PERBANDINGAN KLASIFIKASI NAIVE BAYES DAN SUPPORT VECTOR MACHINE DALAM ANALISIS SENTIMEN DENGAN MULTICLASS DI TWITTER

Roland Vincent, Iqbal Maulana, Oman Komarudin
Program Studi Informatika S1, Fakultas Ilmu Komputer
Universitas Singaperbangsa Karawang
roland.vincent19131@student.unsika.ac.id

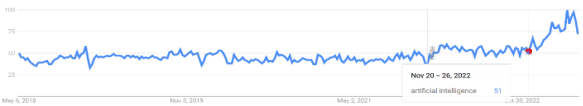
ABSTRAK

Analisis sentimen merupakan salah satu topik yang populer dalam bidang *Natural Language Processing*. Salah satu metode yang sering digunakan untuk melakukan analisis sentimen adalah *Naive Bayes* dan *Support Vector Machine*. Namun, akurasi dari metode ini masih bisa dioptimalkan dengan menggunakan teknik-teknik tertentu. Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi sentimen dengan menggunakan ekstraksi fitur TF-IDF dan *Count Vectorizer* serta seleksi fitur *Information Gain*. Selain itu, penelitian ini juga menggunakan N-gram sebagai salah satu cara untuk mengekstraksi fitur yang lebih informatif. *Dataset* yang digunakan dalam penelitian ini adalah opini pengguna Twitter mengenai perkembangan *Artificial Intelligence* yang dikategorikan menjadi tiga kelas, yaitu positif, negatif, dan netral. Dari eksperimen yang dilakukan, penggunaan *Information Gain* dapat meningkatkan akurasi percobaan secara signifikan. Dengan menggunakan *Information Gain*, akurasi *Naive Bayes* meningkat dari 59.86% menjadi 72.02% dengan menggunakan *Count Vectorizer* dan N-gram, sedangkan akurasi SVM meningkat dari 61.47% (TF-IDF) menjadi 62.38% (*Count Vectorizer* dengan N-gram). Eksperimen juga menunjukkan *Naive Bayes* lebih cocok dengan *Count Vectorizer* dan SVM lebih cocok dengan TF-IDF. Selain itu, eksperimen juga menunjukkan rasio 80:20 merupakan rasio data latih dan uji terbaik.

Kata kunci : *Artificial Intelligence, Count Vectorizer, Information Gain, Naive Bayes, Support Vector Machine, TF-IDF*

1. PENDAHULUAN

Perkembangan *Artificial Intelligence* semakin berkembang dari masa ke masa. Perkembangan teknologi ini tidak selalu disambut positif oleh masyarakat. Banyak pengguna media sosial khususnya Twitter khawatir dengan perkembangan teknologi ini. Perkembangan *Artificial Intelligence* berkembang sangat pesat saat kemunculan GPT-3.5 yang rilis pada November 2022 berdasarkan data dari Google Trends seperti yang disajikan pada Gambar 1. Perkembangan *Artificial Intelligence* meningkat pesat sehingga membuat banyak masyarakat yang takut bahwa pekerjaannya akan direbut oleh AI, seperti *programmer*, pekerjaan dibidang media, desain grafis, dan guru [1]. Banyak juga pengguna yang khawatir mengenai dampak buruk dari *Artificial Intelligence* seperti teknologi *deepfake*, pelanggaran privasi, dan lainnya [2]. Oleh sebab itu, diperlukan analisis untuk mengolah opini-opini masyarakat untuk mengetahui pandangan masyarakat mengenai perkembangan *Artificial Intelligence* di media sosial Twitter ke dalam sentimen positif, negatif, dan netral.



Gambar 1. Grafik peningkatan pencarian mengenai *Artificial Intelligence* di Google Trends di dunia selama rentang waktu 5 tahun terakhir.

Dalam melakukan analisis sentimen, diperlukan langkah dan metode yang tepat guna mencapai akurasi yang lebih baik. Penelitian mengenai analisis sentimen sudah pernah dilakukan sebelumnya menggunakan klasifikasi *Naive Bayes* dan *Support Vector Machine*. Peneliti sebelumnya oleh Isnanda [3] yang menggunakan klasifikasi *Naive Bayes* dan seleksi fitur *Information Gain* untuk menganalisis penggunaan E-Wallet saat pandemi Covid-19. Eksperimen tersebut menggunakan dua kelas positif dan negatif. Hasil akhir penelitian ini menunjukkan seleksi fitur *Information Gain* meningkatkan akurasi *Naive bayes* menjadi 92%, presisi 92% dan recall 100% yang sebelumnya tanpa menggunakan *Information Gain* hanya mendapat akurasi 84%, presisi 91% dan recall 91%. Penelitian lainnya yang dilakukan oleh Saraswita [4] melakukan eksperimen yang sama mengenai penggunaan *E-wallet* di Twitter tetapi menggunakan *Support Vector Machine* dan *Recursive Feature Elimination*. Eksperimen ini menggunakan data dengan 3 kelas (positif, negatif, dan netral) dan mendapatkan akurasi 74%, tetapi setelah menggunakan seleksi fitur *Recursive Feature Elimination* akurasi meningkat menjadi 81%.

Dari penelitian yang sudah dilakukan sebelumnya, peneliti akan membandingkan metode *Naive Bayes* dan SVM untuk melakukan analisis sentimen mengenai perkembangan *Artificial Intelligence* di Twitter. Peneliti menggunakan dua ekstraksi fitur yaitu TF-IDF dan N-gram serta seleksi fitur *Information Gain*. Penelitian ini bertujuan untuk mengetahui metode dan langkah yang tepat untuk mencapai akurasi terbaik dari membandingkan

beberapa kombinasi percobaan antara metode *Naive Bayes* dan SVM.

2. TINJAUAN PUSTAKA

2.1. *Artificial Intelligence*

Artificial Intelligence (AI) adalah teknologi yang memungkinkan mesin untuk meniru kecerdasan manusia dan melakukan tugas-tugas yang biasanya membutuhkan tenaga atau kecerdasan manusia untuk melakukannya. Kecerdasan AI didapat melalui proses pembelajaran, penalaran, dan koreksi diri berdasarkan pengalaman dan data. AI juga bisa belajar sendiri berdasarkan pengalaman yang diperoleh saat digunakan oleh manusia [1].

2.2. Term Frequency — Inverse Document Frequency (TF-IDF)

Algoritma TD-IDF memiliki komputasi yang mudah, efisien dan memiliki hasil yang akurat [2]. Algoritma TF-IDF digunakan untuk memberikan bobot pada setiap kata dalam sebuah teks. Bobot ini dihitung dengan mengukur frekuensi kemunculan kata pada dokumen untuk mengekstraksi teks dan memberikan nilai pada setiap kata dalam teks tersebut. Pembobotan kata dilakukan dengan menggunakan metode *Term Frequency* (TF), yang menghitung jumlah kemunculan kata dalam dokumen, dan metode *Inverse Document Frequency* (IDF) yang mengukur tingkat kepentingan kata dalam dokumen dengan memperhitungkan frekuensi kemunculan kata di seluruh dokumen dalam korpus [3].

