BAB II
TINJAUAN PUSTAKA

2.1 Sentiment Analysis dan Opinion Mining

Sentiment analysis atau opinion mining adalah cabang Natural Language Processing (NLP) yang berfokus pada identifikasi, ekstraksi, dan kuantifikasi opini, sentimen, serta sikap subjektif yang terkandung dalam teks (Liu, 2012). Dalam penelitian ini, sentiment analysis diterapkan pada komentar YouTube berbahasa Indonesia untuk mengklasifikasikan opini komunitas kreator terhadap kebijakan monetisasi YPP ke dalam tiga polaritas: Positif, Negatif, dan Netral.

Secara konseptual, sentiment analysis beroperasi pada tiga tingkatan granularitas. Pertama, document-level mengklasifikasikan keseluruhan dokumen ke dalam satu polaritas sentimen. Kedua, sentence-level mengklasifikasikan setiap kalimat secara terpisah. Ketiga, aspect-level atau ABSA mengidentifikasi sentimen terhadap aspek-aspek spesifik dalam teks tingkatan yang paling informatif dan digunakan dalam penelitian ini. Perkembangan metodologi sentiment analysis terbagi dalam tiga generasi: (1) pendekatan berbasis leksikon; (2) machine learning konvensional seperti SVM dengan representasi Bag-of-Words (CountVectorizer) sebagai baseline; dan (3) deep learning berbasis representasi kontekstual seperti IndoBERT yang menjadi model utama penelitian ini Faisal et al. (2025).

2.2 Tokenisasi dalam Natural Language Processing

Tokenisasi (tokenization) adalah proses memecah teks mentah menjadi unit-unit terkecil yang disebut token (Jurafsky & Martin, 2026). Dalam konteks analisis teks berbahasa Indonesia, token umumnya setara dengan satu kata. Sebagai contoh, kalimat "monetisasi channel saya susah" setelah ditokenisasi menghasilkan empat token: ["monetisasi", "channel", "saya", "susah"].

Tokenisasi merupakan tahap fundamental dalam pipeline NLP karena seluruh proses berikutnya mulai dari penghapusan stopword, pembobotan representasi vektor kata (CountVectorizer) pada SVM, hingga encoding input pada

Dipindai dengan CamScanner

IndoBERT beroperasi pada unit token, bukan pada teks mentah. Tanpa tokenisasi, model tidak dapat memproses teks secara terstruktur Manning et al. (2009).

Dalam penelitian ini, tokenisasi dilakukan menggunakan metode word tokenization berbasis spasi dan tanda baca setelah tahap normalisasi selesai. Selanjutnya dilakukan stopword removal menggunakan library NLTK untuk menghapus token-token yang tidak bermakna secara semantik dalam analisis sentimen, seperti "yang", "di", "ke", dan "saya", mengikuti alur preprocessing tokenisasi dan penghapusan stopword yang umum digunakan pada penelitian sentiment analysis berbasis NLTK Faisal et al. (2025). Komentar yang seluruh isinya menjadi kosong setelah tahap stopword removal (misalnya karena seluruh kata dalam komentar merupakan stopword) dibuang menggunakan dropna(), sehingga tidak diikutsertakan dalam analisis.

2.3 Aspect-Based Sentiment Analysis (ABSA)

Aspect-Based Sentiment Analysis (ABSA) merupakan pendekatan fine-grained dalam sentiment analysis yang mengidentifikasi aspek-aspek spesifik dalam teks dan mengevaluasi sentimen yang terkait dengan setiap aspek tersebut Do et al. (2019);Zhang et al. (2022). Keunggulan ABSA dibandingkan analisis sentimen tradisional dapat diilustrasikan dengan contoh komentar: "Syarat 4.000 jam tayang terlalu berat, tapi konten abang sangat bermanfaat banget." Analisis sentimen biasa hanya menghasilkan satu label ambigu, sedangkan ABSA mampu mengidentifikasi sentimen Negatif terhadap aspek Syarat & Kebijakan Monetisasi dan sentimen Positif terhadap aspek Apresiasi Konten Edukasi secara bersamaan dalam satu komentar.

Dalam penelitian ini, ABSA dilakukan dengan fokus pada Aspect Sentiment Classification (ASC) menggunakan enam kategori aspek yang diidentifikasi berdasarkan analisis konten aktual dari dataset: (1) Dukungan Komunitas & Semangat; (2) Syarat & Kebijakan Monetisasi; (3) Apresiasi Konten Edukasi; (4) Channel & Performa Konten; (5) Engagement Penonton; dan (6) Bantuan Teknis & Akun.

