BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan cabang dari *text mining* yang berfokus pada identifikasi dan ekstraksi opini, emosi, serta penilaian yang terkandung dalam teks. Bidang ini juga dikenal dengan istilah *opinion mining* karena orientasinya pada pemahaman kecenderungan pendapat pengguna, bukan sekadar pencarian fakta. Menurut Liu, analisis sentimen mencakup studi tentang opini, sentimen, evaluasi, sikap, dan emosi seseorang terhadap suatu entitas, peristiwa, atau topik tertentu [4].

2.2 Text Mining dan Natural Language Processing

Text mining adalah proses penggalian informasi dan pola bermakna dari kumpulan data teks yang tidak terstruktur. Berbeda dengan data numerik yang langsung dapat diolah secara komputasional, data teks memerlukan serangkaian tahap pengolahan sebelum dapat dianalisis menggunakan algoritma pembelajaran mesin [11].

Natural Language Processing (NLP) adalah cabang kecerdasan buatan yang mempelajari cara komputer memproses dan memahami bahasa manusia. Dalam konteks analisis sentimen, NLP berperan pada tahap preprocessing yang dimana teks mentah diubah menjadi representasi yang dapat diolah secara komputasional [12]. Komentar YouTube berbahasa Indonesia menghadirkan tantangan tersendiri karena banyak mengandung bahasa informal, singkatan, campuran bahasa daerah dan bahasa Inggris, serta variasi penulisan yang tidak mengikuti kaidah baku. Kondisi ini membuat tahap preprocessing menjadi sangat penting untuk memastikan kualitas data yang digunakan dalam pelatihan model.

2.3 Teks Preprocessing

Preprocessing teks merupakan tahap pembersihan dan normalisasi data teks sebelum digunakan pada proses klasifikasi. Kualitas preprocessing secara langsung memengaruhi performa model yang dihasilkan [13]. Pada penelitian ini, preprocessing dilakukan melalui beberapa tahapan yang terbagi ke dalam dua fase.

Fase pertama adalah *cleaning* minimal yang dilakukan sebelum pelabelan.

Tujuannya bukan untuk mengubah isi teks, melainkan membuang komentar yang tidak dapat dilabeli secara bermakna, seperti komentar yang hanya berisi emoji, komentar berbahasa asing, komentar yang terlalu pendek, dan komentar mengandung spam. Teks asli tetap dipertahankan dalam bentuk aslinya agar pelabel dapat membaca dan memahami konteks komentar secara utuh.

Fase kedua adalah preprocessing NLP yang dilakukan setelah pelabelan, sebagai persiapan ekstraksi fitur untuk model klasifikasi. Tahapan yang dilakukan meliputi berikut ini.

2.3.1 Case Folding

Case folding adalah proses pengubahan seluruh karakter teks menjadi huruf kecil. Tahap ini memastikan bahwa kata yang sama dengan perbedaan kapitalisasi tidak dianggap berbeda oleh sistem.

2.3.2 Cleaning Teks

Cleaning pada fase kedua mencakup penghapusan URL, tanda pagar, *mention*, angka, tanda baca, dan karakter khusus yang tersisa setelah fase pertama, sehingga teks hanya menyisakan kata-kata bermakna.

2.3.3 Normalisasi Slang

Komentar di media sosial banyak mengandung kata tidak baku dan singkatan. Normalisasi slang dilakukan dengan mengganti kata-kata tersebut menggunakan kamus normalisasi, misalnya “yg” menjadi “yang”, “gak” menjadi “tidak”, dan “bgt” menjadi “banget”. Normalisasi ini penting untuk meningkatkan keseragaman teks sehingga kata yang bermaksud sama direpresentasikan dengan bentuk yang konsisten.

2.3.4 Tokenization

Tokenization adalah proses pemecahan kalimat menjadi satuan kata. Pada bahasa Indonesia, proses ini umumnya dilakukan berdasarkan pemisah spasi. Hasil tokenisasi menjadi unit dasar untuk tahapan selanjutnya.