Hasil TF dapat diketahui berdasarkan frekuensi kemunculan kata *t* dalam dokumen *d*. Hasil IDF didapat berdasarkan hasil logaritma jumlah dokumen *N* dibagi jumlah dokumen yang memiliki kata *t*. Hasil TF-IDF (W_t) merupakan hasil perkalian antara TF dengan IDF. Perhitungan motode TF-IDF dinyatakan pada persamaan berikut.

$$TF_{t,d} = F(t,d) \quad (1)$$

$$TF_t = log\frac{N}{DF_t} \quad (2)$$

$$w_t = TF_{t,d} \times IDF_t \quad (3)$$

2.3. Count Vectorizer

Count Vectorizer merupakan sebuah ekstraksi fitur yang mengolah dokumen atau teks menjadi representasi vektor. Pembobotan kata pada *Count Vectorizer* yaitu dengan menghitung frekuensi kemunculan kata pada dokumen kemudian direpresentasikan ke dalam bentuk vektor [4]. Setiap dokumen direpresentasikan oleh sebuah vektor dengan panjang yang sesuai dengan jumlah kosakata, dan elemen-elemen dalam vektor tersebut menunjukkan jumlah kata yang terdapat dalam dokumen tersebut [5].

Contoh cara kerja dari *CountVectorizer* adalah menghitung jumlah frekuensi kemunculan kata (fitur) pada dokumen. Frekuensi kemunculan kata atau fitur dihitung pada setiap dokumen. Sebagai contoh terdapat dokumen yang disajikan pada Tabel 1 dan perhitungan algoritma *CountVectorizer* disajikan pada Tabel 2 [6].

Tabel 1. Contoh dokumen hasil *preprocessing*

Dokumen	**Isi dokumen setelah tahap *pre-processing***
1	takut kembang ai
2	teknologi ai kembang cepat
3	kembang ai cepat
4	teknologi ai cepat takut kembang ai

Tabel 2. Contoh hasil algoritma *CountVectorizer*

	kembang	**ai**	**takut**	**cepat**	**teknologi**
Dokumen 1	1	1	1	0	0
Dokumen 2	1	1	0	1	1
Dokumen 3	1	1	0	1	0
Dokumen 4	1	2	1	1	1

2.4. N-gram

N-gram merupakan sebuah metode yang bertujuan untuk memisahkan kata berdasarkan urutan kata pada suatu kalimat. Dalam pemisahan kalimat, terdapat dua model N-gram yang umum digunakan, yaitu *unigram* dan *bigram*. *Unigram* digunakan untuk memisahkan setiap kata secara individu, sedangkan *bigram* memisahkan setiap dua kata [7].

2.5. GPT (Generative Pre-trained Transformer)

GPT (*Generative Pre-trained Transformer*) adalah sebuah pemodelan bahasa yang dikembangkan oleh perusahaan bernama OpenAI untuk menjawab sebuah perintah teks. Menurut Francisco Caio Lima Paiva [8], GPT memiliki tingkat akurasi yang cukup baik dalam melakukan analisis sentimen. Sebagai contoh, ChatGPT telah mampu mencapai akurasi sebesar 77% dalam mengklasifikasikan sentimen yang telah dievaluasi oleh ahli bahasa pada lokakarya NLP SemEval tahun 2017.

2.6. Support Vector Machine

Metode *Support Vector Machine* (SVM) merupakan teknik klasifikasi kategori *supervised learning* yang cara kerjanya menganalisis pola pada *data training* dan mencari *hyperplane* untuk memisahkan tiap kelas. Hasil dari klasifikasi ini berupa model yang dapat digunakan untuk mengenali kalimat baru yang belum dilabeli sehingga hasilnya merupakan prediksi apakah sentimen tersebut positif atau negatif [9]. Berdasarkan artikel yang ditulis oleh Marius [10], pada kasus data dengan dua kelas digunakan teknik *one-vs-rest* sedangkan pada kasus data dengan tiga kelas seperti positif, negatif, dan netral menggunakan metode *multiple binary classification* yang menggunakan teknik *one-vs-one*.

Konsep matematika dalam memperoleh *hyperlane* pada SVM dirumuskan dengan persamaan 2.4 dengan *w* adalah vektor bobot, x_i adalah vektor fitur untuk data *input i* dan *b* adalah bias.

$$(w.x_i) + b = 0 \quad (2.4)$$

Untuk data dalam x_i yang termasuk pada kelas -1 menggunakan persamaan 4 dengan y_i adalah label kelas untuk data *input i*.

$$(w.x_i) + b \leq 1, y_i = -1 \quad (4)$$

Untuk data dalam x_i yang termasuk pada kelas +1 menggunakan persamaan 5 dengan y_i mewakili kelas positif

$$(w.x_i) + b \geq 1, y_i = 1 \quad (5)$$

Dalam proses klasifikasi menggunakan *Support Vector Machine* (SVM), terkadang terjadi situasi di mana *kernel* linear tidak memberikan hasil klasifikasi yang optimal, sehingga menghasilkan prediksi yang buruk terhadap data. Untuk mengatasi hal ini, kita dapat menggunakan *kernel non-linear* dengan memanfaatkan "*kernel trick*". Dengan menggunakan *kernel trick*, data *input* akan dipetakan ke ruang fitur yang memiliki dimensi yang lebih tinggi, sehingga data *input* tersebut dapat dipisahkan secara linear dan membentuk *hyperplane* yang optimal [9]. Terdapat beberapa persamaan *kernel* SVM yaitu linear, *polynomial*, RBF dan *sigmoid*.

Persamaan *kernel* linier pada SVM ditunjukkan pada persamaan 6. Fungsi *kernel* ini menghitung hasil perkalian titik antara dua vektor: x_i dan x dengan x_i merupakan vektor *input* pertama yang elemennya disusun dalam matriks kolom, dan x merupakan vektor *input* kedua yang elemennya disusun dalam matriks baris. Superskrip T menandakan operasi *transpose*.

$$K(x_i, x) = x_i^T x \quad (6)$$

Persamaan *kernel polynominal* pada SVM ditunjukkan pada persamaan 7. Fungsi *kernel polynominal* ini menghitung hasil pemangkatan dari suatu ekspresi yang melibatkan perkalian titik antara dua vektor: x_i dan x dengan γ adalah parameter yang harus bernilai positif.

$$K(x_i, x) = (\gamma . x_i^T x + r)^p, \gamma > 0 \quad (7)$$

Persamaan *kernel* RBF (*Radial Basis Function*) menggunakan persamaan *Gaussian* yang ditunjukkan pada persamaan 8. Fungsi *kernel Gauss* ini menghitung nilai kesamaan antara dua vektor x_i dan x menggunakan fungsi eksponensial.