Dipindai dengan CamScanner

2.4 BERT: Bidirectional Encoder Representations from Transformers

BERT (Bidirectional Encoder Representations from Transformers) adalah model representasi bahasa berbasis pra-latih yang diperkenalkan oleh Devlin et al. (2019). BERT menggunakan arsitektur Transformer yang sepenuhnya berbasis mekanisme self-attention, memungkinkan model menangkap konteks kalimat secara bidireksional setiap kata direpresentasikan berdasarkan konteks dari kata-kata di kiri dan kanan secara bersamaan. Kemampuan ini menjadikan BERT secara fundamental lebih ekspresif dibandingkan model sekuensial seperti LSTM, khususnya dalam menangani kalimat informal yang kaya konteks implisit seperti komentar YouTube.

BERT menggunakan dua mekanisme pra-latih: (1) Masked Language Modeling (MLM) 15% token dimask dan model dilatih untuk memprediksinya berdasarkan konteks sekitar; dan (2) Next Sentence Prediction (NSP) model dilatih menentukan apakah dua kalimat berurutan dalam konteks aslinya. Setelah pra-latih, BERT dapat di-fine-tune untuk tugas klasifikasi sentimen dengan menambahkan lapisan klasifikasi di atas token [CLS].

2.4.1 Mekanisme Self-Attention

Inti dari arsitektur BERT adalah mekanisme Scaled Dot-Product Attention yang menghitung bobot perhatian antar token dalam satu kalimat secara bersamaan. Rumus self-attention adalah sebagai berikut Devlin et al. (2019):

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) \times V \quad (1)$$

Keterangan :

1. Q (Query): Representasi token yang sedang 'memperhatikan' token lain.
2. K (Key): Representasi token yang 'diperhatikan' oleh Query.
3. V (Value): Informasi yang diteruskan berdasarkan bobot perhatian.
4. d_k: Dimensi vektor Key; digunakan sebagai faktor skala untuk mencegah nilai dot product terlalu besar sehingga gradien softmax tidak menjadi terlalu kecil.
5. softmax: Fungsi normalisasi agar seluruh bobot perhatian berjumlah 1.

Pembagi $\sqrt{d_k}$ sangat penting agar fungsi softmax tidak berada di wilayah gradien yang sangat kecil (vanishing gradient) ketika dimensi vektor besar. Dengan

Dipindai dengan CamScanner

mekanisme ini, setiap token dapat 'memperhatikan' semua token lain dalam satu kalimat secara bidireksional dan bersamaan.

Contoh kontekstual dari data penelitian: pada kalimat "channel saya sepi tapi saya tetap semangat", token "sepi" akan memiliki bobot perhatian tinggi terhadap "channel" (sinyal negatif), sementara token "semangat" memiliki bobot tinggi terhadap "tetap" (sinyal positif). Kedua sinyal berlawanan ini ditangkap sekaligus oleh BERT dalam satu proses forward pass.

2.4.2 Fungsi Softmax untuk Klasifikasi Sentimen

Setelah proses fine-tuning, lapisan klasifikasi yang ditambahkan di atas representasi token (CLS) IndoBERT menggunakan fungsi Softmax untuk mengonversi skor logit menjadi probabilitas per kelas sentimen Devlin et al. (2019):

$$P(y = k \mid x) = e^{(z_k)} / \Sigma_j e^{(z_j)} \quad (2)$$

Keterangan :

1. $P(y = k \mid x)$: Probabilitas bahwa input x termasuk kelas sentimen ke-k.
2. z_k: Skor logit (nilai mentah sebelum normalisasi) untuk kelas ke-k.
3. K: Jumlah kelas sentimen; dalam penelitian ini K = 3 (Positif, Netral, Negatif).
4. $\Sigma_j e^{(z_j)}$: Jumlah eksponensial seluruh kelas sebagai faktor normalisasi.

Kelas dengan nilai $P(y = k \mid x)$ tertinggi dipilih sebagai prediksi sentimen akhir. Misalnya jika P(Positif) = 0,78; P(Netral) = 0,15; P(Negatif) = 0,07, maka model memprediksi komentar tersebut bersentimen Positif.

