar dengan membandingkan data yang benar dengan keseluruhan data. Nilai presisi merupakan persentasi dari jumlah data positif dengan membandingkan jumlah data positif dengan keseluruhan data yang dikategorikan positif. Nilai recall merupakan persentasi dari data positif yang dikategorikan benar. Nilai f1-score merupakan perbandingan rata-rata nilai presisi dengan recall.

$$Akuarsi = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad \textbf{(2.2)}$$

$$Presisi = \frac{TP}{FP + TP} * 100\% \quad \textbf{(2.3)}$$

$$Recall = \frac{TP}{TP + FN} * 100\% \quad \textbf{(2.4)}$$

$$F1 - Score = 2\ x\ \frac{Presisi\ x\ Recall}{Presisi + Recall} \quad \textbf{(2.5)}$$

2.7 Python

Python merupakan bahasa pemograman komputer, sama halnya dengan bahasa pemograman lainnya, seperti bahasa pemograman C, C++, Java, PHP dan

lain-lain. Bahasa pemograman python memiliki varian dan aturan tersendiri yang jelas berbeda dengan bahasa pemograman lainnya. Selain itu python juga bahasa pemograman tingkat tinggi yang bersifat *interpreter, interactive*, *object-oriented* dan dapat beroperasi hampir disemua platform. Python termasuk bahasa pemograman yang mudah dipelajari karena memiliki code yang ringkas, mudah dibaca dan dapat dikombinasikan dengan banyak *library* yang siap pakai[19].