$$K(x_i, x) = exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (8)$$

Persamaan *kernel sigmoid* dikenal sebagai *kernel* tangen hiperbolik (*tanh*) seperti yang ditunjukkan pada persamaan 9. Fungsi *kernel* tangen hiperbolik ini menghitung nilai kesamaan antara dua vektor x_i dan x menggunakan fungsi tangen hiperbolik dengan parameter r menambahkan penyesuaian linear terhadap nilai kesamaan.

$$K(x_i, x) = tanh(\gamma x_i^T x + r), \gamma > 0 \quad (9)$$

2.7. Naive Bayes

Metode *Naive Bayes* dikenal sebagai teknik klasifikasi yang memanfaatkan teori probabilitas untuk menentukan kemungkinan klasifikasi tertentu berdasarkan frekuensi setiap klasifikasi dalam data pelatihan [11]. Dengan memanfaatkan informasi dari data pelatihan, metode ini mencari probabilitas tertinggi dari klasifikasi yang mungkin terjadi. Proses klasifikasi menggunakan metode *Naive Bayes* terdiri dari dua tahapan, yakni analisis sampel dokumen dan klasifikasi. Pada fase analisis contoh dokumen, terlebih dahulu dipilih istilah-istilah yang mungkin muncul dalam kelompok dokumen contoh. Kemudian, probabilitas dihitung untuk setiap kategori berdasarkan dokumen-dokumen contoh tersebut. Pada fase klasifikasi, dilakukan penentuan kategori yang sesuai untuk suatu dokumen berdasarkan istilah yang terdapat dalam dokumen yang sedang diolah [12].

Pada dasarnya, konsep *Naive Bayes* didasarkan pada peluang bersyarat P(X|C), di mana *X* mengacu pada posterior dan *C* mengacu pada *prior*. *Prior* merujuk pada probabilitas dari suatu kejadian sebelum melihat data atau informasi baru yang relevan. *Prior* dapat didasarkan pada pengalaman sebelumnya atau pengetahuan yang tersedia sebelumnya, sedangkan *posterior* merujuk pada probabilitas dari suatu kejadian setelah mempertimbangkan data baru atau informasi yang relevan. *Posterior* diperoleh melalui perhitungan menggunakan *Teorema Bayes*, yang menggabungkan *prior* dengan data baru atau informasi yang relevan untuk mendapatkan estimasi terbaru tentang probabilitas suatu kejadian [13]. Perhitungan *Naive Bayes* menggunakan Persamaan 10.

$$p(C|X) = \frac{p(X|C)p(C)}{p(X)} \quad (10)$$

Keterangan:

a. $p(C|X)$: Probabilitas kondisional dari hipotesis C, diberikan data *X*.
b. $p(X|C)$: Probabilitas kondisional dari data *X*, diberikan hipotesis *C*.
c. $p(C)$: Probabilitas *prior* dari hipotesis *C*.
d. p(X) : Probabilitas dari data *X*.
e. *X* : Sampel data yang memiliki label yang tidak diketahui.
f. *C* : Hipotesis bahwa *X* adalah data label.

2.8. Information Gain

Seleksi fitur *Information Gain* adalah metode sederhana dan umum dipakai pada aplikasi untuk kategorisasi teks, menganalisis data mikro array, serta menganalisis data citra. Metode ini dapat mengurangi gangguan yang diakibatkan oleh fitur yang tidak berkaitan dan membantu mengidentifikasi fitur yang paling informatif berdasarkan kelas tertentu [14]. Menurut penelitian Bijaksana [13] seleksi fitur *Information Gain* memiliki tujuan untuk mengurangi ukuran dimensi serta meningkatkan akurasi klasifikasi. *Information Gain* memiliki kemampuan untuk mengalkulasikan seberapa banyak informasi yang diberikan oleh keberadaan atau ketiadaan suatu kata dalam memutuskan klasifikasi yang tepat di dalam kelas mana pun.

Terdapat langkah-langkah yang perlu dilakukan untuk melakukan perhitungan bobot *Information Gain* sebagai berikut:

1. Langkah pertama menghitung nilai *entropy* menggunakan persamaan 11 dengan *S* merupakan himpunan data dan P(*i*) merupakan probabilitas dari kelas target *i* pada himpunan *S*. Serta *c* adalah jumlah atribut target.

$$\text{Entropy(S)} = -\sum_{i=1}^{c} \text{P(i)} \ \log_2 \text{P(i)} \quad (11)$$

2. Menghitung nilai *entropy* dengan atribut *A* pada set data *D* menggunakan persamaan 12 dengan *Entropy* (*Sv*) merupakan entropi dari set data *Sv*

dan *Sv* merupakan sub-set data dari *D* di mana setiap *instance* memiliki nilai *v* pada atribut *A*:

$$Entropy(S,A) = -\sum_{j=1}^{v} \frac{|D_j|}{D} Entropy(Sv) \quad (12)$$

Dengan *D* merupakan jumlah total *instance* (observasi) dalam set data *D*, dan $|D_j|$ adalah jumlah *instance* dalam set data *D* yang memiliki nilai *v* pada atribut *A*. Atribut *i* merupakan indeks *loop* untuk setiap nilai *v* pada atribut *A* dan *j* merupakan indeks *loop* untuk setiap nilai *v* pada atribut *A*.

3. Terakhir menggunakan Persamaan 13 berikut untuk menghitung nilai bobot *Information Gain*:

$$Gain(S,A) = -Entropy(S) - |Entropy(S,A)| \quad (13)$$

Nilai *Information Gain* adalah hasil dari nilai persamaan 12 dikurang dengan nilai persamaan 11.

2.9. Confusion Matrix

Teknik *Confusion Matrix* biasa digunakan dalam *data mining* untuk menghitung tingkat keakuratan dalam klasifikasi atau model yang sudah dibuat. Tujuan dari tahapan ini untuk mengetahui kebenaran hasil metode yang dipakai [15]. Ilustrasi untuk tabel *Confusion matrix* dapat dilihat pada Tabel 3.

Tabel 3. *Confusion Matrix*

		Nilai Aktual		
		Positif	**Netral**	**Negatif**
Nilai Prediksi	**Positif**	True Positive (TN)	False Positive (FP)	False Positive (FP)
	Netral	False Neutral (FL)	True Neutral (TL)	False Neutral (FL)
	Negatif	False Negative (FN)	False Negative (FN)	True Negative (TN)

Berdasarkan tabel *confusion matrix* pada Tabel 3, diperoleh:

a. *True Positive* (TP) : Data dengan aktual positif dan diprediksi positif.
b. *False Positive* (FP) : Data dengan aktual bukan positif yang diprediksi positif.
c. *True Negatif* (TN) : Data dengan aktual negatif yang diprediksi negatif.
d. *False Negatif* (FN) : Data dengan aktual bukan negatif yang diprediksi negatif.
e. *True Neutral* (TL) : Data dengan aktual netral yang diprediksi netral.
f. *False Neutral* (FL) : Data dengan aktual bukan netral yang diprediksi netral.

