BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan atau beropini negatif atau positif [5].

Analisis sentimen disebut juga *opinion mining* adalah bidang ilmu yang menganalisa pendapat, sentimen, evaluasi, penilaian, sikap dan emosi publik terhadap entitas seperti produk, jasa, organisasi, individu, masalah, peristiwa, topik, dan atribut mereka [3].

2.2 Analisis Sentimen Berdasarkan Aspek

Analisis sentimen berdasarkan aspek merupakan perkembangan dari analisis sentimen yang mengacu pada sebuah kalimat. Analisis sentimen pada tingkat aspek digunakan untuk mengelompokan hal-hal yang dikeluhkan pelanggan ke dalam aspek-aspek tertentu dan divisualisasikan melalui *dashboard* sehingga informasi yang dibutuhkan oleh pihak manajemen dalam menentukan solusi yang tepat menjadi lebih rinci dan efektif [6].

2.3 Scraping

Scraping adalah proses pengambilan sebuah dokumen semi-terstruktur dari halaman web, untuk diambil data dari halaman tersebut. *Scraping* adalah teknik untuk mengekstraki data dari internet dan disimpan ke *file* atau *database* untuk kebutuhan analisis data. *Scraping* adalah metode untuk mengekstraksi informasi dari situs web, sehingga menjadi data yang dapat dianalisis dan dimanfaatkan untuk berbagai tujuan [7].

2.4 Valence Aware Dictionary and Sentiment Reasoner (VADER)

Valence Aware Dictionary and Sentiment Reasoner (VADER) merupakan alat analisis sentimen berbasis aturan yang secara khusus disesuaikan dengan sentimen yang diekspresikan di media sosial. VADER menggunakan kombinasi leksikon seperti kata-kata yang umumnya dilabeli menurut orientasi semantik sebagai kata positif, netral, dan negatif. Skor *compound* adalah metrik yang menghitung jumlah semua peringkat leksikon yang dinormalisasi antara -1 (sangat negatif) dan +1 (sangat positif) [4]. Rumus menghitung *compound* sebagai berikut:

$$Compound = \frac{x}{\sqrt{x^2 + \alpha}} \quad 2.4$$

Keterangan rumus:

x = Jumlah skor kata

α = Parameter normalisasi (nilai = 15)

Tabel 2.1 Skor *Compound*

No	**Sentimen**	**Skor *Compound***	**Keterangan**
1.	Positif	≥ 0.05	Bobot kata lebih banyak mengandung positif
2.	Netral	> -0.05 ~ < 0.05	Bobot kata mengandung netral atau tidak terdiri positif atau negatif
3.	Negatif	≤ -0.05	Bobot kata lebih banyak mengandung negatif

VADER menggunakan aturan-aturan tertentu untuk memasukan dampak dari setiap sub-teks pada intensitas yang dirasakan dari sentimen dalam teks kalimat, aturan tersebut di sub heuristik, aturan heuristik dalam VADER yaitu [8]:

1. Tanda baca, tanda seru (!), meningkatkan besarnya intensitas tanpa mengubah orientasi semantik.
2. Kapitalisasi, secara khusus menggunakan ALL-CAPS untuk menekankan kata yang relavan dengan sentimen di hadapan kata-kata yang tidak menggunakan huruf besar, meningkatkan intensitas sentiment tanpa mempengaruhi orientasi semantik.
3. Perubahan / modifikasi (juga disebut penguat, kata penguat) memengaruhi intensitas sentimen baik dengan meningkatkan atau mengurangi intensitas.
4. Pergeseran polaritas karena konjungsi, konjungsi kontras “tetapi” menandakan pergeseran polaritas sentimen, dengan sentimen teks setelah konjungsi menjadi dominan.

VADER mengambil sebuah string dan mengembalikan kamus skor di masing-masing dari 4 kategori positif, netral, negatif dan gabungan (*compound*).

2.5 Preprocessing

Tahap *pre-processing* adalah tahap dimana dilakukan seleksi data agar data yang akan digunakan menjadi lebih terstruktur [9]. *Preprocessing* menjadi tahap awal dalam klasifikasi teks untuk mempersiapkan data teks sebelum digunakan pada proses lainnya. Pada tahap ini akan mengubah data teks menjadi bentuk yang lebih baik sehingga menghasilkan infomasi teks dengan kualitas yang baik dan siap digunakan pada proses selanjutnya [10]. Dalam penelitian ini *text-preprocessing* terbagi menjadi *filtering*, *case folding*, *tokenizing*, *normalization*, *stopword* dan *stemming*.

2.5.1 Filtering

Filtering merupakan proses untuk menghilangkan tanda baca, angka, simbol, *link* URL, dan *username* yang terdapat pada teks [10].

