BAB II
KAJIAN LITERATUR

2.1 Artificial Intelligence (AI)

Kecerdasan buatan (*Artificial Intelligence*/AI) adalah bentuk kemampuan buatan yang memungkinkan komputer meniru cara berpikir dan bertindak seperti manusia sehingga dapat menyelesaikan berbagai tugas secara mandiri [9]. Karena *Artificial Intelligence* (AI) dirancang untuk meniru cara berpikir manusia, teknologi ini memiliki keunggulan dalam proses pengambilan keputusan yang dilakukan secara cepat, efisien, dan akurat dengan memanfaatkan data yang telah tersedia [10].

Pada masa kini, kecerdasan buatan digunakan secara luas dalam berbagai bidang, mulai dari asisten suara, sistem rekomendasi, dan perangkat lunak pengenalan wajah hingga proses manufaktur otomatis, diagnosis di bidang medis, analisis keuangan, bahkan aktivitas yang bersifat kreatif. Meskipun memiliki banyak keunggulan, kecerdasan buatan juga menghadirkan sejumlah risiko, seperti meningkatnya pengangguran dan kesenjangan sosial, potensi keputusan yang keliru atau bias akibat data yang tidak berkualitas, ancaman terhadap privasi serta keamanan data pribadi, dan ketergantungan berlebihan pada teknologi yang dapat melemahkan kemampuan berpikir kritis serta pengambilan keputusan manusia [11].

2.1.1 Machine Learning

Machine learning adalah proses otomatis yang memungkinkan komputer mengidentifikasi pola dalam data secara mandiri untuk melakukan tugas-tugas seperti klasifikasi dan prediksi. Sebagai subbidang utama dari kecerdasan buatan, *machine learning* meniru kemampuan berpikir manusia dengan memproses informasi secara cerdas tanpa pemrograman eksplisit untuk setiap skenario [12].

Terdapat empat metode *Machine Learning* yaitu [13]:

1. *Supervised Learning*

   Pembelajaran yang diawasi merupakan pendekatan yang memungkinkan sistem untuk mempelajari hubungan antara data masukan dan keluaran yang diinginkan, sehingga dapat digunakan secara efektif dalam membuat prediksi berdasarkan pola yang telah dipelajari.

2. *Unsupervised Learning*

Pembelajaran tanpa pengawasan adalah metode yang digunakan untuk mengidentifikasi struktur tersembunyi, pola, atau koneksi antar data ketika data tersebut tidak memiliki label atau penanda jawaban yang jelas.

3. *Semi-supervised Learning*

   Pembelajaran semi-terawasi menggabungkan karakteristik dari kedua pendekatan pembelajaran dengan memanfaatkan dataset yang berisi campuran data yang sudah memiliki label dan data yang belum berlabel, di mana jumlah data tanpa label biasanya jauh lebih besar dibandingkan dengan data yang sudah berlabel.

4. *Reinforcement Learning*

   Pembelajaran penguatan adalah suatu sistem di mana agen (entitas pembelajaran) berinteraksi dengan lingkungan yang telah ditentukan dan menggunakan hasil atau umpan balik dari setiap tindakan yang diambil untuk terus-menerus memperbaiki dan mengoptimalkan keputusan atau tindakan di masa depan.

2.1.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan dan linguistik yang dirancang untuk membuat komputer mampu memahami pernyataan atau kata-kata dalam bahasa manusia, sekaligus mempermudah pekerjaan pengguna dengan memungkinkan mereka berkomunikasi dengan komputer menggunakan bahasa alami [14].

Saat ini, teknologi *Natural Language Processing* (NLP) telah digunakan secara luas dalam berbagai aspek kehidupan, mulai dari asisten cerdas, penerjemahan mesin, analisis sentimen, penambangan teks, peringkasan otomatis hingga sistem tanya jawab, sehingga tidak hanya meningkatkan efisiensi dalam pemrosesan informasi, tetapi juga memberikan kemudahan yang signifikan bagi aktivitas hidup dan pekerjaan manusia. Meskipun demikian, pemrosesan bahasa alami masih menghadapi sejumlah tantangan, seperti ambiguitas bahasa, pemahaman konteks, serta pengelolaan berbagai bahasa yang berbeda [15].

2.2 Text Mining

Text mining merupakan metode untuk menggali informasi dari dokumen teks guna mendukung berbagai proses, seperti klasifikasi, *clustering*, *information extraction*, dan *information retrieval* [16]. Salah satu proses dalam *text mining* adalah klasifikasi teks. Klasifikasi teks merupakan proses yang digunakan untuk menempatkan dokumen ke dalam

kelas tertentu. Dalam melakukan klasifikasi teks, beberapa algoritma yang umum digunakan antara lain *Support Vector Machine* (SVM), *Naïve Bayes*, *k-Nearest Neighbor* (KNN), *Decision Tree*, dan *Artificial Neural Networks* (ANN) dan lain-lain [17].

2.3 Pembagian Data

Pembagian data merupakan teknik pembagian dataset menjadi dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Data latih digunakan untuk melatih model agar dapat mempelajari pola dari data, sedangkan data uji digunakan untuk mengevaluasi kinerja model terhadap data yang belum pernah digunakan selama proses pelatihan [18]. Terdapat berbagai rasio yang dapat digunakan dalam pembagian data latih dan data uji, di antaranya 50:50, 60:40, 70:30, 80:20, dan 90:10. Proporsi data latih yang melebihi 70% diketahui mampu menghasilkan kinerja klasifikasi yang lebih baik karena model memperoleh data yang lebih banyak untuk proses pembelajaran [19].