2.5 IndoBERT: BERT untuk Bahasa Indonesia

IndoBERT adalah varian BERT yang dikembangkan khusus untuk Bahasa Indonesia oleh (Wilie, et al., 2020) sebagai bagian dari proyek IndoNLU. IndoBERT dilatih dari awal menggunakan korpus Indo4B yang terdiri dari sekitar empat miliar kata dari berbagai sumber publik berbahasa Indonesia, termasuk Wikipedia Bahasa Indonesia, berita daring, dan forum diskusi. Dengan pelatihan pada korpus monolingual yang masif ini, IndoBERT mampu menangkap karakteristik linguistik Bahasa Indonesia jauh lebih baik dibandingkan multilingual

Dipindai dengan CamScanner

BERT, termasuk dalam menangani slang, singkatan, dan bentuk informal yang sangat dominan dalam komentar YouTube.

Efektivitas IndoBERT untuk teks berbahasa Indonesia telah dikonfirmasi oleh sejumlah studi: (Firdaus & Trisnawarman, 2025)membuktikan IndoBERT Lite efektif pada komentar YouTube berbahasa Indonesia; (Sumartha, 2025) mengkonfirmasi efektivitasnya untuk analisis sentimen kebijakan pemerintah; dan (Anugrah & Agatha, 2025) menunjukkan IndoBERT mampu mengekstrak opini publik dari postingan media sosial berbahasa Indonesia.

2.6 Support Vector Machine (SVM) sebagai Model Baseline

Support Vector Machine (SVM) adalah algoritma machine learning yang mencari hyperplane pemisah optimal antara kelas-kelas dalam ruang fitur berdimensi tinggi (Vapnik, 1995). SVM dipilih sebagai model baseline dalam penelitian ini karena dua alasan utama. Pertama, SVM merupakan algoritma machine learning konvensional terbaik untuk klasifikasi teks sebelum era deep learning dan diakui sebagai tolok ukur representatif dalam literatur NLP. Kedua, Azzouzy et al. (2024)mengkonfirmasi bahwa SVM merupakan model konvensional dengan performa paling kompetitif untuk klasifikasi komentar YouTube dibandingkan algoritma lainnya.

Dalam implementasinya, SVM menggunakan representasi CountVectorizer, yaitu representasi Bag-of-Words berbasis frekuensi kemunculan mentah (raw count) setiap kata, tanpa pembobotan tambahan. Perbandingan langsung antara SVM (representasi Bag-of-Words statis) dan IndoBERT (representasi kontekstual) mencerminkan perbandingan antara generasi kedua dan ketiga metodologi NLP.

2.6.1 Fungsi Keputusan SVM

SVM bekerja dengan mencari hyperplane optimal yang memisahkan kelas – kelas dalam ruang fitur. Fungsi keputusan SVM dinyatakan sebagai berikut (Vapnik, 1995):

$$f(x) = w^T x + b \quad (3)$$

Keterangan :

Dipindai dengan CamScanner

1. f(x): Nilai keputusan yang menentukan kelas prediksi.
2. w: Vektor bobot (vektor normal terhadap hyperplane pemisah).
3. x: Vektor fitur input (representasi CountVectorizer dari komentar).
4. b: Bias (intercept); menggeser hyperplane dari titik asal.

Prediksi kelas ditentukan oleh tanda nilai f(x): jika $f(x) > 0$ maka diprediksi kelas tertentu, jika $f(x) < 0$ maka diprediksi kelas lainnya. SVM memaksimalkan margin jarak antara hyperplane dan titik data terdekat dari setiap kelas (support vectors) sehingga menghasilkan batas keputusan yang paling robust terhadap data baru.

2.6.2 Kernel Linear

Representasi Bag-of-Words (CountVectorizer) menghasilkan vektor fitur berdimensi sangat tinggi (satu dimensi untuk setiap kata unik dalam kosakata). Data berdimensi tinggi semacam ini secara empiris cenderung sudah dapat dipisahkan secara linear pada ruang fiturnya, sehingga penelitian ini menggunakan kernel Linear, yaitu kernel yang tidak melakukan pemetaan non-linear terhadap data, melainkan langsung menghitung hasil kali titik (dot product) antar vektor fitur pada ruang aslinya (Vapnik, 1995):

$$K(x_i, x_j) = x_i^T x_j \quad (4)$$

Keterangan :

1. $K(x_i, x_j)$: Nilai kemiripan (similarity) antara dua vektor fitur x_i dan x_j, dihitung sebagai hasil kali titik keduanya.
2. $x_i^T x_j$: Perkalian titik (dot product) antara vektor fitur x_i dan x_j pada ruang fitur asli, tanpa transformasi ke dimensi yang lebih tinggi.