Confusion matrix tersebut digunakan untuk menghitung nilai *accuracy*, *precision*, *recall* dan *f-measure* sebagai hasil evaluasi dari model.

$$Accuracy = \frac{TP+TL+TN}{TP+TL+TN+FN+FP+FL} \quad (14)$$

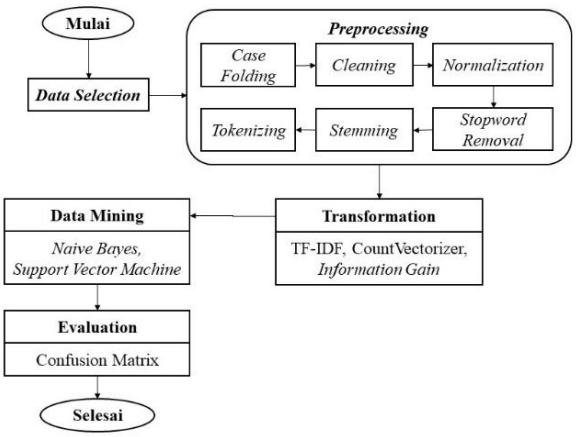
$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FL+FN} \quad (16)$$

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall+Precision} \quad (17)$$

3. METODE PENELITIAN

Penelitian ini menggunakan metode KDD (*Knowledge Discovery in Database*) sebagai pendekatan penelitian, karena metode ini memiliki keunggulan dalam mengidentifikasi pola yang terstruktur dari kumpulan data yang kompleks, sehingga memudahkan pemahaman [16]. Gambar 2 menunjukkan alur penelitian yang menerapkan metode KDD.



Gambar 2. Alur kerja KDD (*Knowledge Discovery in Database*)

Alur penelitian pada Gambar 2 dijelaskan sebagai berikut:

3.1. Data Selection

Pada tahap ini dilakukan pengambilan data dari Twitter. Data yang diambil adalah data pada rentang waktu 1 November 2022 hingga 3 Mei 2023 dengan kata kunci 'Perkembangan *Artificial Intelligence*' dan 'Teknologi *Artificial Intelligence*'. Setelah data didapatkan dan diseleksi, lalu dilakukan pelabelan dengan ChatGPT dan BingChat.

3.2. Preprocessing

Pada tahap ini dilakukan pengurangan volume kata untuk menghilangkan noise. Tahap ini terdiri dari 6 proses yaitu:

1. *Case Folding*, merupakan tahap mengubah teks menjadi huruf kecil
2. *Cleaning*, merupakan tahap pembersihan pada data, yaitu menghapus emoticon, hashtag, username, url, maupun karakter yang tidak diperlukan.
3. *Normalization*, merupakan tahap memperbaiki kata yang tidak baku, disingkat atau kesalahan ketik.
4. *Tokenizing*, merupakan tahap memecah kalimat jadi kata per kata.
5. *Stemming*, merupakan tahap menghapus imbuhan pada kata.

6. *Stopword Removal*, merupakan tahap menghapus kata yang tidak memiliki nilai pada proses *data mining*, seperti kata sambung.

3.3. Transformation

Pada tahap ini, kata-kata dibobot menggunakan algoritma TF-IDF (*Term Frequency Inverse Document Frequency*) dan *Count Vectorizer*. Untuk meningkatkan akurasi, digunakan N-gram yang bertujuan untuk memisahkan kata berdasarkan urutan kata pada suatu kalimat, pada penelitian ini digunakan model *bigram* dan *unigram*. Penggunaan ekstraksi fitur akan dibandingkan saat menggunakan N-gram dan tanpa N-gram. Penelitian ini juga membandingkan akurasi dengan menggunakan dan tanpa seleksi fitur *Information Gain*.

3.4. Data Mining

Pada tahap ini, dilakukan pembelajaran dan klasifikasi dengan menggunakan metode *Naive Bayes* dan *Support Vector Machine* menggunakan dataset *multiclass* yaitu sentimen positif, negatif, dan netral. Sebelum dilakukan klasifikasi, data dibagi menjadi data latih dan data uji yang masing-masing dengan rasio 70:30, 80:20, dan 90:10. Pembagian tersebut bertujuan untuk menentukan rasio terbaik dengan akurasi tertinggi.

3.5. Evaluation

Tahap ini bertujuan untuk menguji performa dari algoritma klasifikasi yang digunakan pada penelitian ini yaitu *Naive Bayes* dan *Support Vector Machine* (SVM). Tahap ini menggunakan *Confusion Matrix* untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f-measure* dari model yang telah dilatih.

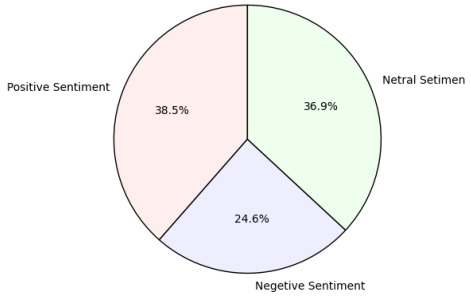
4. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah bagaimana melakukan klasifikasi terhadap pandangan masyarakat terhadap perkembangan *Artificial Intelligence* di Indonesia yang diambil melalui platform media sosial Twitter dengan menggunakan algoritma *Naive Bayes* dan *Support Vector Machine*.

4.1. Data Selection

Pada proses ini dikumpulkan data dari Twitter mengenai perkembangan *Artificial Intelligence* yang diambil dalam rentang waktu 1 November 2022 hingga 3 Mei 2023. Data yang telah dikumpulkan berjumlah 2.579 data dan dilabel dengan ChatGPT dan BingChat. Pada proses pelabelan, sebanyak 1629 data diklasifikasi menggunakan BingChat dan 950 data diklasifikasi menggunakan ChatGPT, alasan menggunakan kedua platform adalah karena limitasi perintah per harinya sehingga aplikasi digunakan bergantian. Berdasarkan hasil dari pelabelan dengan menggunakan model GPT, dihasilkan 536 data dengan sentimen negatif, 839 data dengan sentimen positif, 803 data dengan sentimen netral, 111 data duplikat, dan 290 data dengan kesalahan seperti kalimat tidak lengkap, penggunaan bahasa daerah, atau tidak dapat dimengerti oleh ChatGPT dan BingChat. Data yang mengalami kesalahan tersebut tidak akan digunakan dan akan dihapus, sehingga tersisa 2.178 data yang valid.

Percobaan	TD-IDF	*Count Vectorizer*	N-gram	*Information Gain*
1	✓			
2		✓		
3	✓		✓	
4		✓	✓	
5	✓			✓
6		✓		✓
7	✓		✓	✓
8		✓	✓	✓



Gambar 3. Diagram jumlah sentimen

4.2. Preprocessing

Pada proses ini data sentimen diolah dengan membersihkan dan memperbaiki data, seperti menghilangkan *emoticon*, *hashtag*, *usertag*, *url*, angka, simbol, kata sambung, imbuhan kata dan melakukan normalisasi kata. Tahap ini meliputi proses *case folding*, *cleaning*, *normalization*, *tokenizing*, *stemming*, dan *stopword removal*.