Selain pembagian dataset menjadi data latih dan data uji, terdapat metode pembagian lain yang membagi dataset menjadi data latih, data validasi, dan data uji. Data validasi umumnya digunakan pada model *deep learning* untuk memantau performa model selama proses pelatihan, mengevaluasi kemampuan generalisasi, serta mengurangi risiko *overfitting* [20]. Dalam penelitian *machine learning*, pembagian data yang umum diterapkan adalah 80:20, yaitu 80% data digunakan sebagai data pelatihan (*training*) dan 20% sebagai data pengujian (*testing*) [21].

2.4 Preprocessing

Preprocessing merupakan tahap awal yang penting dalam analisis sentimen karena mengubah data mentah menjadi data yang lebih layak olah sehingga dapat dimanfaatkan secara optimal pada proses berikutnya. Tahap ini berperan dalam mengurangi kesalahan dalam pengambilan ciri atau atribut yang berpotensi menurunkan kinerja analisis sentimen secara signifikan. Dengan penerapan teknik *preprocessing* yang tepat, kualitas data meningkat sehingga mampu mendukung peningkatan kinerja klasifikasi secara keseluruhan [22].

Beberapa tahapan yang dilakukan dalam proses *preprocessing* adalah sebagai berikut [23][24][25]**:**

1. *Lowercasing*

Lowercasing adalah metode untuk mengubah seluruh kata dalam teks menjadi huruf kecil. Contoh *lowercasing*: ”Ketua DPD RI” menjadi ”ketua dpd ri”.

2. *Removing URL*, *Mention*, *Hashtag*, *and Punctuation*

   Pada tahap ini dilakukan penghapusan elemen berupa URL (tautan), *mention* (tanda ‘@’), *hashtag* (tanda ‘#’), serta tanda baca seperti titik dua pada teks.

3. *Remove number*

   Pada tahap ini dilakukan penghapusan seluruh karakter angka yang terdapat dalam teks agar data menjadi lebih bersih dan fokus pada kata-kata yang memiliki makna.

4. *Normalize space*

   Pada tahap ini dilakukan penyeragaman spasi pada teks, seperti menghapus spasi berlebih atau spasi ganda, sehingga format teks menjadi lebih rapi dan konsisten.

5. *Slang word normalization*

   Pada tahap ini, kata gaul atau kata tidak baku dinormalisasi menjadi bentuk baku agar teks lebih mudah diproses dalam analisis teks. Tahap ini penting karena kata gaul sering muncul dalam data media sosial dan dapat memengaruhi hasil pengolahan teks. Contoh normalisasi adalah kata ”gak” menjadi ”tidak”.

6. *Stopword Removal*

   Stopword adalah daftar kata yang dianggap tidak membawa makna penting. Kata-kata yang termasuk dalam daftar ini dibuang dan tidak diproses pada tahap selanjutnya. Contoh *stopword* adalah kata ”di”, ”ke”.

7. *Tokenization*

   Tokenization adalah proses pemisahan suatu *string* teks menjadi unit-unit terkecil yang disebut token, dengan menggunakan spasi atau tanda baca sebagai pemisah. Contoh *Tokenization*: {“ketua”, “dpd”, “ri”}.

8. *Stemming*

   Stemming adalah proses menghilangkan imbuhan suatu kata hingga menjadi bentuk dasarnya, meskipun bentuk dasar tersebut tidak selalu sama dengan kata akar. Dari sisi efisiensi, *stemming* bertujuan untuk mengurangi jumlah kata unik pada indeks sehingga kebutuhan ruang penyimpanan berkurang dan proses pencarian menjadi lebih cepat. Contoh *stemming* adalah kata "bekerja" menjadi "kerja".

2.5 Pelabelan Sentimen

Pelabelan sentimen merupakan proses pengelompokan untuk mengukur emosi pada setiap komentar [26]. Pelabelan sentimen dapat dilakukan pada *data training* dan *data testing* [27]. Pelabelan dapat dilakukan secara manual dan otomatis. Pelabelan manual dilakukan oleh anotator manusia terlatih secara manual dengan cara mengklasifikasikan setiap data menjadi kategori positif, negatif, atau netral berdasarkan pemahaman konteks [28]. Namun, pelabelan sentimen secara manual memiliki beberapa kendala utama, meliputi subjektivitas tinggi, durasi lama, dan biaya besar [29]. Metode pelabelan secara otomatis menjadi pilihan yang praktis untuk menggantikan atau mendukung pelabelan secara manual, yang biasanya memakan banyak waktu, bersifat subyektif, serta tidak konsisten, khususnya ketika menangani data dalam jumlah yang sangat besar [30].

Salah satu metode pelabelan otomatis untuk sentimen berbahasa Indonesia adalah pelabelan berdasarkan Leksikon InSet (*InSet Lexicon*), dimana Teknik pelabelan ini menggunakan kamus sentimen berbahasa Indonesia yang telah ada sebelumnya. Setiap kata dalam teks akan dicocokkan dengan leksikon tersebut, lalu dihitung skor sentimen keseluruhan untuk setiap data. Dengan menerapkan batas nilai tertentu, data kemudian dikelompokkan ke dalam kategori positif, negatif, atau netral [28].