2.3.5 Stopword Removal

Stopword removal adalah penghapusan kata-kata umum yang tidak membawa kontribusi makna yang signifikan dalam klasifikasi sentimen, seperti “yang”, “di”, “dan”, “ke”, dan “dari”. Daftar stopword yang digunakan bersumber dari pustaka NLTK bahasa Indonesia.

2.3.6 Stemming

Stemming adalah proses pengubahan kata berimbuhan menjadi bentuk dasarnya. Dalam bahasa Indonesia, kata “keracunan”, “meracuni”, dan “diracuni” semuanya memiliki akar kata “racun”. Stemming dilakukan menggunakan pustaka Sastrawi yang mengimplementasikan algoritma Nazief-Adriani [14].

2.4 Pelabelan Data

Pelabelan data merupakan proses pemberian anotasi kelas pada setiap data sehingga dapat digunakan sebagai data latih pada algoritma pembelajaran mesin. Pelabelan dapat dilakukan secara manual, otomatis menggunakan kamus sentimen atau model pralatih, maupun secara semi-otomatis yang menggabungkan keduanya [15].

Pada penelitian ini, pelabelan dilakukan sepenuhnya secara manual. Pendekatan ini dipilih karena data berkaitan dengan kebijakan publik yang memiliki nuansa kontekstual tinggi, sehingga penilaian manusia diperlukan untuk menangkap makna komentar yang mengandung sarkasme, sindiran, atau konteks situasional yang tidak dapat dikenali oleh sistem otomatis. Proses pelabelan mengacu pada tiga kriteria yang telah ditetapkan sebelumnya, dan untuk memastikan konsistensi, dilakukan pemeriksaan kualitas (*quality assurance*) melalui pengambilan sampel acak dari hasil pelabelan.

2.5 Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) adalah metode pembobotan kata yang digunakan untuk mengubah teks menjadi representasi numerik. TF-IDF memberikan bobot tinggi pada kata yang sering muncul dalam sebuah dokumen tetapi jarang muncul di dokumen lain, sehingga kata tersebut dianggap memiliki nilai pembeda yang tinggi [5].Pemilihan metode TF-IDF pada

penelitian ini didasarkan pada kemampuannya dalam memberikan bobot yang lebih besar kepada kata-kata yang memiliki tingkat kepentingan tinggi dalam suatu dokumen. Berbeda dengan representasi frekuensi kata biasa, TF-IDF tidak hanya mempertimbangkan frekuensi kemunculan kata dalam dokumen, tetapi juga memperhatikan tingkat penyebaran kata pada seluruh dokumen dalam korpus. Dengan demikian, kata-kata yang sering muncul pada seluruh dokumen akan memperoleh bobot yang lebih rendah, sedangkan kata-kata yang lebih spesifik terhadap suatu sentimen akan memperoleh bobot yang lebih tinggi. Karakteristik tersebut menjadikan TF-IDF banyak digunakan pada penelitian analisis sentimen dan klasifikasi teks karena mampu menghasilkan representasi fitur yang lebih informatif.

TF-IDF terdiri dari dua komponen. *Term Frequency* (TF) mengukur frekuensi kemunculan suatu kata dalam sebuah dokumen, sedangkan *Inverse Document Frequency* (IDF) mengukur tingkat keunikan kata tersebut di seluruh koleksi dokumen. Rumus TF-IDF dapat dituliskan sebagai:

$$TF(t,d) = f(t,d) \tag{2.1}$$

Keterangan: $TF(t,d)$ merupakan nilai Term Frequency dari term t pada dokumen d, sedangkan $f(t,d)$ menyatakan jumlah kemunculan term t pada dokumen d.