4.3. Transformation

Pada proses ini dilakukan transformasi data dengan ekstraksi fitur dan seleksi fitur. Ekstraksi fitur yang digunakan yaitu TF-IDF dan *Count Vectorizer*. Kedua ekstraksi fitur ini juga akan diuji dengan N-gram menggunakan model *unigram* dan *bigram*. Sedangkan pada seleksi fitur menggunakan *Information Gain*. Penelitian ini bertujuan untuk memeriksa dan membandingkan segala kemungkinan kombinasi algoritma transformasi. Hal ini bertujuan untuk menemukan kombinasi transformasi yang memberikan akurasi terbaik. Oleh karena itu, sebuah tabel percobaan telah disusun dan disajikan dalam Tabel 4. Tabel tersebut berisi semua kombinasi yang akan diuji. Tanda centang pada tabel memiliki makna bahwa algoritma tersebut digunakan sesuai dengan judul kolom.

Tabel 4. Kombinasi percobaan *data mining*

Dari hasil ekstraksi fitur yang dilakukan, saat tidak menggunakan N-gram, didapatkan fitur sebanyak 6401 fitur dari 2178 dokumen. Sedangkan saat menggunakan N-gram didapatkan 34.399 fitur.

4.4. Data Mining

Pada proses ini dilakukan pembagian data latih dan uji dengan rasio masing-masing yaitu 70:30, 80:20 dan 90:10 yang bertujuan untuk menentukan rasio terbaik. Pada percobaan ke-1 hingga ke-4, dilakukan tanpa menggunakan seleksi fitur *Information Gain*. Hasil dari percobaan ini disajikan pada Tabel 5.

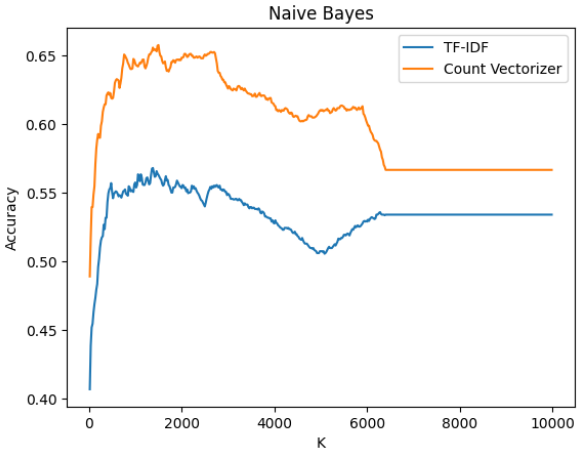
Tabel 5. Hasil percobaan 1-4

Metode	Ekstraksi Fitur	Rasio			Rata-rata
		70:30	80:20	90:10	
Naive Bayes	TF-IDF	52.752%	52.982%	**53.211%**	52.98%
SVM	TF-IDF	58.716%	**61.468%**	59.633%	**59.94%**
Naive Bayes	CV	56.575%	**59.633%**	55.046%	57.08%
SVM	CV	54.281%	**57.569%**	54.587%	55.48%
Naive Bayes	TF-IDF (N-gram)	**52.293%**	52.064%	50.917%	51.76%
SVM	TF-IDF (N-gram)	57.645%	**59.174%**	57.798%	58.21%
Naive Bayes	CV (N-gram)	56.728%	**59.862%**	55.046%	57.21%
SVM	CV (N-gram)	56.116%	**58.716%**	57.339%	57.39%

Hasil yang tertera pada Tabel 5 menunjukkan bahwa rasio 80:20 memiliki tingkat akurasi yang baik dibanding rasio yang lain, dikarenakan 6 dari 8 percobaan memiliki akurasi tertinggi pada rasio 80:20. Pada data tersebut akurasi tertinggi dari *Naive Bayes* yaitu 59.86% dengan menggunakan *Count Vectorizer* dan N-gram, dan akurasi tertinggi dari SVM yaitu 61.47% dengan menggunakan TF-IDF tanpa N-gram. Dari informasi yang sudah didapatkan, dapat disimpulkan *Naive Bayes* lebih optimal saat menggunakan *Count Vectorizer*, sedangkan SVM lebih optimal saat menggunakan TF-IDF.

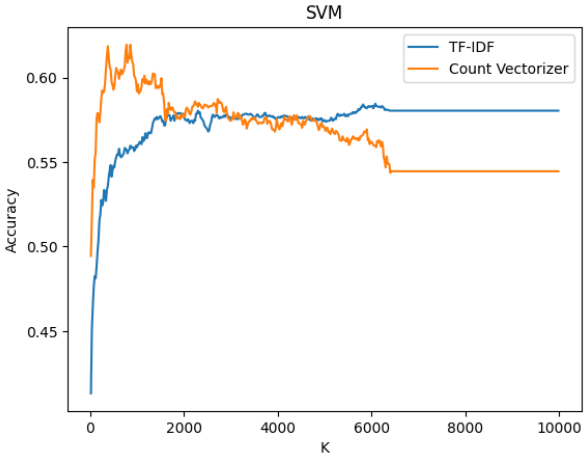
Pada percobaan ke-5 hingga ke-8 dilakukan menggunakan seleksi fitur *Information Gain*. Percobaan ini dilakukan untuk menentukan jumlah fitur (k) yang dipakai. Pertama, dilakukan pembobotan dengan *Information Gain* untuk menentukan tingkat kepentingan kata. Hasil pembobotan lalu diurutkan dari nilai bobot terbesar atau terpenting. Jumlah fitur diambil mulai dari bobot terbesar hingga terkecil. Jumlah fitur (k) ditentukan dengan membuat grafik akurasi berdasarkan jumlah fitur yang dipakai. Jumlah fitur (k) merupakan grafik dinamis dan grafik dimulai dengan nilai k=20 hingga k=10000 dengan setiap langkahnya berjarak 20. Dari hasil grafik tersebut, diambil nilai k dengan akurasi tertinggi.

Langkah pertama yaitu menentukan nilai k untuk *Naive Bayes* dengan ekstraksi fitur TF-IDF dan *Count Vectorizer* tanpa menggunakan N-gram. Langkah ini dilakukan menggunakan 10-*fold cross validation*. Alasannya karena teknik ini menggunakan semua *dataset* sebagai data latih dan data uji, sehingga tidak perlu menentukan rasio tertentu. Teknik 10-*fold cross validation* cukup umum digunakan dan cara kerjanya membagi *dataset* menjadi 10 bagian dan setiap bagiannya dijadikan data latih dan data uji. Kemudian, akurasi yang didapat pada setiap uji diambil rata-ratanya. Dengan menggunakan 10-*fold cross validation* didapatkan grafik yang disajikan pada Gambar 4. Grafik tersebut menunjukkan nilai akurasi tertinggi pada TF-IDF adalah 0.568 dengan nilai k=1380, dan pada *Count Vectorizer* adalah 0.657 dengan nilai k=1500.