2.6 Ekstraksi Fitur

Ekstraksi fitur adalah proses mengubah teks menjadi bentuk numerik agar dapat diproses oleh algoritma *machine learning* secara lebih efektif dan sesuai dengan kebutuhan analisis [25]. ekstraksi fitur dilakukan karena algoritma *machine learning* tidak dapat memproses input berbentuk *string* secara langsung. Oleh karena itu, teks perlu diolah terlebih dahulu melalui metode ekstraksi fitur yang menghasilkan bobot kata, kemudian bobot tersebut dikonversi menjadi vektor numerik untuk setiap kata dalam dokumen agar dapat digunakan dalam proses analisis oleh algoritma *machine learning*. Secara umum, proses ekstraksi fitur pada teks dilakukan melalui beberapa tahap. Pertama, teks dibersihkan melalui *preprocessing*, seperti penghapusan tanda baca, huruf kapital, dan kata yang tidak penting. Setelah itu, teks dipecah menjadi kata atau token. Lalu setiap kata diubah menjadi fitur numerik menggunakan metode tertentu sehingga hasil akhirnya berupa vektor numerik yang dapat digunakan untuk pelatihan model *machine learning* [31]. Salah satu metode ekstraksi fitur adalah TF-IDF. TF-IDF merupakan metode statistik yang digunakan untuk mengukur tingkat kepentingan sebuah kata pada sebuah dokumen dengan

mempertimbangkan kemunculannya di seluruh kumpulan dokumen. TF-IDF memadukan dua ukuran utama, yaitu *Term Frequency* (TF) yang menghitung frekuensi kemunculan suatu kata dalam satu dokumen, dan *Inverse Document Frequency* (IDF) yang menurunkan bobot kata-kata yang terlalu sering muncul di berbagai dokumen [32].

Persamaan pembobotan TF-IDF disajikan sebagai berikut [32]:

$$\text{TF-IDF}(i,j) = \text{TF}(i,j) \times \text{IDF}(i) \tag{1}$$

Dengan:

$$\text{TF}(i,j) = \frac{n(i,j)}{\sum_{k=1}^{q} n(k,j)} \tag{2}$$

$$\text{IDF}(i) = \log\left(\frac{N}{df(i)}\right) \tag{3}$$

Keterangan:

$n(i,j)$: jumlah kemunculan term t_i pada dokumen D_j.

N : jumlah seluruh dokumen dalam korpus.

$df(i)$: jumlah dokumen yang mengandung term t_i.

2.7 Klasifikasi

Klasifikasi adalah proses pengelompokan data ke dalam kelas atau kategori tertentu berdasarkan karakteristik yang dimilikinya. Dalam *machine learning*, klasifikasi merupakan bagian dari *supervised learning* karena model dilatih menggunakan data berlabel untuk memprediksi kelas pada data baru [33]. Contoh yang umum dari klasifikasi adalah *spam filter*, yaitu sistem yang digunakan untuk mengelompokkan email masuk ke dalam kategori *spam* atau bukan *spam*. Untuk melatih sistem ini, model diberikan sejumlah besar data email beserta labelnya, sehingga model dapat mempelajari pola dan ciri yang menunjukkan email *spam*, seperti kata kunci, frasa, atau format tertentu. Setelah proses pelatihan selesai, model dapat digunakan untuk mengklasifikasikan email baru berdasarkan fitur yang dimilikinya dan memprediksi label yang sesuai. *Spam filter* yang baik dapat membantu meningkatkan kenyamanan pengguna karena mampu mengurangi email yang tidak diinginkan atau berbahaya, sekaligus menjaga agar email yang sah tidak salah dikategorikan sebagai *spam* [34].

2.8 Naïve Bayes

Naïve Bayes adalah metode klasifikasi berbasis probabilitas bersyarat yang menerapkan Teorema Bayes [35]. Algortima ini melakukan klasifikasi data dengan mengasumsikan bahwa setiap fitur bersifat independen terhadap variabel kelas [36].

Kelebihan *Naïve Bayes* antara lain [37]:

1. Kesederhanaan Algoritma

   Desain yang sederhana memudahkan implementasi dan interpretasi hasil klasifikasi oleh berbagai pengguna.
2. Rendah Beban Komputasi

   Algoritma ini memberikan efisiensi optimal dalam penggunaan sumber daya komputasional, terutama pada dataset dengan volume besar dan dimensi fitur yang tinggi.
3. Skalabilitas pada Data Multidimensi

   Mampu berkinerja baik ketika fitur jauh lebih banyak dibanding jumlah sampel, menjadikannya ideal untuk aplikasi *NLP* dan *text mining*.
4. Kejelasan Hasil Prediksi

   Menghasilkan output yang transparan dengan menunjukkan probabilitas dan kontribusi masing-masing fitur dalam proses pengambilan keputusan klasifikasi.

