BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen, yang juga dikenal sebagai penambangan opini, merupakan proses dalam komputasi yang digunakan untuk mengidentifikasi, mengevaluasi, dan mengelompokkan pendapat, perasaan, atau sikap yang terdapat dalam suatu teks. Tujuan utama dari analisis ini adalah untuk mengetahui kecenderungan isi suatu teks, apakah bersifat positif, negatif, atau netral [6].

2.2 YouTube

Platform YouTube pertama kali diluncurkan pada tahun 2005 oleh tiga pendirinya: Chad Hurley, Steve Chen, dan Jawed Karim. Nama platform ini terinspirasi dari sebuah tempat makan yang berada di San Mateo, California. Sejak diluncurkan, YouTube menunjukkan perkembangan yang sangat pesat. Hingga pertengahan tahun 2006, jumlah video yang diunggah mencapai lebih dari 100.000, dan pada periode tersebut YouTube juga mulai menjalin kerja sama komersial, salah satunya dengan NBC dalam bidang pemasaran. Perkembangan semakin pesat setelah Google mengakuisisi YouTube pada Oktober 2006 senilai USD 1,65 miliar, menjadikannya platform video terbesar di dunia. Pada 2010, YouTube mulai menyiarkan acara olahraga seperti Indian Premier League secara gratis, dan setahun kemudian terintegrasi dengan Google+, memperluas jangkauan serta interaksi pengguna. Kini, YouTube tidak hanya menjadi platform berbagi video, tetapi juga sarana edukasi, hiburan, dan pemasaran digital yang terus berkembang di era modern [7].

2.3 Text Pre-Processing

Tahap *Text Pre-Processing* merupakan langkah awal dalam pengolahan data teks yang bertujuan untuk meningkatkan kualitas data sebelum digunakan dalam analisis lebih lanjut. Data mentah sering kali mengandung elemen yang tidak konsisten, seperti simbol, angka, imbuhan, serta kata-kata yang tidak memiliki relevansi terhadap sentimen yang dianalisis. Keberadaan elemen-elemen ini dapat mengganggu kinerja model dalam proses klasifikasi. Oleh karena itu,

proses ini difokuskan pada pembersihan data dari unsur yang tidak diperlukan, mengurangi duplikasi, serta menstandarisasi format teks agar lebih seragam. Dengan pendekatan ini, data yang telah diproses menjadi lebih terstruktur dan siap digunakan untuk berbagai teknik analisis, seperti klasifikasi dokumen, clustering, ekstraksi informasi, analisis sentimen, serta information retrieval [8]. Adapun langkah-langkah yang dilakukan dalam *Text Pre-Processing* adalah sebagai berikut:

- *Cleansing* proses ini bertujuan untuk membersihkan dokumen dengan menghapus karakter-karakter yang tidak diperlukan, seperti tanda baca, tautan, tagar, mention, emotikon, dan sebagainya.
- *Case Folding* proses ini mencakup penyeragaman format huruf dalam sebuah dokumen. Langkah ini diperlukan karena dalam dokumen teks sering kali terdapat inkonsistensi dalam penggunaan huruf besar dan kecil [9]. Oleh karena itu, peran *case folding* diperlukan untuk menyesuaikan seluruh teks dalam sebuah dokumen dengan mengonversi huruf kapital ke dalam format standar, yang umumnya menggunakan huruf kecil.
- *Tokenizing* yang merupakan proses memisahkan teks menjadi kata-kata individu agar lebih mudah dianalisis dalam sebuah dokumen percakapan [10]. Proses ini menghilangkan tanda baca seperti titik (.), Koma (,), tanda tanya (?), dan lain-lain.
- *Stopword Removal* merupakan proses menyaring dan menghilangkan kata-kata umum yang tidak memberikan kontribusi signifikan terhadap makna keseluruhan dokumen [11].
- *Normalization* yang merupakan proses mengonversi kata-kata tidak baku, seperti slang atau variasi ejaan, menjadi bentuk yang lebih formal dan standar [12].
- *Stemming* merupakan proses mengubah kata yang memiliki imbuhan menjadi bentuk dasarnya. Namun, proses ini dapat menyebabkan perubahan makna, karena suatu kata dapat memiliki arti yang berbeda setelah diberi imbuhan.