Dengan kernel Linear, fungsi keputusan SVM pada 2.6.1 dihitung langsung dari kombinasi linear fitur input tanpa memerlukan parameter tambahan seperti gamma yang dibutuhkan kernel non-linear (misalnya RBF). Pemilihan kernel Linear ini konsisten dengan karakteristik data teks berdimensi tinggi (jumlah fitur kata jauh lebih besar dari jumlah sampel) yang secara empiris umumnya sudah cukup terpisah secara linear, sehingga kernel non-linear tidak selalu memberikan peningkatan performa yang signifikan namun menambah kompleksitas komputasi (Joachims, 1998).

Dipindai dengan CamScanner

Parameter C = 1.0 digunakan untuk mengontrol keseimbangan antara memaksimalkan margin dan meminimalkan kesalahan klasifikasi pada data training. Nilai C yang lebih besar membuat model lebih berusaha mengklasifikasikan seluruh data training dengan benar (risiko overfitting), sedangkan nilai C lebih kecil menghasilkan margin lebih lebar namun toleran terhadap beberapa kesalahan.

2.6.3 CountVectorizer (Representasi Bag-of-Words)

Dalam implementasinya, SVM pada penelitian ini menggunakan representasi CountVectorizer, yaitu metode Bag-of-Words yang mengubah teks komentar menjadi vektor numerik berdasarkan frekuensi kemunculan mentah (raw count) setiap kata, tanpa pembobotan tambahan seperti Inverse Document Frequency Manning et al. (2009). Setiap dokumen (komentar) direpresentasikan sebagai vektor yang panjangnya sama dengan ukuran kosakata (vocabulary) hasil pembelajaran dari data training, dengan nilai setiap elemen vektor:

$$CV(t, d) = \text{jumlah kemunculan kata t dalam dokumen d} \quad (5)$$

Keterangan :

1. CV(t, d): Nilai representasi kata t pada dokumen d, berupa jumlah kemunculan (count) kata tersebut dalam dokumen.
2. t: Kata (term) tertentu dalam kosakata hasil pembelajaran dari data training.
3. d: Dokumen (komentar) yang direpresentasikan.

Berbeda dengan TF-IDF yang memberikan bobot lebih rendah pada kata yang sering muncul di banyak dokumen melalui komponen Inverse Document Frequency, CountVectorizer hanya menghitung frekuensi kemunculan kata tanpa mempertimbangkan keunikannya di seluruh koleksi dokumen. Pendekatan ini lebih sederhana secara komputasi namun tetap mampu menangkap kata-kata yang paling sering muncul dan relevan dengan topik tertentu.

Contoh dari data penelitian: kata "monetisasi" yang muncul berulang kali pada komentar-komentar bertema syarat YPP akan memiliki nilai CountVectorizer yang tinggi pada komentar-komentar tersebut, sehingga tetap menjadi fitur yang cukup diskriminatif bagi SVM meskipun tanpa pembobotan IDF.

Dipindai dengan CamScanner

2.7 YouTube Partner Program dan Kebijakan Monetisasi

YouTube Partner Program (YPP) adalah program resmi YouTube yang memungkinkan kreator konten menghasilkan pendapatan melalui iklan (AdSense), Channel Membership, Super Chat, dan Super Thanks. Persyaratan utama YPP adalah 1.000 subscriber dan 4.000 jam watch time dalam 12 bulan terakhir untuk jalur utama, atau 1.000 subscriber dan 10 juta penayangan Shorts dalam 90 hari untuk jalur Shorts (Google LLC, 2025). Kebijakan ini berdampak langsung pada ratusan ribu kreator Indonesia yang sedang berjuang memenuhi persyaratan tersebut.

Setiap perubahan kebijakan YPP memicu gelombang respons di kolom komentar video YouTube. Beragam aspek kebijakan mulai dari persyaratan kelayakan, mekanisme monetisasi Shorts, hingga pembatasan konten berbantu AI menjadi topik diskusi aktif komunitas kreator Indonesia. Pemahaman mendalam terhadap dimensi-dimensi opini ini memerlukan pendekatan ABSA yang mampu mengurai sentimen secara per-aspek (Firdaus & Trisnawarman, 2025) (Sumartha, 2025).