Selain kelebihan yang disebutkan, *Naïve Bayes* memiliki kekurangan yaitu:

1. Ketergantungan pada Asumsi Independensi Fitur

   Asumsi bahwa setiap fitur independen secara kondisional terhadap label kelas sering kali tidak terpenuhi, menghasilkan prediksi yang kurang akurat ketika atribut-atribut saling berinteraksi.
2. Rentan terhadap Korelasi dan Interaksi Fitur

   Algoritma kesulitan menangkap hubungan kompleks antar fitur, sehingga akurasi klasifikasi dapat berkurang ketika terdapat ketergantungan yang kuat antar variabel penjelas.
3. Asumsi Distribusi untuk Fitur Numerik

   Pembatasan pada asumsi distribusi Gaussian membuat *Naïve Bayes* kurang fleksibel dalam menangani data numerik dengan pola distribusi yang tidak konvensional atau multimodal.
4. Ketidakefektifan pada Dataset Tidak Seimbang

   Kinerja algoritma menurun pada data dengan ketidakseimbangan kelas yang ekstrem, dengan akurasi prediksi yang buruk khususnya untuk kelas-kelas minoritas.

Berikut merupakan formulasi matematis dari perhitungan *Naïve Bayes* [38]:

$$P(C_j \mid X) = \frac{P(X \mid C_j) * P(C_j)}{P(X)} \quad (4)$$

Keterangan:

$P(C_j \mid X)$: Probabilitas hipotesis C mengingat bahwa bukti X telah diamati (hasil akhir prediksi).

$P(X \mid C_j)$: Kemungkinan bukti X akan muncul dalam kondisi hipotesis C benar (konsistensi data).

$P(C_j)$: Asumsi probabilitas awal untuk hipotesis C_j tanpa melibatkan bukti apa pun (informasi dasar).

$P(X)$: Probabilitas keseluruhan dari bukti X di semua kemungkinan skenario (konstanta penyeimbang).

Probabilitas prior $P(C_j)$ merupakan probabilitas awal suatu hipotesis tanpa mempertimbangkan bukti apapun. Nilai ini dihitung berdasarkan frekuensi kemunculan hipotesis dalam data latih menggunakan persamaan berikut [38]:

$$P(C_j) = \frac{N(C_j)}{N(sample)} \quad (5)$$

Dimana:

$P(C_j)$: Probabilitas prior hipotesis C_j

$N(C_j)$: Jumlah dokumen pada kelas C_j

$N(sample)$: Jumlah seluruh dokumen dalam data latih

$$P(X_i \mid C_j) = \frac{1 + n_i}{n + |X|} \quad (6)$$

Dimana:

$P(X_i \mid C_j)$: Probabilitas kemunculan fitur kata X_i terhadap kelas C_j

n_i : Frekuensi kemunculan kata X_i pada kelas C_j

n : Jumlah seluruh kata yang terdapat dalam dokumen pada kelas tertentu

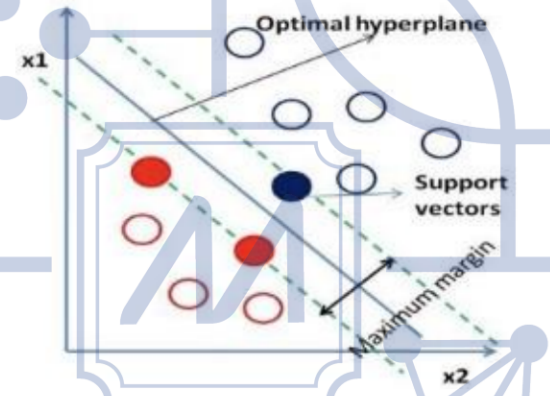
$|X|$: Banyaknya kosakata unik yang terdapat pada keseluruhan data latih

$$C_{MAP} = argmax\, P(C_j)\, P(X_i \mid C_j) \quad (7)$$

Setelah nilai hasil perkalian probabilitas untuk setiap kategori dokumen diperoleh, dilakukan proses perbandingan untuk menentukan nilai probabilitas maksimum (C_{MAP}). Kategori yang memiliki nilai probabilitas terbesar kemudian ditetapkan sebagai hasil klasifikasi dokumen [38].

2.9 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma pembelajaran dengan pengawasan yang dapat diterapkan untuk klasifikasi data, analisis regresi, dan pendeteksian *outlier*. Dikembangkan pada dekade 1990-an melalui penelitian Vladimir Vapnik beserta kolaboratornya, dengan fondasi teoretis yang kuat dalam disiplin ilmu pembelajaran statistik. Secara operasional, SVM bekerja dengan mencari *hyperplane* (permukaan pembatas) yang optimal untuk memisahkan data ke dalam kelompok-kelompok berbeda dengan jarak margin terbesar, pendekatan yang menjamin tingkat akurasi tinggi serta kemampuan generalisasi yang baik terhadap data baru [39].



Gambar 2. 1 Support Vector Machine [40]

Hyperplane adalah garis pemisah optimal antara dua kelas data. Proses pencarian *hyperplane* terbaik dilakukan dengan mengidentifikasi margin *hyperplane* (jarak antara *hyperplane* dan data *point* terdekat dari masing-masing kelas) dan memaksimalkan nilai *margin* tersebut. Secara matematis, fungsi kendala yang harus dipenuhi oleh setiap titik data agar berada pada sisi yang benar dari *hyperplane* adalah [39]:

$$y_i(w^T x_i + b) \geq 1 \quad (8)$$

Dimana:

x_i = vektor fitur dari data ke-i

y_i = label kelas ($+1$ atau -1)

w = vektor bobot (normal dari *hyperplane*)

b = bias (*offset* dari *hyperplane* terhadap titik asal)

Pertidaksamaan ini memastikan bahwa setiap titik data berada setidaknya pada jarak *margin* minimal 1 dari *hyperplane*. Data poin yang berada paling dekat dengan *hyperplane* disebut sebagai *support vector* [40].

