BAB II
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan proses yang digunakan untuk mengenali dan mengklasifikasikan opini yang diekspresikan dalam teks, terutama untuk mengidentifikasi sikap penulis terhadap topik tertentu, apakah positif, negatif, atau netral. Teknik ini sering digunakan dalam analisis ulasan produk, umpan balik pelanggan, dan media sosial untuk memahami pendapat publik. Analisis sentimen kini menjadi alat yang sangat penting. dalam pemasaran, layanan pelanggan, dan riset politik (Liu, 2012). Algoritma yang digunakan dalam analisis sentimen sering kali melibatkan teknik pemrosesan bahasa alami *NLP (Neuro Linguistic Program)* untuk mengekstraksi informasi subjektif dari data teks. Perkembangan terbaru dalam analisis sentimen mencakup penggunaan deep learning dan model bahasa yang lebih canggih seperti transformer, yang memungkinkan analisis yang lebih tepat dan kontekstual. Dalam analisis sentimen, data biasanya diambil dari sumber-sumber seperti ulasan produk, komentar media sosial, dan artikel berita. Proses ini melibatkan beberapa tahap, termasuk *Preprocessing* teks (seperti tokenisasi, penghapusan *stopwords*, dan *stemming*), ekstraksi fitur, dan klasifikasi sentimen. Salah satu tantangan utama dalam analisis sentimen adalah menangani ironi dan sarkasme, yang bisa membuat analisis menjadi kurang akurat. Selain itu, pendekatan berbasis machine learning yang lebih baru, seperti model *BERT* dan *GPT*, telah menunjukkan peningkatan kinerja yang signifikan dalam tugas-tugas analisis sentimen, karena kemampuannya untuk memahami konteks dan nuansa bahasa.

Analisis sentimen berfokus pada pengelompokan ulasan berdasarkan polaritasnya. Berdasarkan pengelompokan ini, analisis Sentimen terbagi dalam dua kategori utama: pertama, pengelompokan dokumen berdasarkan apakah berisi opini atau fakta, yang disebut klasifikasi subjektivitas, dan kedua, pengelompokan dokumen menjadi positif, negatif, atau netral, yang disebut analisis sentimen.

Proses ini penting untuk menentukan dokumen yang mengandung opini dan apakah opini tersebut bernilai positif, negatif, atau netral (Nurhuda, Sihwi, & Doewes, 2016).

2.2 *Text Mining*

Proses penemuan informasi yang bermanfaat dari data teks yang tidak terstruktur. Teknik ini sering digunakan untuk mengambil informasi dari dokumen teks besar, seperti artikel berita, ulasan produk, dan posting media sosial. *Text Mining* mencakup berbagai metode seperti clustering, klasifikasi, dan analisis sentimen. Salah satu tantangan utama dalam *Text Mining* adalah mengatasi ambiguitas dan variasi dalam bahasa alami (Aggarwal & Zhai, 2012). Oleh karena itu, penggunaan teknik *NLP* menjadi sangat penting untuk meningkatkan akurasi dan efisiensi *Text Mining*.

Text Mining melibatkan beberapa tahapan, termasuk *Preprocessing* teks untuk membersihkan dan mempersiapkan data, ekstraksi fitur untuk mengidentifikasi elemen penting dalam teks, dan penerapan algoritma *machine learning* untuk mengklasifikasikan atau mengelompokkan teks. *Text Mining,* yang juga dikenal sebagai Data Mining berbasis teks atau penemuan pengetahuan dari basis data teks, adalah proses untuk memperoleh informasi dari kumpulan dokumen dengan menggunakan alat analisis yang merupakan bagian dari Data Mining, seperti yang dijelaskan dalam buku *The Text Mining Handbook* (Feldman & Sanger, 2007). Tujuan utama dari Text Mining adalah untuk mengambil informasi yang berguna dari sekumpulan dokumen. Data yang digunakan dalam Text Mining umumnya berupa teks dalam format yang tidak terstruktur atau setidaknya semi-terstruktur. Beberapa tugas utama dalam Text Mining mencakup pengkategorian teks dan pengelompokan teks.

2.3 X

X merupakan platform media sosial yang memungkinkan pengguna untuk mengirim dan membaca pesan singkat dengan batasan hingga 280 karakter. X telah menjadi alat yang efektif untuk komunikasi real-time dan berbagi informasi secara cepat. X memiliki dampak signifikan dalam bidang komunikasi, jurnalistik, dan

penelitian opini publik. Fitur seperti hashtag dan trending topics memudahkan pengguna untuk mengikuti diskusi tentang topik tertentu dan mengukur sentimen publik (Murthy, 2013).