2.4 Labeling

Pada tahap labeling atau pelabelan data sentimen, penelitian ini menggunakan pendekatan berbasis lexicon, yaitu metode yang mengandalkan daftar kata yang

telah diberikan skor atau label sentimen tertentu. Salah satu kamus lexicon yang digunakan secara luas untuk Bahasa Indonesia adalah InSet (Indonesian Sentiment Lexicon), yang dikembangkan oleh Fajri Koto dan Gemala Y. Rahmaningtyas (2017). Kamus ini dirancang khusus untuk mendukung analisis sentimen pada data berbahasa Indonesia, terutama yang berasal dari media sosial. InSet Lexicon terdiri dari sekitar 3.600 kata positif dan 6.600 kata negatif. Kata-kata ini diperoleh melalui proses semi-otomatis dari berbagai sumber seperti korpus berita, kamus Bahasa Indonesia, dan konten dari media sosial. Setiap entri dalam kamus ini telah dianotasi oleh beberapa anotator untuk memastikan validitas dan konsistensi label sentimen yang diberikan. Dalam proses pelabelan, setiap teks atau kalimat dianalisis berdasarkan jumlah dan jenis kata yang terdapat dalam kamus InSet. Jika jumlah kata positif lebih banyak daripada negatif, maka teks tersebut diberi label sentimen positif. Sebaliknya, jika kata negatif lebih dominan, maka teks dilabeli sebagai negatif. Dengan pendekatan ini, proses pelabelan data dapat dilakukan secara otomatis tanpa memerlukan anotasi manual dalam jumlah besar. Penggunaan metode berbasis lexicon seperti InSet memiliki keunggulan dalam hal kesederhanaan dan transparansi, serta sangat berguna dalam kondisi ketika tidak tersedia data latih berlabel yang cukup untuk membangun model pembelajaran mesin. Oleh karena itu, dalam penelitian ini, InSet digunakan sebagai alat utama dalam proses labeling data sentimen[13].

2.5 SMOTE

SMOTE(*Synthetic Minority Over-sampling Technique*) merupakan salah satu metode yang umum digunakan untuk menangani ketimpangan distribusi kelas pada data. Alih-alih melakukan duplikasi langsung terhadap data minoritas, SMOTE bekerja dengan menghasilkan data sintetis baru. Proses ini dilakukan dengan membuat kombinasi linier (convex combination) dari sampel minoritas yang berdekatan dalam ruang fitur, sehingga menghasilkan sampel baru yang merepresentasikan karakteristik umum dari kelas minoritas tersebut. Dengan cara ini, distribusi kelas dalam dataset menjadi lebih seimbang dan mengurangi risiko overfitting yang sering terjadi pada metode oversampling tradisional yang hanya menggandakan data [14].

2.6 Multinomial Naive Bayes

Multinomial Naive Bayes merupakan model turunan dari algoritma *Naive Bayes* yang terbukti baik dalam menyelesaikan masalah klasifikasi teks. Model ini mengasumsikan bahwa kemunculan suatu fitur atau dalam hal ini kata bersifat tidak tergantung terhadap fitur lainnya untuk setiap kelas yang ada[15]. *Naive Bayes Classifier* dikenal dengan kecepatan serta akurasinya dalam melakukan klasifikasi. Rumus dasar dari *Naive Bayes* dapat dituliskan sebagai berikut:

$$P(w_i|x) = \frac{P(x|w_i) \cdot P(w_i)}{P(x)} \tag{2.1}$$

Dengan penjelasan sebagai berikut:

1. $P(x|w_i)$ = peluang kata x muncul di kelas w.
2. $P(w_i)$ = peluang kata pada kelas c.
3. $P(x)$ = peluang kemunculan kata x.

Proses klasifikasi dengan *Multinomial Naive Bayes* dilakukan dengan menghitung total frekuensi kemunculan kata dalam masing-masing kelas yang terdapat di setiap dokumen [16]. Perhitungan ini mengacu pada persamaan berikut:

$$C_{map} = \arg\max P(c|d) \prod_{1 \leq k \leq n_d} P(t_k|C) \tag{2.2}$$

Keterangan:

1. $\arg\max$ = mencari nilai *posterior probability* tertinggi dari setiap kelas.
2. $P(t_k|C)$ = *Conditional probability*, yaitu probabilitas kemunculan kata k pada kelas tertentu.
3. $P(c)$ = *Prior probability* untuk kelas c.

Perhitungan nilai *prior probability* $P(c)$ dilakukan menggunakan rumus:

$$P(c) = \frac{N_c}{N} \tag{2.3}$$

Keterangan:

- N_c = jumlah dokumen pada kelas c.
- N = total jumlah dokumen.

Untuk menghitung nilai *conditional probability*, digunakan rumus sebagai berikut:

$$P(t_k|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}} \tag{2.4}$$

Keterangan:

- T_{ct} = frekuensi kemunculan kata t pada dokumen dalam kelas c.
- $\sum_{t' \in V} T_{ct'}$ = jumlah total seluruh kata dalam kelas c.