Beberapa kelebihan yang ada pada SVM antara lain [41]:

1. Menangani data nonlinear melalui kernel *trick*.
2. Efektif pada ruang fitur berdimensi tinggi (*high-dimensional vector spaces*).
3. Menghasilkan *hyperplane* optimal dengan batas keputusan yang jelas.
4. *Supervised classifier* yang terbukti untuk klasifikasi dan regresi.

Beberapa kekurangan yang ada pada SVM antara lain [42]:

1. Tidak *scalable* untuk dataset besar karena kompleksitas komputasinya yang tinggi.
2. Performa optimal hanya pada *binary classification*, sedangkan pada *multiclass classification* kinerjanya menurun signifikan.

2.10 Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan suatu bidang kajian yang menitikberatkan pada kegiatan mengolah dan menganalisis opini, perilaku, serta sentimen individu terhadap berbagai isu, peristiwa, organisasi, produk, maupun layanan beserta atribut-atribut yang melekat pada objek tersebut [43]. Analisis sentimen dapat dikategorikan ke dalam sentimen positif, negatif, atau netral [44].

Terdapat tiga tahapan utama dalam analisis sentimen, yaitu [24]:

1. Pra-pemrosesan

   Pada tahap ini, data teks mentah diproses melalui pembersihan untuk mengeliminasi elemen-elemen yang tidak bermakna seperti kata-kata fungsional (*stopword*), simbol-simbol khusus, dan karakter numerik. Selanjutnya, teks yang telah dibersihkan ditransformasikan menjadi representasi fitur numerik menggunakan berbagai teknik seperti *Term Frequency–Inverse Document Frequency* (TF-IDF), *GloVe*, *fastText*, dan *word2vec*.
2. Ekstraksi Fitur

   Pada tahap ini, berfokus pada konversi data teks mentah menjadi representasi numerik. Ini adalah tahap persiapan data di mana fitur-fitur penting dari teks diidentifikasi dan diubah menjadi vektor yang dapat dipahami oleh algoritma.
3. Klasifikasi

   Pada tahap ini, teks yang telah melalui pra-pemrosesan dan ekstraksi fitur digunakan untuk memprediksi kategori sentimen. Pendekatan yang diterapkan mencakup *Machine Learning* atau *Deep Learning*.

Analisis sentimen mencakup tiga jenis klasifikasi yaitu [45]:

1. Sentimen positif

   Sentimen positif menunjukkan reaksi atau sikap yang meningkatkan nilai seseorang atau sesuatu.

2. Sentimen negatif

   Sentimen negatif menunjukkan reaksi atau sikap yang menurunkan nilai seseorang atau sesuatu. Kalimat ini sering membentuk penurunan terhadap nilai pandang yang biasanya ditandai kata negasi.

3. Sentimen netral

   Sentimen netral menunjukkan ekspresi kalimat yang tidak bersifat positif maupun negatif.

Analisis sentimen dapat dilakukan pada berbagai tingkat kedetilan yang berbeda, dengan empat level utama yang membentuk hierarki analisis [46]:

1. Level dokumen (*Document level*):

   Analisis sentimen level dokumen mengklasifikasikan seluruh dokumen dengan satu polaritas tunggal (positif, negatif, atau netral). Pendekatan ini dapat diterapkan pada unit teks yang lebih besar seperti bab atau halaman buku. Meskipun dapat menggunakan baik pendekatan pembelajaran terawasi maupun tak terawasi, level ini memiliki keterbatasan signifikan dalam menangani teks yang mengandung sentimen campuran. Tantangan utama yang dihadapi adalah masalah lintas domain dan lintas bahasa, di mana akurasi analisis bergantung pada seberapa spesifik fitur (kata-kata) yang digunakan sesuai dengan domain target.

2. Level kalimat (*Sentence level*):

   Pada level kalimat, setiap kalimat dianalisis secara independen untuk menentukan polaritasnya. Pendekatan ini lebih efektif untuk dokumen yang mengandung berbagai sentimen karena memungkinkan identifikasi sentimen yang beragam dalam satu dokumen. Polaritas individual setiap kalimat dapat diagregasi untuk memperoleh sentimen dokumen secara keseluruhan atau digunakan secara terpisah sesuai kebutuhan. Tingkat klasifikasi ini melibatkan aspek subjektivitas karena tidak semua kalimat mengekspresikan sentimen. Tantangan yang lebih kompleks muncul ketika menangani kalimat bersyarat dan pernyataan ambigu, yang memerlukan pemrosesan yang lebih canggih dibandingkan analisis level dokumen.