X sering digunakan dalam penelitian untuk menganalisis opini publik dan tren sosial. Data dari X dapat digunakan untuk mempelajari reaksi terhadap peristiwa tertentu, mengidentifikasi topik yang sedang tren, dan memahami sikap publik terhadap isu-isu tertentu. Selain itu, X juga telah menjadi platform penting untuk pemasaran digital dan manajemen reputasi, di mana perusahaan dapat memantau dan merespons umpan balik pelanggan secara real-time. Dengan analisis sentimen dan teknik *Text Mining*, peneliti dan profesional dapat mengekstraksi wawasan berharga dari data X untuk mendukung pengambilan keputusan dan strategi bisnis.

X *API* adalah antarmuka menghubungkan aplikasi pihak ketiga yang membantu melakukan berbagai tugas ke platform media sosial. Perubahan terbaru pada fitur ini telah membuat banyak layanan pihak ketiga kehilangan akses ke *API* X. Meskipun akses ke *API* platform sebagian besar gratis, perubahan terbaru telah menyertakan pembatasan tertentu pada fitur tersebut. Misalnya, ada biaya yang berbeda untuk tingkatan *API* yang berbeda (TweetDelete, 2024)).

2.4 *Text Preprocessing*

Text Preprocessing adalah langkah yang krusial dalam analisis teks, termasuk analisis sentimen. Proses ini bertujuan untuk membersihkan dan mempersiapkan teks agar dapat diproses oleh algoritma machine learning. Tahapan dalam *Text Preprocessing* meliputi *Casefolding*, *cleanesing, tokenizing, stopword removal, stemming.*

Tahapan *Preprocessing* dalam pengolahan teks terdiri diantaranya sebagai berikut:

1. *Casefolding*

Proses ini mengubah semua huruf dalam teks menjadi huruf kecil. Tujuannya adalah untuk memastikan bahwa kata-kata yang seharusnya dianggap sama tidak

diperlakukan berbeda hanya karena perbedaan huruf besar/kecil. Misalnya, kata "*Text*" dan "*text*" akan dianggap sama setelah proses ini.

2. *Cleansing*

Pada tahap ini, teks dibersihkan dari karakter atau elemen yang tidak relevan seperti tanda baca, simbol, atau karakter khusus lainnya yang tidak diperlukan dalam analisis.

3. *Tokenizing*

Proses ini memecah teks menjadi unit-unit yang lebih kecil, biasanya kata atau token. Setiap kata dalam kalimat dipecah menjadi token individual.

4. *Stopword Removal*

Stopword adalah kata-kata umum yang sering muncul dalam teks tetapi dianggap tidak memiliki makna penting dalam analisis, seperti "yang", "dan", "di", "ke", dll. Proses ini menghapus stopword dari teks.

5. *Stemming*

Stemming adalah proses mengubah kata-kata ke bentuk dasarnya atau akarnya. Tujuannya adalah untuk mengurangi variasi kata ke bentuk dasarnya agar lebih mudah dianalisis.

2.5 Pembobotan Kata *TF-IDF*

TF-IDF adalah ukuran statistik yang digunakan untuk menunjukkan seberapa penting sebuah kata dalam sebuah dokumen dalam korpus (kumpulan dokumen). *TF-IDF* adalah produk dari dua statistik: *Term Frequency (TF)* dan *Inverse Document Frequency (IDF).*

1. *Term Frequency (TF)*

Mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Terdapat beberapa cara untuk menghitung TF, namun cara yang paling umum digunakan adalah:

di mana:

$$TF(t,d) = \frac{f_{t,d}}{N_d} \quad (2.1)$$

a. $f_{t,d}$ adalah jumlah kemunculan term *t* dalam dokumen *d*.

b. N_d adalah jumlah total term dalam dokumen *d*.

2. *Inverse Document Frequency (IDF)*

Inverse Document Frequency mengukur kepentingan sebuah term dalam korpus. Semakin jarang sebuah term muncul di dokumen-dokumen lain, semakin tinggi nilai *IDF*-nya. Rumus yang umum digunakan adalah:

$$IDF(t,d) = \log\left(\frac{N}{|\{d \in D: t \in d\}|}\right) \quad (2.2)$$

di mana

a. *N* adalah jumlah total dokumen dalam korpus.

b. |{d∈D:t∈d}| adalah jumlah dokumen yang mengandung term *t*.