Namun, dalam praktiknya, terdapat kemungkinan bahwa suatu kata tidak pernah muncul pada kelas tertentu, sehingga menyebabkan nilai *conditional probability* menjadi nol [16]. Hal ini dapat mengganggu proses klasifikasi. Untuk mengatasi permasalahan tersebut, digunakan metode *Laplace Smoothing*, yaitu dengan menambahkan angka satu pada setiap nilai yang dihitung, seperti ditunjukkan pada rumus:

$$P(t_k|c) = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'} \tag{2.5}$$

Dengan:

- B' = jumlah keseluruhan kosakata unik dari seluruh kelas dalam dokumen.

2.7 TermFrequency-Inverse Document Frequency (TF-IDF)

1. *Term Frequency* (TF)

   Term Frequency merupakan metode pembobotan kata yang paling dasar, yang dihitung menggunakan rumus berikut[16]:

   $$tf(t,d) = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.6}$$

   Keterangan:

   - $tf(t,d)$ = Frekuensi *term* t dalam dokumen d

- $n_{i,j}$ = Jumlah kemunculan *term* i dalam dokumen j
- $\sum_k n_{k,j}$ = Total seluruh kata dalam dokumen j

2. *Inverse Document Frequency* (IDF)

   Inverse Document Frequency digunakan untuk mengukur pentingnya suatu kata dengan melihat keberadaannya di seluruh kumpulan dokumen. Semakin sedikit dokumen yang mengandung kata tersebut, maka semakin tinggi nilai IDF-nya. Rumus IDF dituliskan sebagai berikut:

$$idf = \log \frac{N}{df_j} \tag{2.7}$$

   Keterangan:

   - N = Jumlah total dokumen
   - df_j = Jumlah dokumen yang mengandung term i

3. Menghitung TF-IDF

   Nilai TF-IDF diperoleh dengan mengalikan hasil dari TF dan IDF menggunakan rumus:

$$w_{ij} = tf_{ij} \times idf \tag{2.8}$$

   Keterangan:

   - w_{ij} = Bobot dari kata i dalam dokumen j
   - tf_{ij} = Frekuensi kemunculan kata i dalam dokumen j
   - idf = Nilai Inverse Document Frequency dari kata i

2.8 Confusion Matrix

Confusion Matrix adalah sebuah metode evaluasi yang digunakan dalam *machine learning* untuk mengukur kinerja model klasifikasi dengan membandingkan hasil prediksi dengan nilai sebenarnya [17].

Tabel 2.1. *Confusion Matrix*

Prediction	***Actual Positive***	***Actual Negative***
Predicted Positive	TP (*True Positive*)	FP (*False Positive*)
Predicted Negative	FN (*False Negative*)	TN (*True Negative*)

Keterangan untuk Tabel 2.1 dinyatakan sebagai berikut:

- ***True Positive (TP)***: Data positif yang diprediksi dengan benar.
- ***True Negative (TN)***: Data negatif yang diprediksi dengan benar.
- ***False Positive (FP)***: Data negatif yang salah diklasifikasikan sebagai positif.
- ***False Negative (FN)***: Data positif yang salah diklasifikasikan sebagai negatif.

Precision mengukur ketepatan model dalam mengidentifikasi data positif yang benar dibandingkan dengan semua prediksi positif. Rumus precision adalah:

$$Precision = \frac{TP}{TP + FP} \tag{2.9}$$

Recall mengukur seberapa baik model dapat menemukan semua data positif yang benar. Rumus recall adalah:

$$Recall = \frac{TP}{TP + FN} \tag{2.10}$$

Accuracy mengukur sejauh mana prediksi model sesuai dengan data sebenarnya, mencakup prediksi positif dan negatif yang benar. Rumus accuracy adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{2.11}$$

F1-Score adalah ukuran keseimbangan antara precision dan recall. Rumusnya sebagai berikut:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.12}$$

2.9 K-fold Cross-Validation

Cross-validation merupakan salah satu teknik validasi yang digunakan untuk mengevaluasi performa model pada data yang belum pernah dilihat sebelumnya dengan tujuan mengurangi risiko *overfitting*. Salah satu metode *cross-validation* yang paling umum digunakan adalah *K-fold cross-validation*. Model dilatih pada setiap iterasi model menggunakan *K-1 fold* sebagai data pelatihan dan *fold* yang tersisa digunakan sebagai data pengujian. Hasil evaluasi dari setiap iterasi kemudian dirata-ratakan sehingga diperoleh estimasi performa model yang lebih akurat dan representatif [18].