2.5.2 Case Folding

Case Folding adalah proses mengkonversi keseluruhan teks kedalam format huruf kecil (lowercase) [9]. Hal ini bertujuan untuk memberikan bentuk standar pada teks.

2.5.3 Tokenizing

Tokenizing adalah proses pemotongan teks menjadi bagian-bagian yang lebih kecil, yang disebut token. Token adalah kata-kata yang dipisahkan oleh spasi dalam teks [9]. Proses ini dilakukan untuk mempermudah dalam pemberian bobot pada setiap katanya.

2.5.4 Normalization

Normalization adalah proses untuk mengubah kata yang tidak baku menjadi baku dan mengubah kata singkatan menjadi kata aslinya [10].

2.5.5 Stopword

Stopword removal adalah proses untuk menghilangkan kata-kata yang dianggap tidak penting di dalam teks [10]. Kata-kata yang termasuk ke dalam stoplist akan dibuang dan tidak digunakan pada proses selanjutnya.

2.5.6 Stemming

Stemming merupakan proses pencarian kata dasar dari sebuah kata yang telah berhasil melalui proses *stopword*. Pada tahap ini sebuah kata akan diubah menjadi kata dasar hingga kata yang memiliki imbuhan akan dihapus dan disesuaikan dengan kata dasarnya.

2.6 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 adalah ukuran yang digunakan untuk mengevaluasi model topik, metode yang secara otomatis menghasilkan topik dari kumpulan dokumen,

menggunakan model variabel laten. Setiap topik yang dihasilkan terdiri dari kata-kata, dan koherensi topik diterapkan ke *N* kata teratas dari topik tersebut. Ini didefinisikan sebagai rata-rata / median dari skor kemiripan kata berpasangan dari kata-kata dalam topik tersebut. Semakin sering kata-kata dalam topik tersebut muncul secara bersamaan maka nilai koherensi dari topik tersebut semakin tinggi [2]. Topik yang baik adalah topik yang dapat dideskripsikan dengan label pendek. Koherensi dapat digunakan pada Latent Dirichlet Allocation untuk mengetahui nilai topik atau aspek terbaik. Pengujian koherensi juga dapat dilakukan untuk memudahkan proses interpretasi jumlah aspek

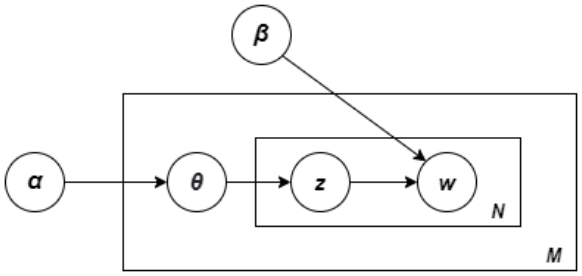
2.7 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) adalah teknik pemodelan topik yang secara otomatis menemukan topik dalam dokumen teks. LDA menganggap dokumen sebagai campuran dari berbagai topik dan setiap kata termasuk dalam salah satu topik dokumen. LDA membayangkan serangkaian topik yang tetap, setiap topik memiliki sekumpulan kata. Tujuan LDA adalah memetakan semua dokumen ke topik sedemikian rupa, sehingga kata-kata dalam setiap dokumen sebagian besar terkait dengan topik tersebut [11].

Latent Dirichlet Allocation (LDA) saat ini sering digunakan karena dapat melakukan klasterisasi, melakukan peringkasan, menghubungkan, dan dapat memproses data dengan memberikan bobot pada masing-masing dokumen yang nantinya menghasilkan daftar topik. Ide dasar dari Latent Dirichlet Allocation (LDA) ini menganggap bahwa dokumen yang kita ujikan dapat direpresentasikan sebagai sebuah model yang dicampur dari berbagai topik yang dibutuhkan, oleh sebab itu teknik ini disebut sebagai laten [12]. Secara formal, didefinisikan notasi sebagai berikut:

a. Kata adalah bentuk dasar dari data diskrit.
b. Sebuah dokumen adalah barisan kata-kata *N* yang dinotasikan dengan $\boldsymbol{w} = (w_1, w_2, \dots, w_n)$, dimana w_n adalah barisan kata ke-*n*.

c. Sebuah corpus adalah koleksi dari M dokumen dinotasikan dengan $D = (\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_n)$.