TF-IDF membantu menyoroti kata-kata yang lebih penting dalam dokumen berdasarkan frekuensi kata tersebut dalam dokumen itu dan di seluruh korpus. Ini sangat berguna dalam berbagai aplikasi seperti pemrosesan teks, pencarian informasi, dan analisis sentimen.

2.6 *Naïve Bayes Classifier*

Algoritma Naive Bayes merupakan salah satu metode klasifikasi yang sederhana namun efektif, terutama dalam masalah klasifikasi teks seperti analisis sentimen. Metode ini didasarkan pada Teorema Bayes dengan asumsi bahwa fitur-fitur yang digunakan untuk klasifikasi bersifat independen satu sama lain.

Teorema Bayes menyatakan bahwa:

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (2.3)$$

Di mana X adalah data tuple yang diperoleh dari pengujian pada sebuah set data tertentu dan dikategorikan ke dalam kelas tertentu. H merepresentasikan hipotesis yang menentukan apakah X termasuk dalam kelas C. Probabilitas P(H|X) menunjukkan kemungkinan bahwa X, sebagai data tuple atau bukti yang dihasilkan dari observasi, merupakan anggota kelas C. Probabilitas ini dikenal sebagai probabilitas posterior, yaitu probabilitas hipotesis H dengan asumsi adanya bukti X. Sebaliknya, P(H) adalah probabilitas prior atau probabilitas awal yang tidak bergantung pada data observasi. Sementara itu, P(X|H) merepresentasikan probabilitas X dengan asumsi bahwa hipotesis H benar, dan P(X) adalah probabilitas awal atau probabilitas keseluruhan dari X.

Dalam penelitian ini, aturan Bayes akan diterapkan pada studi kasus yang spesifik. Dengan demikian, aturan Bayes dapat dijelaskan sebagai berikut:

$$P(C_J|X) = \frac{P(X|C_J).P(C_J)}{P(X)} \quad (2.4)$$

Di mana cj adalah kategori teks yang akan diklasifikasikan, dan p(cj) adalah probabilitas prior dari kategori teks cj. Sementara itu, d adalah dokumen teks yang direpresentasikan sebagai himpunan kata (W1, W2, ... Wn), di mana W1 adalah kata pertama, W2 adalah kata kedua, dan seterusnya. Dalam proses pengklasifikasian dokumen teks, pendekatan Bayes akan memilih kategori dengan probabilitas tertinggi (CMAP), yaitu:

$$C_{MAP} = argmax \frac{P(C_J).P(X|C_J)}{P(X)} \quad (2.5)$$

Nilai p(X) dapat diabaikan karena merupakan konstan untuk semua cjc_jcj, sehingga persamaan (2.6) dapat disederhanakan menjadi:

$$C_{MAX} = argmax\, p(C_j)\, p(X|C_j) \quad (2.6)$$

Probabilitas p(cj) dapat diperkirakan dengan menghitung jumlah dokumen pelatihan untuk setiap kategori cj. Namun, menghitung distribusi p(X|cj) menjadi sulit karena jumlah istilah yang sangat besar, yaitu kombinasi posisi kata yang

dikalikan dengan jumlah kategori yang akan diklasifikasikan. Dengan pendekatan Naïve Bayes yang mengasumsikan bahwa setiap kata dalam kategori tersebut saling independen, perhitungan dapat disederhanakan seperti berikut:

$$C_{MAX} = argmax\, p(C_j)\, p(X|C_j) \tag{2.7}$$

Dengan menerapkan persamaan (2.4), persamaan (2.8) dapat disederhanakan menjadi:

$$C_{MAX} = argmax\, p(C_j)\, p(X|C_j) \tag{2.8}$$

Nilai p(cj) dan p(wi | cj) dihitung selama proses pelatihan dengan menggunakan persamaan berikut:

$$p\,(C_j) = \frac{|docs\ j|}{|contoh|} \tag{2.9}$$

$$p(w_i|c_j) = \frac{1 + n_i}{|c| + n(kosakata)} \tag{2.10}$$

p(wi|cj) = menggambarkan probabilitas terjadinya kata wi dalam kategori cj.

|docs j| = dokumen yang ada pada kategori j.

|contoh| = total jumlah dokumen yang digunakan dalam pelatihan.

ni = jumlah kemunculan kata wi padakategori cj.

| C | = totoal jumlah semua kata pada kategori cj.