3. Level frasa (*Phrase level*):

   Level frasa melibatkan penambangan kata-kata opini pada unit yang lebih kecil dari kalimat. Setiap frasa dapat berisi satu atau lebih aspek yang diekspresikan. Pendekatan

ini khususnya berguna untuk analisis ulasan produk multi-baris, di mana opini terhadap berbagai fitur atau aspek produk diekspresikan dalam frasa-frasa terpisah. Polaritas kata-kata individual erat terkait dengan karakteristik subjektivitas teks; kata sifat, misalnya, memiliki probabilitas tinggi mengindikasikan suatu pernyataan subjektif. Pemilihan istilah dalam ekspresi juga mencerminkan atribut demografis pembicara dan memengaruhi interpretasi sentimen secara keseluruhan.

4. Level aspek (*Aspect level*):

   Analisis sentimen level aspek mengidentifikasi dan mengklasifikasikan sentimen terhadap aspek-aspek spesifik dalam kalimat. Berbeda dengan level sebelumnya yang memberikan polaritas tunggal, level ini mempertahankan polaritas untuk setiap aspek yang teridentifikasi, kemudian mengagregasikan hasil tersebut untuk sentimen keseluruhan. Pendekatan ini memungkinkan analisis yang lebih granular dan akurat, terutama berguna ketika target opini memiliki berbagai fitur atau karakteristik yang dapat dinilai secara independen.

Penggunaan analisis sentimen mencakup beberapa sektor seperti [47] [48]:

1. Bisnis

   Dalam konteks bisnis, analisis sentimen memungkinkan perusahaan memahami persepsi pelanggan terhadap produk atau layanan. Hal ini mendukung peningkatan strategi pemasaran dan pengambilan keputusan.

2. Politik

   Pada konteks politik, analisis sentimen mengidentifikasi tanggapan masyarakat terhadap kebijakan dan kandidat dalam pemilu.

3. Analisis ulasan

   Analisis sentimen digunakan untuk menganalisis ulasan film, serial TV, dan film pendek guna meningkatkan pengalaman pelanggan sebagai pengambilan keputusan cerdas berbasis data.

4. Pemantauan media sosial

   Pemantauan data media sosial memungkinkan pelacakan opini pelanggan secara *real-time* sehingga dapat menanggapi komentar negatif secara cepat dan memanfaatkan komentar positif untuk memperkuat reputasi.

2.11 Confusion Matrix

Confusion matrix merupakan tabel yang digunakan dalam *machine learning* untuk mendeskripsikan sekaligus menilai performa suatu model klasifikasi terhadap data uji [49].

Metode ini digunakan dalam konsep *data mining* untuk menghitung tingkat akurasi model. Proses evaluasi dilakukan dengan menentukan nilai akurasi (*accuracy),* presisi (*precision*), serta daya ingat atau sensitivitas (*recall*) [50]. *Confusion matrix* berperan krusial dan idealnya menjadi komponen utama yang dipertimbangkan saat menentukan *threshold* dalam sebuah model klasifikasi [51]. *Confusion matrix* untuk klasifikasi multikelas berbeda dengan *confusion matrix* pada klasifikasi biner. Bentuk *confusion matrix* untuk tiga kelas, digambarkan pada Tabel 2. 1 [52].

Tabel 2. 1 *Confusion Matrix* Tiga Kelas

Aktual / Prediksi	**A**	**B**	**C**
A	c11	c12	c13
B	c21	c22	c23
C	c31	c32	c33

Penentuan nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) dilakukan untuk setiap kelas secara terpisah, bukan berdasarkan keseluruhan kelas. Untuk kelas A, definisinya adalah sebagai berikut.

1. TP_A : data kelas A, diprediksi sebagai kelas A (c11).
2. FP_A : data selain kelas A, diprediksi sebagai kelas A (c21 + c31).
3. FN_A : data kelas A, diprediksi sebagai kelas lain (c12 + c13).
4. TN_A : data selain kelas A, diprediksi sebagai selain kelas A (c22 + c23 + c32 + c33).

Prinsip perhitungan yang sama juga diterapkan pada kelas **B** dan **C**.

Tabel 2. 2 Elemen *Confusion Matrix* Untuk Kelas A

Aktual / Prediksi	**A**	**B**	**C**
A	TP_A	FN_A	FN_A
B	FP_A	TN_A	TN_A
C	FP_A	TN_A	TN_A

Tabel 2. 3 Elemen *Confusion Matrix* Untuk Kelas B

Aktual / Prediksi	**A**	**B**	**C**
A	TN_B	FP_B	TN_B
B	FN_B	TP_B	FN_B
C	TN_B	FP_B	TN_B

Tabel 2. 4 Elemen *Confusion Matrix* Untuk Kelas C

Aktual / Prediksi	A	B	C
A	TN_C	TN_C	FP_C
B	TN_C	TN_C	FP_C
C	FN_C	FN_C	TP_C

Dalam mengevaluasi performa model klasifikasi yang menggunakan *confusion matrix*, terdapat beberapa metrik utama yang dimanfaatkan, yaitu sebagai berikut [53] [54]:

1. *Accuracy*

   Accuracy adalah bagian (proporsi) dari prediksi yang benar dibandingkan dengan seluruh prediksi, atau seberapa sering sebuah model menghasilkan prediksi yang benar.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$Overall\ Accuracy = \frac{Jumlah\ prediksi\ benar}{Jumlah\ seluruh\ data} \tag{10}$$

2. *Precision*

   Precision bertujuan untuk menunjukkan seberapa besar bagian dari semua prediksi positif yang benar-benar tepat.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

3. *Recall*

   Recall bertujuan untuk menunjukkan seberapa besar bagian dari seluruh data yang benar-benar positif yang dapat dikenali dengan tepat oleh model sebagai positif.