Gambar 2.1 Graphical Model LDA

Berdasarkan *graphical model* pada Gambar 2.1. Kotak-kotak tersebut adalah "pelat" yang mewakili pengulangan. Pelat luar mewakili dokumen, sedangkan pelat dalam mewakili pilihan perulangan topik dan kata-kata dalam dokumen [13]. Parameter α dan β diberikan untuk corpus. Parameter α adalah parameter untuk distribusi topik dari dokumen, dan parameter β adalah parameter untuk distribusi kata dari topik. Semakin besar nilai α, maka setiap dokumen mengandung sebagian besar topik, artinya tidak hanya ada satu topik spesifik. Sedangkan semakin besar nilai β, maka setiap topik mengandung sebagian besar kata, tidak hanya ada beberapa kata spesifik yang membedakan topik satu dengan lainnya. Untuk setiap dokumen N, terdapat distribusi topik yaitu θ. Karena LDA adalah soft clustering, maka setiap dokumen bisa terdiri dari beberapa topik yang berbeda. Kemudian menentukan topik dari setiap kata dalam setiap dokumen yang dinotasikan dengan z, untuk nantinya dikumpulkan menjadi cluster-cluster. Maka hasilnya adalah campuran kata-kata di tiap topik/cluster yang sudah ditentukan sebelumnya kemudian diinterpretasi hasil tiap cluster tersebut membahas topik apa [14]. Rumus perhitungan LDA sebagai berikut:

$$P(Z_t = j | z_{-t}, w_t, d_t) = \frac{C_{w.j}^{WT} + \beta}{\sum_{w=1}^{W} C_{w,j}^{WT} + W\beta} \times \frac{C_{d,j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d,t}^{DT} + T\alpha} \qquad 2.5$$

Keterangan dari rumus:

$P(Zt = j)$ = Probabilitas token pada topik

z_{-t} = Representasi topik dari semua token

w_t = Kata dari token

d_t = Dokumen yang berisi token

β = Distribusi kata per topik (parameter konsentrasi)

W = Panjang kosakata (jumlah token/kata unik dalam dokumen lengkap)

α = Distribusi topik per dokumen

T = Jumlah topik

$C_{w.j}^{WT}$ = Jumlah kemunculan kata/token pada topik

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

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

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

$\frac{C_{w.j}^{WT}+\beta}{\sum_{w=1}^{W} C_{w,j}^{WT}+W\beta}$ = Distribusi probabilitas kata pada suatu topik

$\frac{C_{d,j}^{DT}+\alpha}{\sum_{t=1}^{T} C_{d,t}^{DT}+T\alpha}$ = Distribusi probabilitas topik pada suatu dokumen

Untuk penentuan topik akhir dimana topik yang akan dipilih berdasarkan probabilitas tertinggi dari dua topik untuk sebuah kata.

2.8 Confusion Matrix

Confusion Matrix merupakan suatu instrumen yang digunakan untuk mengevaluasi performa dari model klasifikasi yang telah dihasilkan. Pada *Confusion Matrix*, hasil kelas prediksi akan dibandingkan dengan kelas data yang sebenarnya. Hasil tersebut kemudian akan digunakan untuk menghitung nilai *accuracy*, *precision*,

recall, dan *f-score*. Pengukuran evaluasi pada confusion matrix dapat dilihat pada tabel 2.2 berikut [15]:

Tabel 2.2 Confusion Matrix

Data Aktual	**Data Prediksi**		
	TRUE	**FALSE**	**TOTAL**
TRUE	TP	FN	P
FALSE	FP	TN	N
TOTAL	P’	N’	P+N

Keterangan:

TP (True Positive) = Data positif yang terklasifikasi secara benar.

TN (True Negative) = Data negative yang terklasifikasi secara benar.

FP (False Positive) = Data negatif yang terklasifikasi menjadi positif.

FN (False Negative) = Data positif yang terklasifikasi menjadi negatif.

Keempat parameter tersebut digunakan untuk menghitung metode evaluasi yakni:

a. Accuracy, adalah jumlah proporsi prediksi yang benar. Adapun rumus perhitungan akurasi dapat dilihat dari persamaan berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (2.8)$$

b. Precision, adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen teks yang terpilih oleh sistem. Rumus precision dapat dilihat pada persamaan berikut:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2.9)$$

c. Recall, adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen teks yang ada pada koleksi. Rumus recall dapat dilihat pada persamaan berikut:

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.10)$$

d. F-Score, adalah nilai yang mewakili semua kinerja sistem yang merupakan penggabungan nilai Recall dan Precision. Rumus F-Score dapat dinyatakan dengan:

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100\% \quad (2.11)$$

e. Error rate, adalah nilai sebagai jumlah semua prediksi yang salah dari total kumpulan data. Rumus Error rate dapat dinyatakan dengan:

$$Error\ rate = \frac{FP+FN}{TP+TN+FN+FP} \times 100\% \quad (2.12)$$