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

Keterangan: N menyatakan jumlah seluruh dokumen dalam korpus, sedangkan $df(t)$ merupakan jumlah dokumen yang mengandung term t.

$$TFIDF(t,d) = TF(t,d) \times IDF(t) \tag{2.3}$$

Nilai TF-IDF diperoleh dari hasil perkalian antara nilai Term Frequency dan Inverse Document Frequency sehingga menghasilkan bobot yang menunjukkan tingkat kepentingan suatu term dalam dokumen. Implementasi dalam penelitian ini menggunakan kelas `TfidfVectorizer` dari pustaka Scikit-learn [8].Pada implementasinya, proses ekstraksi fitur dilakukan menggunakan kelas

`TfidfVectorizer` yang tersedia pada pustaka *Scikit-learn*. Representasi TF-IDF digunakan sebagai masukan bagi algoritma klasifikasi karena mampu memberikan bobot yang lebih besar terhadap kata-kata yang memiliki daya pembeda tinggi antar dokumen.

2.6 Naive Bayes

Naive Bayes adalah algoritma klasifikasi probabilistik yang didasarkan pada Teorema Bayes. Disebut "naive" karena algoritma ini mengasumsikan bahwa setiap fitur bersifat independen satu sama lain, sebuah asumsi yang disederhanakan namun terbukti efektif pada banyak kasus klasifikasi teks [7].Dalam penelitian ini digunakan dua varian algoritma Naive Bayes, yaitu Multinomial Naive Bayes dan Complement Naive Bayes. Kedua algoritma tersebut diterapkan menggunakan implementasi yang tersedia pada pustaka *Scikit-learn* sehingga proses pelatihan dan pengujian model dilakukan dengan konfigurasi parameter bawaan (*default parameter*).

Teorema Bayes yang menjadi dasar algoritma ini diformulasikan sebagai:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{2.4}$$

Pada Persamaan 2.4, $P(C|X)$ merupakan probabilitas posterior suatu dokumen termasuk ke dalam kelas C setelah diketahui fitur X. Nilai $P(X|C)$ menyatakan probabilitas kemunculan fitur X pada kelas C, $P(C)$ merupakan probabilitas prior kelas, dan $P(X)$ merupakan probabilitas kemunculan fitur secara keseluruhan. Teorema Bayes digunakan sebagai dasar dalam algoritma Naive Bayes untuk menentukan kelas dengan probabilitas posterior tertinggi sebagai hasil klasifikasi.

2.6.1 Multinomial Naive Bayes

Multinomial Naive Bayes (MNB) adalah varian Naive Bayes yang dirancang untuk fitur diskrit seperti frekuensi kata. Varian ini cocok digunakan untuk klasifikasi teks karena dapat menangani data berdimensi tinggi dengan baik [6]. Dalam praktiknya, MNB juga dapat digunakan bersama representasi TF-IDF [8].

$$P(C_k|d) \propto P(C_k) \prod_{i=1}^{n} P(t_i|C_k)^{f_i} \tag{2.5}$$

Pada Persamaan 2.5, $P(C_k|d)$ merupakan probabilitas posterior dokumen d terhadap kelas C_k, $P(C_k)$ merupakan probabilitas prior kelas, $P(t_i|C_k)$ merupakan probabilitas kemunculan term t_i pada kelas C_k, dan f_i merupakan frekuensi kemunculan term ke-i dalam dokumen. Kelas dengan nilai probabilitas posterior tertinggi akan dipilih sebagai hasil klasifikasi.

2.6.2 Complement Naive Bayes

Complement Naive Bayes (CNB) merupakan perbaikan dari MNB yang dirancang khusus untuk mengatasi keterbatasan pada data dengan distribusi kelas yang tidak seimbang. Berbeda dengan MNB yang menghitung probabilitas berdasarkan kelas tertentu, CNB menghitung probabilitas berdasarkan komplemen dari kelas tersebut, sehingga lebih robust pada kondisi data tidak seimbang [16]. CNB digunakan dalam penelitian ini sebagai pembanding terhadap MNB.

$$\hat{\theta}_{ci} = \frac{\alpha + \sum_{j:y_j \neq c} d_{ij}}{\alpha n + \sum_{j:y_j \neq c} \sum_k d_{kj}} \tag{2.6}$$