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

4. *F1-score*

   F1-score adalah ukuran yang menyatukan *precision* dan *recall* dalam satu nilai, digunakan ketika keduanya sama penting. Ukuran ini berupa rata-rata harmonik yang mempertimbangkan dua jenis kesalahan, yaitu prediksi positif yang salah (*false positive*) dan kasus positif yang gagal terdeteksi (*false negative*).

$$\mathrm{F1} - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

5. *Weighted average*

Weighted average merupakan metode untuk menghitung nilai rata-rata dari berbagai metrik evaluasi, seperti *accuracy*, *precision*, *recall*, dan *f1-score*, dengan mempertimbangkan jumlah sampel pada setiap kelas [55]. Perhitungan *weighted average* untuk *precision*, *recall*, dan *f1-score* dapat dilakukan menggunakan persamaan berikut [52]:

$$Precision_{weighted} = \frac{N_A \times (Precision_A) + N_B \times (Precision_B) + N_C \times (Precision_C)}{N} \quad (14)$$

$$Recall_{weighted} = \frac{N_A \times (Recall_A) + N_B \times (Recall) + N_C \times (Recall_C)}{N} \quad (15)$$

$$F1_{weighted} = 2 \times \frac{Precision_{weighted} \times Recall_{weighted}}{Precision_{weighted} + Recall_{weighted}} \quad (16)$$

Keterangan:

A, B, C = kategori atau kelas

N = jumlah sampel

N_A, N_B, N_C = jumlah sampel per kelas

2.12 Metode Pengembangan Waterfall

Metode pengembangan *waterfall* merupakan salah satu model klasik dalam rekayasa perangkat lunak yang diperkenalkan oleh Winston W. Royce pada tahun 1970. Setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya agar tidak ada kemungkinan untuk kembali ke tahap sebelumnya. Model ini cocok digunakan pada proyek dengan kebutuhan yang jelas, stabil, dan terdokumentasi dengan baik [56].

Tahapan dalam metode *waterfall* terdiri dari [57]:

1. Analisis

   Tahap analisis dilakukan untuk mengidentifikasi kebutuhan pengguna dan perangkat lunak yang dibutuhkan guna mendukung proses pengembangan sistem.

2. Perancangan (desain)

   Tahapan ini dilakukan sebelum proses pengkodean dimulai. Tujuan dari tahap ini adalah untuk memberikan gambaran mengenai sistem yang akan dikembangkan beserta rancangan tampilannya. Pada tahap ini, seluruh kebutuhan pengguna yang telah dianalisis

sebelumnya diterapkan ke dalam bentuk desain sistem, seperti perancangan antarmuka, serta membantu menentukan arsitektur sistem secara menyeluruh.

3. Pengembangan sistem (pembuatan kode program)

   Pada tahap ini dilakukan proses implementasi sistem melalui kegiatan pengkodean. Penulisan kode program merupakan proses penerjemahan desain sistem yang telah dirancang sebelumnya ke dalam bentuk instruksi yang dapat dipahami oleh komputer menggunakan bahasa pemrograman. Tahapan ini menjadi proses utama dalam merealisasikan sistem yang telah dirancang agar dapat dijalankan sesuai fungsinya.

4. Pengujian

   Tahap pengujian (*testing*) dilakukan dengan berfokus pada aspek logika dan fungsional perangkat lunak untuk memastikan bahwa seluruh bagian sistem telah berjalan dan diuji dengan baik.

Beberapa kelebihan metode *waterfall* antara lain [58]:

1. Sistem yang dihasilkan memiliki kualitas yang baik karena proses pengembangannya dilakukan secara bertahap dan terstruktur.
2. Tahapan pengembangan dilakukan secara berurutan (*one by one*), sehingga dapat mengurangi kemungkinan terjadinya kesalahan selama proses pengembangan sistem.
3. Dokumentasi pengembangan sistem tersusun dengan rapi karena setiap tahapan harus diselesaikan secara lengkap sebelum melanjutkan ke tahap berikutnya.

Beberapa kekurangan metode *waterfall* antara lain [58]:

1. Proses pengembangan sistem membutuhkan waktu yang relatif lama serta biaya yang cukup besar.
2. Diperlukan pengelolaan yang baik karena proses pengembangan tidak dapat dilakukan secara berulang sebelum produk selesai dikembangkan.
3. Kesalahan kecil yang tidak terdeteksi sejak awal dapat berkembang menjadi masalah yang lebih besar dan memengaruhi tahapan berikutnya.
4. Dalam penerapannya, proses pengembangan jarang berjalan sepenuhnya sesuai urutan sekuensial seperti pada teori karena sering terjadi iterasi yang dapat menimbulkan permasalahan baru.

2.13 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan teknik *oversampling* data yang banyak digunakan dalam *machine learning* untuk mengatasi permasalahan ketidakseimbangan kelas (*imbalanced data*). Metode ini bekerja dengan

menghasilkan data sintetis baru yang memiliki karakteristik serupa dengan data pada kelas minoritas berdasarkan sampel yang telah ada sebelumnya. Dengan demikian, distribusi data menjadi lebih seimbang sehingga model dapat mempelajari karakteristik kelas minoritas dengan lebih baik. Dalam analisis sentimen, kelas minoritas merupakan label sentimen yang memiliki jumlah data paling sedikit dibandingkan dengan kelas lainnya [59].