2.7 Python

Python adalah bahasa yang sangat direkomendasikan. Python menjadi salah satu dari tiga bahasa pemrograman paling banyak digemari selama beberapa tahun terakhir. Dengan perpustakaan yang luas, Python memungkinkan pengembang untuk menggunakannya di berbagai bidang. Beberapa pustaka atau framework populer untuk data science dan machine learning yang menggunakan Python adalah *Scikit-Learn, TensorFlow*, dan PyTorch. Meskipun demikian, Python bukanlah bahasa pemrograman yang baru. Python adalah bahasa pemrograman serbaguna

yang dirancang dengan fokus pada kemudahan pembacaan kode sumber. Selain itu, Python dilengkapi dengan berbagai pustaka yang komprehensif, memungkinkan pengembang untuk membuat aplikasi yang kompleks dan canggih (Harahap, 2024). Python memiliki berbagai pustaka yang sangat berguna untuk analisis teks dan machine learning, seperti *NLTK, Scikit-learn, Pandas, dan Numpy*.

NLTK (Natural Language Toolkit) adalah pustaka yang menyediakan alat-alat untuk pemrosesan teks, termasuk tokenisasi, *stemming, lemmatization,* dan penghapusan *stopwords* (Bird, Klein, & Loper, 2009). Scikit-learn adalah pustaka *machine learning* yang menyediakan berbagai algoritma klasifikasi, termasuk *Naive Bayes*, serta alat untuk evaluasi model (Pedregosa et al., 2011). Python juga memiliki pustaka Pandas dan Numpy yang digunakan untuk manipulasi data dan komputasi numerik. Pandas memungkinkan manipulasi data yang mudah dan efisien dalam bentuk *DataFrame*, sedangkan Numpy menyediakan dukungan untuk *array multidimensional* dan berbagai fungsi matematis.

2.8 Evaluasi

Evaluasi model merupakan langkah krusial untuk mengukur kinerja model klasifikasi. Salah satu metode evaluasi yang sering digunakan adalah confusion matrix. Confusion matrix adalah tabel yang memperlihatkan perbandingan antara prediksi model dan label yang sebenarnya, yang terdiri dari empat komponen utama: *True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN*) (Powers, 2011).

Tabel 2.1 *Confusion matrix*

	Predicted Class		
True Class	**Negatif**	**Netral**	**Positif**
Negatif	TNF	FNF	FNF
Netral	FNL	TNL	FNL
Positif	FP	FP	TP

Penjelasan singkat tentang elemen-elemen dalam *confusion matrix:*

a. *TP (True Positive):* Jumlah sampel yang benar-benar Positif dan diprediksi sebagai Positif.
b. *TNV (True Negative):* Jumlah sampel yang benar-benar Negatif dan diprediksi sebagai Negatif.
c. *TNL (True Neutral)* Jumlah sampel yang benar-benar Netral dan diprediksi sebagai Netral.
d. *FP (False Positive):* Jumlah sampel yang sebenarnya bukan kelas tertentu tetapi diprediksi sebagai kelas tersebut. Misalnya, Negatif yang diprediksi sebagai Positif.
e. *FNV (False Negative):* Jumlah sampel yang sebenarnya adalah kelas tertentu tetapi diprediksi bukan sebagai kelas tersebut. Misalnya, Positif yang diprediksi sebagai Negatif.
f. *FNL (False Neutral):* Jumlah sampel yang sebenarnya adalah kelas tertentu tetapi diprediksi bukan sebagai kelas tersebut. Misalnya, Positif yang diprediksi sebagai Netral.

Akurasi adalah metrik yang menghitung persentase prediksi benar dari keseluruhan prediksi, dihitung sebagai:

$$Akurasi = \frac{TP + TNL + TNF}{TP + TNL + TNF + FP + FNL + FNF} \quad (2.11)$$

Precision adalah metrik yang menunjukkan persentase prediksi positif yang benar-benar positif, dihitung sebagai:

$$Precision = \frac{T(Class)}{T(Class) + F(Class)} \quad (2.12)$$

Recall adalah metrik yang menunjukkan persentase kasus positif yang berhasil diidentifikasi dengan benar, dihitung sebagai:

$$Recall = \frac{T(Class)}{T(Class) + F(Class)} \quad (2.13)$$

F1-score adalah metrik yang menggabungkan precision dan recall menjadi satu nilai, dihitung sebagai:

$$F1 - score = 2 \,.\, \frac{Precision(Class)\ x\ Recall(Class)}{Precision(Class) + Recall(Class)} \qquad (2.14)$$

Metrik-metrik ini memberikan gambaran yang lebih komprehensif tentang kinerja model, khususnya dalam konteks klasifikasi yang tidak seimbang.