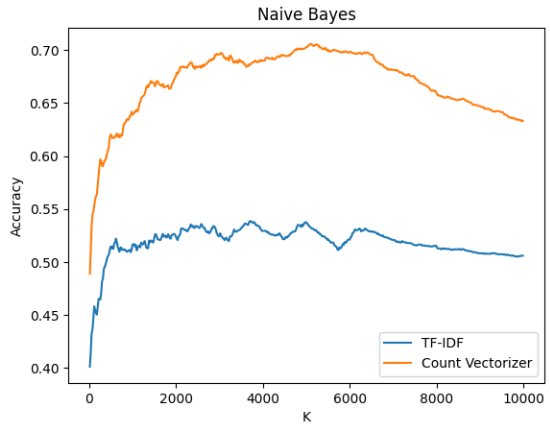
Gambar 4. Grafik akurasi dan jumlah fitur Naive Bayes.

Langkah kedua yaitu menentukan nilai k untuk SVM dengan ekstraksi fitur TF-IDF dan *Count Vectorizer* tanpa menggunakan N-gram. Dengan menggunakan 10-*fold cross validation* didapatkan grafik yang disajikan pada Gambar 5. Grafik tersebut menunjukkan nilai akurasi tertinggi pada TF-IDF adalah 0.584 dengan nilai k=6080, dan pada *Count Vectorizer* adalah 0.619 dengan nilai k=780.



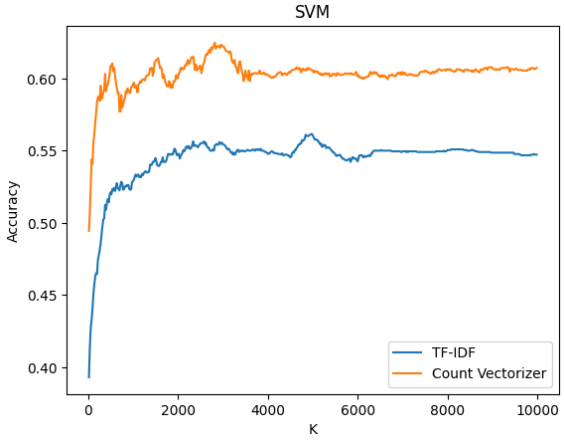
Gambar 5. Grafik akurasi dan jumlah fitur SVM.

Langkah ketiga yaitu menentukan nilai k untuk *Naive Bayes* dengan ekstraksi fitur TF-IDF dan *Count Vectorizer* yang menggunakan N-gram. Dengan menggunakan 10-*fold cross validation* didapatkan grafik yang disajikan pada Gambar 6. Grafik tersebut menunjukkan nilai akurasi tertinggi pada TF-IDF adalah 0.538 dengan nilai k=3700, dan pada CV adalah 0.705 dengan nilai k=5100.



Gambar 6. Grafik akurasi dan jumlah fitur Naive Bayes.(N-gram)

Langkah keempat yaitu menentukan nilai k untuk SVM dengan ekstraksi fitur TF-IDF dan *Count Vectorizer* yang menggunakan N-gram. Dengan menggunakan 10-*fold cross validation* didapatkan grafik yang disajikan pada Gambar 7. Grafik tersebut menunjukkan nilai akurasi tertinggi pada TF-IDF adalah 0.561 dengan nilai k=4960, dan pada CV adalah 0.625 dengan nilai k=2820



Gambar 7. Grafik akurasi dan jumlah fitur SVM.(N-gram)

Dari data yang ditemukan tersebut, ditemukan nilai k untuk masing-masing kombinasi, kemudian dibuat tabel akurasi menggunakan nilai k yang sudah ditemukan. Tabel tersebut digunakan untuk mencari rasio terbaik seperti yang disajikan pada Tabel 6.

Tabel 6. Hasil percobaan 5-8

Metode	**Ekstraksi Fitur**	**Nilai *k***	**Rasio**			**Rata-rata**
			70:30	**80:20**	**90:10**	
Naive Bayes	TF-IDF	1380	56.422%	**56.880%**	53.669%	55.66%
SVM	TF-IDF	6080	58.409%	**60.550%**	59.633%	59.53%
Naive Bayes	CV	1500	62.997%	**67.660%**	65.748%	65.47%
SVM	CV	780	58.716%	**59.633%**	57.798%	58.72%
Naive Bayes	TF-IDF (N-gram)	3700	53.058%	**54.587%**	**54.587%**	54.08%
SVM	TF-IDF (N-gram)	4960	55.963%	58.945%	**60.092%**	58.33%
Naive Bayes	CV (N-gram)	5100	66.820%	**72.018%**	70.642%	**69.83%**
SVM	CV (N-gram)	2820	61.620%	**62.385%**	60.550%	61.52%

Dari hasil yang tertera dalam Tabel 6, terlihat bahwa *Information Gain* cukup baik dalam meningkatkan akurasi dari *Naive Bayes* dan SVM jika dibandingkan tanpa menggunakan seleksi fitur. Pada data tersebut, akurasi tertinggi dari *Naive Bayes* mencapai 72.02% dengan menggunakan *Count Vectorizer* dan N-gram, sedangkan akurasi tertinggi dari SVM mencapai 62.38% dengan menggunakan *Count Vectorizer* dan N-gram. Penggunaan *Information Gain* sangat berpengaruh pada *Naive Bayes*, dikarenakan dapat meningkatkan akurasi sebanyak 12.16% saat menggunakan ekstraksi fitur *Count Vectorizer* dengan N-gram. Data tersebut juga menunjukkan bahwa pembagian data latih dan uji terbaik yaitu 80:20 dibanding rasio yang lain.

Dari hasil percobaan 1-4 dengan tanpa *Information Gain* jika dibandingkan dengan hasil percobaan 5-8 dengan *Information Gain* menunjukkan peningkatan akurasi hampir di seluruh kombinasi. Peningkatan tersebut bisa dilihat dari rata-rata akurasi pada setiap kombinasi dari percobaan tanpa *Information Gain* dan dengan *Information Gain*. Peningkatan tersebut bisa dilihat pada Tabel 7 yang menunjukkan peningkatan rata-rata akurasi hampir di seluruh kombinasi percobaan saat menggunakan *Information Gain*.

Tabel 7. Peningkatan akurasi dengan *Information Gain*

Metode	**Ekstraksi Fitur**	**Rata-rata akurasi**	
		Tanpa Information Gain	**Dengan Information Gain**
Naive Bayes	TF-IDF	52.98%	**55.66%**
SVM	TF-IDF	**59.94%**	59.53%
Naive Bayes	CV	57.08%	**65.47%**
SVM	CV	55.48%	**58.72%**
Naive Bayes	TF-IDF (N-gram)	51.76%	**54.08%**
SVM	TF-IDF (N-gram)	58.21%	**58.33%**
Naive Bayes	CV (N-gram)	57.21%	**69.83%**
SVM	CV (N-gram)	57.39%	**61.52%**

Langkah selanjutnya yaitu mencari nilai akurasi, presisi, *recall*, dan *f-measure* dengan membuat *confusion matrix* pada data yang mendapat akurasi terbaik pada percobaan yang sudah dilakukan sebelumnya pada data yang dilakukan dengan dan tanpa *Information Gain*. Untuk membuat *confusion matrix* dilakukan dengan menggunakan validasi 10-*fold cross validation* sehingga seluruh *dataset* akan dijadikan sebagai data latih dan data uji. Data dengan

akurasi terbaik dengan dan tanpa *Information Gain* pada percobaan sebelumnya ditampilkan pada Tabel 8.