Pada Persamaan 2.6, $\hat{\theta}_{ci}$ merupakan estimasi probabilitas term ke-i terhadap kelas komplemen dari kelas c, α merupakan parameter smoothing, dan d_{ij} merupakan frekuensi term pada dokumen ke-j. Berbeda dengan Multinomial Naive Bayes, Complement Naive Bayes menggunakan informasi dari seluruh kelas selain kelas target sehingga lebih robust terhadap ketidakseimbangan distribusi kelas.

2.7 Penanganan Ketidakseimbangan Data

Ketidakseimbangan kelas (*class imbalance*) merupakan kondisi ketika jumlah data pada setiap kelas tidak terdistribusi secara proporsional. Pada permasalahan klasifikasi, kondisi tersebut dapat menyebabkan model lebih cenderung memprediksi kelas mayoritas dibandingkan kelas minoritas sehingga performa klasifikasi terhadap kelas minoritas menjadi menurun [17].

Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode *SMOTE-Tomek*. Metode ini merupakan pendekatan hibrida yang menggabungkan *Synthetic Minority Over-sampling Technique* (SMOTE) dengan *Tomek Links*.

SMOTE digunakan untuk menghasilkan data sintetis pada kelas minoritas melalui proses interpolasi terhadap sampel-sampel yang memiliki kedekatan berdasarkan tetangga terdekat (*nearest neighbors*) [18].

Setelah proses oversampling dilakukan, tahap berikutnya adalah penerapan *Tomek Links*. Metode ini bertujuan mengidentifikasi pasangan data dari kelas yang berbeda namun memiliki jarak paling dekat. Pasangan tersebut umumnya berada pada batas antar kelas sehingga berpotensi menimbulkan ambiguitas dalam proses klasifikasi. Dengan menghapus pasangan data tersebut, distribusi data menjadi lebih bersih sehingga batas keputusan model dapat terbentuk dengan lebih baik [19].

Pada penelitian ini, proses *SMOTE-Tomek* hanya diterapkan pada data latih (*training set*), sedangkan data uji (*testing set*) tetap menggunakan distribusi data asli. Pendekatan tersebut dilakukan untuk menghindari terjadinya *data leakage* sehingga hasil evaluasi model tetap mencerminkan kemampuan klasifikasi terhadap data yang belum pernah dilihat sebelumnya.

2.8 Pembagian Data

Pelatihan model dilakukan dengan cara dataset dibagi menjadi data latih (*training set*) dan data uji (*testing set*). Pembagian data bertujuan agar model dapat mempelajari pola dari data latih, sedangkan data uji untuk mengevaluasi atau memvalidasi kemampuan dari model dalam melakukan prediksi terhadap data yang belum pernah digunakan atau baru [20].

Penelitian ini menggunakan metode *stratified train-test split* dengan proporsi 80% untuk data latih dan 20% untuk data uji. Pendekatan *stratified* dipilih karena mampu mempertahankan distribusi setiap kelas pada data latih maupun data uji sehingga karakteristik dataset tetap terjaga. Teknik ini sesuai digunakan pada penelitian klasifikasi dengan distribusi kelas yang tidak seimbang karena mampu menghasilkan proses evaluasi yang lebih representatif.

2.9 Confusion Matrix

Confusion Matrix merupakan metode evaluasi yang berguna untuk membandingkan hasil prediksi dari model yang telah dilatih dengan label sebenarnya. Pada klasifikasi multikelas, confusion matrix menampilkan jumlah prediksi dari yang benar maupun salah pada setiap kelas yang ada sehingga dapat digunakan untuk mengidentifikasi pola kesalahan yang dilakukan model.

Melalui confusion matrix, peneliti dapat mengetahui distribusi prediksi pada masing-masing kelas sentimen, yaitu positif, negatif, dan netral. Selain digunakan untuk mengevaluasi performa model secara visual, confusion matrix juga menjadi dasar dalam perhitungan metrik accuracy, precision, recall, dan F1-score.