2.9 Google Collaboratory

Google Collaboratory, atau Google Colab, adalah platform berbasis cloud yang tersedia secara gratis untuk keperluan penelitian. Alat ini dirancang dengan lingkungan Jupyter dan mendukung hampir semua pustaka yang diperlukan untuk pengembangan Kecerdasan Buatan (AI). Secara fungsi, Google Colab serupa dengan Jupyter Notebook, namun dijalankan secara online dengan akses gratis.

2.10 Penelitian Sejenis

Terdapat penelitian-penelitian sebelumnya yang telah dilakukan oleh orang lain yang serupa dan menjadi referensi untuk penelitian ini. Beberapa di antaranya adalah sebagai berikut:

1. Putri Dwi Rahayu Maulidiana dkk. (2024) melakukan penelitian yang membandingkan algoritma *Naïve Bayes (NB)* dan *Support Vector Machine (SVM)* untuk menganalisis sentimen masyarakat terhadap putusan Mahkamah Agung, khususnya yang berfokus pada kasus Ferdy Sambo. Penelitian ini bertujuan untuk memahami persepsi masyarakat terhadap putusan pengadilan dan mengevaluasi efektivitas algoritma NB dan SVM dalam mengklasifikasikan sentimen yang diungkapkan di media sosial. Studi tersebut menemukan bahwa algoritma SVM mencapai akurasi 84%, presisi 61%, recall 78%, dan skor F1 66%. Sebagai perbandingan, algoritma NB memiliki akurasi 73%, presisi 37%, recall 57%, dan skor F1 35%, menunjukkan bahwa SVM mengungguli NB dalam konteks ini. Temuan menunjukkan bahwa SVM adalah alat yang lebih efektif untuk analisis sentimen dalam kasus ini, memberikan wawasan berharga bagi pembuat kebijakan dan profesional hukum mengenai opini publik terhadap keputusan pengadilan (Maulidiana et al., 2024).

2. Indriyani dkk (2023) dalam penelitiannya menggunakan metode *Naive Bayes* untuk menganalisis sentimen pengguna X terhadap vaksin Covid-19. Tujuan dari penelitian ini adalah untuk memahami opini publik mengenai vaksinasi Covid-19 dan membantu pemerintah dalam menyusun strategi komunikasi yang efektif untuk meningkatkan kesadaran dan penerimaan vaksin di masyarakat dengan akurasi mencapai 85%. Penelitian ini membantu pemerintah dalam strategi komunikasi vaksinasi (Indriyani, Sari, & Febriyanto, 2021).
3. Rani Puspita dan Agus Widodo (2020) dalam penelitiannya membandingkan metode *K-Nearest Neighbors (KNN), Decision Tree*, dan *Naive Bayes* dalam analisis sentimen terhadap layanan BPJS di X. Hasil penelitian menunjukkan bahwa metode *Naïve Bayes* mencapai akurasi tertinggi sebesar 98.40%., sementara *KNN* dan *Decision Tree* juga memberikan hasil yang kompetitif. Penelitian ini bertujuan untuk membantu BPJS dalam memahami kepuasan dan keluhan pengguna layanan mereka (Puspita & Widodo, 2021).
4. Rahman dkk. (2024) dalam penelitiannya menggunakan metode *Naive Bayes* untuk mengembangkan platform media sosial yang dirancang khusus bagi penyandang disabilitas. Penelitian ini mengevaluasi bagaimana algoritma *Naive Bayes* dapat digunakan untuk mengklasifikasikan sentimen dan preferensi pengguna, dengan tujuan akhir untuk meningkatkan interaksi sosial dan dukungan bagi komunitas disabilitas. Akurasi yang dicapai dengon metode *Naive Bayes* adalah 82% untuk mengklasifikasikan sentimen dan preferensi pengguna (Rahman et al., 2024).
5. Musthofa Galih dkk (2022) dalam penelitiannya membandingkan efektivitas metode *Support Vector Machine (SVM)* dan *Naive Bayes* dalam mengklasifikasikan peluang terjadinya serangan jantung berdasarkan data medis pasien. Penelitian ini menunjukkan bahwa kedua metode memiliki keunggulan masing-masing, namun Naive Bayes menunjukkan hasil yang lebih stabil dalam klasifikasi kasus yang lebih kompleks. membandingkan *SVM* dan *Naive Bayes* dalam klasifikasi penyakit jantung. Penelitian ini menggunakan dua skenario pengujian, yaitu pembagian data latih sebesar 20% pada skenario pertama dan 40% pada skenario kedua. Hasil akhir menunjukkan bahwa akurasi terbaik

dicapai oleh metode Support Vector Machine pada skenario pertama, dengan nilai akurasi sebesar 87% (Pradana, Saputro, & Wijaya, 2022).