Tabel 8. Hasil percobaan dengan 10-*fold cross validation*

No	Motode Klasifikasi	Ekstraksi Fitur	Seleksi Fitur	Akurasi
1	Naive Bayes	CV (N-gram)	-	59.862%
2	SVM	TF-IDF	-	61.468%
3	Naive Bayes	CV (N-gram)	IG	72.018%
4	SVM	CV (N-gram)	IG	62.385%

Pada percobaan pertama, digunakan *Naive Bayes* tanpa menerapkan seleksi fitur *Information Gain*. Ekstraksi fitur yang digunakan adalah *Count Vectorizer* dengan N-gram. Selanjutnya, dilakukan 10-*fold cross validation* untuk memvalidasi hasil klasifikasi, dan hasilnya dapat dilihat pada Tabel 9. Selain itu, juga disajikan tabel *confusion matrix* dari hasil klasifikasi pada Tabel 10.

Tabel 9. Naive Bayes dengan *k-fold cross validation*

k	CV (N-gram)
1	0.573
2	0.518
3	0.564
4	0.592
5	**0.596**
6	0.569
7	0.569
8	0.541
9	0.544
10	0.562
Rata-rata	**0.563**

Tabel 10. Confusion Matrix percobaan 1

		Nilai Aktual		
		Negatif	**Positif**	**Netral**
Nilai Prediksi	**Negatif**	**320**	96	163
	Positif	74	**530**	264
	Netral	142	213	**376**

Tabel 10 merupakan tabel *confusion matrix* dari *Naive Bayes* menggunakan *Count Vectorizer* dengan N-gram. Dengan menggunakan 10-*fold cross validation*, akurasi yang didapatkan sebesar 56.29%. Selain itu, presisi untuk sentimen negatif adalah 59.70%, presisi untuk sentimen positif adalah 63.17%, dan presisi untuk sentimen netral adalah 46.82%. Sedangkan *recall* untuk sentimen negatif adalah 55.27%, *recall* untuk sentimen positif adalah 61.06%, dan recall untuk sentimen netral adalah 51.44%. *F-measure* untuk sentimen negatif adalah 57.40%, *f-measure* untuk sentimen positif adalah 62.10%, dan *f-measure* untuk sentimen netral adalah 49.02%.

Pada percobaan kedua, digunakan SVM tanpa menerapkan seleksi fitur *Information Gain*. Ekstraksi fitur yang digunakan adalah TF-IDF tanpa N-gram. Selanjutnya, dilakukan 10-*fold cross validation* untuk memvalidasi hasil klasifikasi, dan hasilnya dapat dilihat pada Tabel 11. Selain itu, juga disajikan tabel *confusion matrix* dari hasil klasifikasi pada Tabel 12.

Tabel 11. SVM dengan *k-fold cross validation*

k	TF-IDF
1	0.550
2	0.550
3	0.587
4	0.605
5	0.610
6	0.550
7	0.550
8	0.587
9	0.585
10	**0.636**
Rata-rata	**0.580**

Tabel 12. Confusion Matrix percobaan 2

		Nilai Aktual		
		Negatif	**Positif**	**Netral**
Nilai Prediksi	**Negatif**	**292**	60	110
	Positif	90	**519**	240
	Netral	154	260	**453**

Tabel 12 merupakan tabel *confusion matrix* dari SVM menggunakan TF-IDF tanpa N-gram. Dengan menggunakan 10-*fold cross validation*, akurasi yang didapatkan sebesar 58.03%. Presisi untuk sentimen negatif adalah 54.48%, presisi untuk sentimen positif adalah 61.86%, dan presisi untuk sentimen netral adalah 56.41%. *Recall* untuk sentimen negatif adalah 63.20%, *recall* untuk sentimen positif adalah 61.13%, dan *recall* untuk sentimen netral adalah 52.25%. *F-measure* untuk sentimen negatif adalah 58.52%, *f-measure* untuk sentimen positif adalah 61.49%, dan *f-measure* untuk sentimen netral adalah 54.25%.

Pada percobaan ketiga, digunakan *Naive Bayes* dengan menerapkan seleksi fitur *Information Gain*. Ekstraksi fitur yang digunakan adalah *Count Vectorizer* dengan N-gram. Selanjutnya, dilakukan 10-*fold cross validation* untuk memvalidasi hasil klasifikasi, dan hasilnya dapat dilihat pada Tabel 13. Selain itu, juga disajikan tabel *confusion matrix* dari hasil klasifikasi pada

Tabel 14.

Tabel 13. Naive Bayes dengan *k-fold cross validation*

k	CV (N-gram)
1	0.683
2	0.679
3	0.725
4	**0.752**
5	0.716
6	0.725
7	0.693
8	0.642
9	0.723
10	0.719
Rata-rata	**0.706**

Tabel 14. Confusion Matrix percobaan 3

		Nilai Aktual		
		Negatif	**Positif**	**Netral**
Nilai Prediksi	**Negatif**	**352**	42	83
	Positif	48	**634**	169
	Netral	136	163	**551**

Tabel 14 merupakan tabel *confusion matrix* dari *Naive Bayes* menggunakan *Count Vectorizer* dengan N-gram. Dengan menggunakan 10-*fold cross validation*, akurasi yang didapatkan sebesar 70.57%. Presisi untuk sentimen negatif adalah 65.67%, presisi untuk sentimen positif adalah 75.57%, dan presisi untuk sentimen netral adalah 68.62%. *Recall* untuk sentimen negatif adalah 73.79%, *recall* untuk sentimen positif adalah 74.50%, dan *recall* untuk sentimen netral adalah 64.82%. *F-measure* untuk sentimen negatif adalah 69.50%, *f-measure* untuk sentimen positif adalah 75.03%, dan *f-measure* untuk sentimen netral adalah 66.67%.

Pada percobaan keempat, digunakan SVM dengan menerapkan seleksi fitur *Information Gain*. Ekstraksi fitur yang digunakan adalah *Count Vectorizer* dengan N-gram. Selanjutnya, dilakukan 10-*fold cross validation* untuk memvalidasi hasil klasifikasi, dan hasilnya dapat dilihat pada Tabel 15. Selain itu, juga disajikan tabel *confusion matrix* dari hasil klasifikasi pada Tabel 16.