2.8 Metrik Evaluasi Model Klasifikasi

Evaluasi performa model klasifikasi dalam penelitian ini menggunakan empat metrik standar yang umum digunakan dalam tugas klasifikasi teks multiclass, yaitu akurasi, presisi, recall, dan F1-Score. Seluruh metrik dihitung berdasarkan empat nilai dasar dari confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) Manning et al. (2009) (Jurafsky & Martin, 2026). Karena penelitian ini merupakan klasifikasi multiclass (tiga kelas: Positif, Netral, dan Negatif), keempat nilai dasar ini dihitung menggunakan pendekatan one-vs-rest, yaitu setiap kelas secara bergantian diperlakukan sebagai kelas acuan ('positif') terhadap gabungan dua kelas lainnya ('negatif/rest). Tabel 2.1 berikut menyajikan definisi tersebut dengan kelas Negatif sebagai contoh ilustrasi; definisi yang sama berlaku secara analog untuk kelas Positif dan Netral.

Dipindai dengan CamScanner

Tabel 2. 1 Nilai Dasar Confusion Matrix (Contoh ilustrasi untuk kelas negatif)

Simbol	Kepanjangan	Arti dalam Penelitian Ini
TP	True Positive	Komentar Negatif diprediksi sebagai Negatif (prediksi benar)
TN	True Negative	Komentar bukan Negatif diprediksi sebagai bukan Negatif (prediksi benar)
FP	False Positive	Komentar bukan Negatif diprediksi sebagai Negatif (prediksi salah)
FN	False Negative	Komentar Negatif diprediksi sebagai bukan Negatif (prediksi salah)

2.8.1 Akurasi (Accuracy)

Akurasi mengukur proporsi prediksi benar dari seluruh data testing dan digunakan sebagai metrik perbandingan utama antara model SVM dan IndoBERT dalam penelitian ini:

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN) \quad (6)$$

Akurasi mudah diinterpretasikan, namun dapat menyesatkan ketika distribusi kelas tidak seimbang. Oleh karena itu dalam penelitian ini akurasi digunakan bersama dengan metrik F1-Score weighted average.

2.8.2 Presisi (Precision)

Presisi mengukur dari semua prediksi positif yang dibuat model, berapa proporsi yang benar-benar positif. Presisi tinggi berarti model jarang salah memberi label positif pada data yang sebenarnya negatif (Jurafsky & Martin, 2026):

$$\text{Presisi} = TP / (TP + FP) \quad (7)$$

Dalam konteks penelitian ini, presisi tinggi pada kelas Negatif berarti ketika model memprediksi sebuah komentar bersentimen negatif, prediksi tersebut hampir selalu benar.

Dipindai dengan CamScanner

2.8.3 Recall

Recall mengukur dari semua data positif yang sebenarnya (ground truth), berapa proporsi yang berhasil terdeteksi oleh model. Recall tinggi berarti model tidak banyak melewatkan data positif yang seharusnya terdeteksi Manning et al. (2009):

$$\text{Recall} = TP / (TP + FN) \quad (8)$$

Dalam konteks penelitian ini, recall tinggi pada kelas Negatif berarti model berhasil mendeteksi sebagian besar komentar yang benar-benar negatif tanpa banyak melewatkannya.

2.8.4 F1-Score

F1-Score adalah rata-rata harmonik dari Presisi dan Recall. Metrik ini sangat berguna ketika distribusi kelas tidak seimbang seperti dalam dataset penelitian ini di mana kelas Negatif lebih sedikit dibandingkan Positif karena F1-Score memberikan gambaran performa yang lebih seimbang dibandingkan akurasi saja Manning et al. (2009):

$$\text{F1-Score} = 2 \times (\text{Presisi} \times \text{Recall}) / (\text{Presisi} + \text{Recall}) \quad (9)$$

Dalam penelitian ini digunakan F1-Score Weighted Average, di mana F1-Score setiap kelas dibobot berdasarkan jumlah sampel kelas tersebut, sehingga kelas dengan jumlah data lebih besar memberikan kontribusi lebih pada nilai akhir:

$$\text{F1-Weighted} = \Sigma_k (n_k / N) \times F1_k \quad (10)$$

Keterangan :

1. n_k: Jumlah sampel pada kelas ke-k dalam data testing.
2. N: Total jumlah seluruh sampel data testing.
3. $F1_k$: Nilai F1-Score untuk kelas ke-k.
4. Σ_k: Penjumlahan atas seluruh K kelas (Positif, Netral, Negatif).

Hubungan antar keempat metrik dapat dirangkum sebagai berikut: Akurasi memberi gambaran umum performa; Presisi dan Recall memberi gambaran per kelas dari dua sudut berbeda; F1-Score Weighted Average menyatukan keduanya dalam satu nilai tunggal yang mempertimbangkan ketidakseimbangan kelas (Jurafsky & Martin, 2026).

Dipindai dengan CamScanner