Proses pembentukan data sintetis pada metode SMOTE dilakukan melalui beberapa tahapan sebagai berikut [60]:

1. Memilih secara acak satu sampel dari kelas minoritas sebagai sampel awal (x_0).
2. Menentukan sebanyak K tetangga terdekat (*K-nearest neighbors*) dari sampel tersebut yang masih berada pada kelas minoritas.
3. Memilih secara acak salah satu dari K tetangga terdekat sebagai sampel pembanding (x_k).
4. Membentuk data sintetis baru (z) dengan melakukan interpolasi linier antara sampel awal (x_0) dan sampel tetangga (x_k) menggunakan persamaan berikut.

$$z = x_0 + w(x_k - x_0) \tag{17}$$

Dimana:

z = data sintetis yang dihasilkan

x_0 = sampel dari kelas minoritas

x_k = salah satu tetangga terdekat dari x_0

w = bilangan acak yang bernilai antara 0 dan 1

Proses tersebut dilakukan secara berulang hingga jumlah data sintetis yang diinginkan berhasil dibangkitkan. Data sintetis yang dihasilkan berada di antara sampel kelas minoritas dengan salah satu tetangga terdekatnya sehingga tetap mempertahankan karakteristik data asli [60].

Dalam implementasinya, proses SMOTE dapat dilakukan secara otomatis menggunakan *library* Python, salah satunya yaitu *imbalanced-learn* (*imblearn*), yang menyediakan berbagai fungsi untuk penanganan ketidakseimbangan data dan dapat diakses melalui *imbalanced-learn* GitHub *Repository* [61].

2.14 Dewan Perwakilan Rakyat (DPR)

Dewan Perwakilan Rakyat (DPR) merupakan salah satu pilar utama sistem pemerintahan Indonesia, mengemban tanggung jawab representasi rakyat melalui

pelaksanaan tiga fungsi pokok. Pertama adalah fungsi legislasi, kedua adalah fungsi anggaran, dan ketiga adalah fungsi pengawasan. Ketiga fungsi ini bersifat integratif dan saling melengkapi dengan tujuan utama menjaga stabilitas kekuasaan, mewujudkan transparansi dalam proses pemerintahan, dan membangun sistem yang bertanggung jawab kepada masyarakat [62]. Pasca amandemen Undang-Undang Dasar Tahun 1945, kedudukan DPR mengalami transformasi signifikan. Sebelumnya, DPR berada dalam posisi subordinat terhadap kekuasaan eksekutif yang didominasi Presiden dalam proses pembentukan undang-undang. Namun, melalui empat tahap amandemen konstitusi (1999–2002), terjadi pergeseran fundamental yang menempatkan DPR sebagai lembaga legislatif sejajar dengan Presiden, dilengkapi dengan kewenangan untuk mengajukan rancangan undang-undang, menyetujui anggaran negara, dan menjalankan fungsi pengawasan terhadap pemerintah [63]. DPR memiliki tanggung jawab representatif yang krusial, pelaksanaannya dihadapkan pada tantangan nyata yang memengaruhi hubungan institusi dengan publik. Tiga masalah utama antara lain [62]:

1. Anggota DPR mengalami tekanan untuk mengutamakan agenda partai atau kepentingan pribadi dibanding aspirasi konstituen.
2. Komunikasi antara DPR dan masyarakat belum efektif, mengakibatkan penurunan akuntabilitas dan responsivitas.
3. Transparansi dalam proses legislasi masih terbatas, menghambat partisipasi publik yang seharusnya menjadi fondasi demokrasi. Kondisi ini menciptakan persepsi negatif publik terhadap kinerja DPR.

2.15 Aplikasi TikTok

Selama periode *lockdown* COVID-19, pertumbuhan pengguna platform media sosial mengalami peningkatan signifikan sebagai dampak dari transformasi media sosial terhadap lanskap komunikasi dan diseminasi informasi global. Evolusi ini telah mengakibatkan perubahan fundamental dalam cara berita diciptakan dan didistribusikan, sekaligus meningkatkan kualitas interaksi antar pengguna. Salah satu aplikasi yang mengalami peningkatan pengguna secara signifikan adalah TikTok [64]. TikTok adalah aplikasi jaringan sosial *mobile* yang diakui secara internasional yang memberikan izin kepada pengguna untuk memposting video pendek [65].

Aplikasi ini telah berkembang pesat di pasar digital Indonesia dengan 157,6 juta pengguna aktif (73,5% dari pengguna internet), menjadikannya platform komunikasi utama antara merek dan konsumen. Konten video pendeknya memungkinkan penyebaran informasi

yang cepat dengan engagement tinggi. Secara global, media sosial telah menjangkau 5,04 miliar pengguna (62,3% populasi dunia) [66]. Konten di TikTok terus relevan dengan isu-isu terkini dan memainkan peran penting dalam memengaruhi persepsi dan tindakan publik, khususnya di kalangan anak muda, karena status platform sebagai media berbagi video singkat yang paling populer saat ini [67].