2.10 Evaluasi Model Klasifikasi

Evaluasi performa model dilakukan menggunakan *confusion matrix* dan empat metrik turunannya. *Confusion matrix* adalah matriks $k \times k$ yang merangkum jumlah prediksi benar dan salah untuk setiap kelas pada data uji.

Accuracy mengukur persentase prediksi yang benar dari keseluruhan data:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

Precision mengukur ketepatan model dalam memprediksi suatu kelas:

$$Precision = \frac{TP}{TP + FP} \tag{2.8}$$

Recall mengukur kemampuan model menemukan data yang benar-benar termasuk dalam suatu kelas:

$$Recall = \frac{TP}{TP + FN} \tag{2.9}$$

F1-score merupakan rata-rata harmonik antara *precision* dan *recall*:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.10}$$

Pada data yang tidak seimbang, evaluasi tidak dapat hanya mengandalkan *accuracy* karena nilai yang tinggi dapat dicapai hanya dengan memprediksi kelas mayoritas. Oleh karena itu, penelitian ini menggunakan *macro-averaged F1-score* sebagai metrik utama, yaitu rata-rata F1-score dari seluruh kelas tanpa pembobotan berdasarkan jumlah data [21].Selain menggunakan metrik accuracy, precision, recall, dan F1-score, penelitian ini juga memanfaatkan *confusion matrix* untuk menggambarkan hasil klasifikasi setiap kelas secara lebih rinci. Confusion matrix

menunjukkan jumlah prediksi yang benar maupun salah untuk setiap kategori sentimen sehingga dapat digunakan untuk menganalisis kesalahan klasifikasi yang dilakukan oleh model.

2.11 Penelitian Terdahulu

Beberapa penelitian terdahulu yang relevan dengan penelitian ini diuraikan sebagai berikut.

Setyabudi dkk. [9] melakukan analisis sentimen komentar YouTube terhadap program MBG menggunakan model LSTM. Penelitian tersebut menggunakan 4.983 komentar dengan pelabelan otomatis berbasis TextBlob dan kata kunci spesifik MBG. Perbedaan utama dengan penelitian ini terletak pada metode klasifikasi yang digunakan, di mana penelitian ini menerapkan Naive Bayes dengan pelabelan manual.

Anwar dkk. [10] membandingkan performa SVM dan Random Forest pada 7.522 komentar YouTube terkait MBG. Hasil penelitian menunjukkan distribusi label yang sangat tidak seimbang dengan dominasi kelas netral sebesar 77,17%. Penelitian ini berbeda dalam hal metode klasifikasi (Naive Bayes) dan pendekatan pelabelan (manual).

Joshua Ardin Putra Perdana [22] menggunakan Multinomial dan Complement Naive Bayes untuk analisis sentimen terhadap isu pencampuran Pertamax dan Pertalite pada media sosial X. Alur metodologi penelitian tersebut, yang mencakup pelabelan berbasis lexicon dan perbandingan dua varian Naive Bayes, menjadi rujukan utama dalam penelitian ini.

Sabilillah Amory Reyhan Rusjdi [23] menerapkan Naive Bayes dengan penyeimbangan data ADASYN pada penelitian sentimen naturalisasi pemain timnas di media sosial X. Pendekatan penyeimbangan data tersebut diadopsi dalam penelitian ini dengan menggunakan SMOTE.

Dari tinjauan penelitian-penelitian di atas, dapat diidentifikasi bahwa belum ada penelitian yang secara spesifik menerapkan Naive Bayes dengan pelabelan manual dan SMOTE untuk analisis sentimen komentar YouTube terhadap program MBG. Penelitian ini mengisi kesenjangan tersebut sekaligus memberikan perbandingan performa antara Multinomial Naive Bayes dan Complement Naive Bayes pada kondisi data yang tidak seimbang.