Tabel 15. SVM dengan *k-fold cross validation*

k	**CV (N-gram)**
1	0.583
2	0.596
3	0.615
4	**0.674**
5	0.624
6	0.633
7	0.633
8	0.601
9	0.659
10	0.631
Rata-rata	**0.625**

Tabel 16. Confusion Matrix percobaan 4

		Nilai Aktual		
		Negatif	**Positif**	**Netral**
Nilai Prediksi	**Negatif**	**337**	88	137
	Positif	76	**545**	187
	Netral	123	206	**479**

Tabel 16 merupakan tabel *confusion matrix* dari SVM menggunakan *Count Vectorizer* dengan N-gram. Dengan menggunakan 10-*fold cross validation*, akurasi yang didapatkan sebesar 62.49%. Presisi untuk sentimen negatif adalah 62.87%, presisi untuk sentimen positif adalah 64.96%, dan presisi untuk sentimen netral adalah 59.65%. *Recall* untuk sentimen negatif adalah 59.96%, *recall* untuk sentimen positif adalah 67.45%, dan *recall* untuk sentimen netral adalah 59.28%. *F-measure* untuk sentimen negatif adalah 61.38%, *f-measure* untuk sentimen positif adalah 66.18%, dan *f-measure* untuk sentimen netral adalah 59.47%.

Tabel 17. Hasil percobaan dengan 10-*fold cross validation*

No	**Motode Klasifikasi**	**Ekstraksi Fitur**	**Seleksi Fitur**	**Akurasi dengan Rasio**	**Akurasi dengan 10-*fold cross validation***
1	Naive Bayes	CV (N-gram)	-	59.862%	56.29%
2	SVM	TF-IDF	-	61.468%	58.03%
3	Naive Bayes	CV (N-gram)	IG	72.018%	70.57%
4	SVM	CV (N-gram)	IG	62.385%	62.49%

Hasil percobaan dengan validasi silang (*fold cross validation*) menunjukkan akurasi yang menurun dibanding dengan akurasi hasil terbaik dari rasio data latih dan data uji seperti yang terlihat pada Tabel 17. Akan tetapi hasil tersebut lebih akurat dikarenakan tidak tergantung rasio data latih dan data uji. Hasil percobaan juga menunjukkan bahwa penggunaan seleksi fitur *Information Gain* cukup baik dalam meningkatkan akurasi klasifikasi. Seleksi *fitur Information Gain* dapat mengurangi dimensi fitur yang digunakan dalam klasifikasi, sehingga dapat mengurangi *noise* dan *redundansi*. Namun, penting untuk menentukan nilai k yang optimal atau jumlah fitur yang dipilih untuk menghilangkan *noise*. Nilai k yang optimal adalah nilai k yang memberikan akurasi tertinggi.

5. KESIMPULAN DAN SARAN

Dari hasil percobaan yang dilakukan, ditemukan bahwa rasio pembagian data latih dan data uji yang paling optimal adalah 80:20. Seleksi fitur *Information Gain* dapat memberikan peningkatan akurasi yang signifikan untuk algoritma *Naive Bayes*. Hasil penelitian juga menunjukkan bahwa *Naive Bayes* memiliki performa yang lebih baik dengan menggunakan *Count Vectorizer* sebagai metode ekstraksi fitur, sedangkan SVM memiliki performa yang lebih baik dengan menggunakan TF-IDF sebagai metode ekstraksi fitur. Penggunaan N-gram dapat meningkatkan akurasi klasifikasi, terutama untuk *Count Vectorizer*, tetapi dapat menurunkan akurasi klasifikasi untuk TF-IDF pada beberapa percobaan. Hasil eksperimen menunjukkan bahwa SVM memberikan kinerja yang lebih baik dibandingkan dengan *Naive Bayes*. Akurasi tertinggi yang dihasilkan oleh SVM tanpa menggunakan *Information Gain* sebesar 61.47% menggunakan ekstraksi fitur TF-IDF. Sedangkan, *Naive Bayes* mencapai akurasi tertinggi

sebesar 59.86% pada tanpa menggunakan *Information Gain* dengan menggunakan ekstraksi fitur *Count Vectorizer* dengan N-gram.

Seleksi fitur *Information Gain* mampu meningkatkan akurasi klasifikasi pada *Naive Bayes* dan. Pada *Naive Bayes*, seleksi fitur *Information Gain* mampu meningkatkan akurasi hingga 12%. Dengan menggunakan *Information Gain*, akurasi *Naive Bayes* meningkat dari 59.86% menjadi 72.02% dengan menggunakan *Count Vectorizer* dengan N-gram. Sedangkan pada SVM mampu meningkatkan dari 58.72% menjadi 62.38% saat menggunakan ekstraksi fitur *Count Vectorizer* dengan N-gram.

DAFTAR PUSTAKA

[1] F. Kurnia, "Artificial Intelligence: Pengertian, Contoh, dan Tantangannya," *dailysocial*, Jan. 03, 2023. https://dailysocial.id/post/artificial-intelligence (accessed May 04, 2023).

[2] D. Sierra, "Algoritma TF — IDF," *Medium*, Feb. 13, 2019. https://dltsierra.medium.com/algoritma-tf-idf-633e17d10a80 (accessed May 03, 2023).

[3] A. Isnanda, Y. Umaidah, and J. H. Jaman, "Implementasi Naïve Bayes Classifier Dan Information Gain Pada Analisis Sentimen Penggunaan E-Wallet Saat Pandemi," *Jurnal Teknologi Informatika dan Komputer*, vol. 7, no. 2, pp. 144–153, Sep. 2021, doi: 10.37012/jtik.v7i2.648.

[4] E. Salim and M. Syafrullah, "Analisis Sentimen Pada Ulasan Pelayanan Suku Dinas Kependudukan Dan Pencatatan Sipil Kota Administrasi Jakarta Barat Menggunakan Algoritme K-Nearest Neighbor," *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 20, no. 1, pp. 58–65, 2023, [Online]. Available: https://kemsalim.space/ulasan_dukcapil/

[5] S. Khomsah and A. S. Aribowo, "Model Text-Preprocessing Komentar Youtube Dalam Bahasa Indonesia," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 4, pp. 648–654, 2020.

[6] R. Sarwiyana, "Memahami fit_transform() pada Module CountVectorizer," *Medium*, May 20, 2019. https://medium.com/sesuapnasi/memahami-fit-transform-pada-module-countvectorizer-d78bc8b8036f (accessed Jul. 30, 2023).

[7] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 2, p. 406, Apr. 2021, doi: 10.30865/mib.v5i2.2835.

[8] Francisco Caio Lima Paiva, "Can ChatGPT Compete with Domain-Specific Sentiment Analysis Machine Learning Models," *Meduim*, Apr. 26, 2023. https://towardsdatascience.com/can-chatgpt-compete-with-domain-specific-sentiment-analysis-machine-learning-models-cdcd9937b460 (accessed Jun. 25, 2023).

[9] H. C. Husada and A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, pp. 18–26, Feb. 2021, doi: 10.34148/teknika.v10i1.311.

[10] H. Marius, "Multiclass Classification with Support Vector Machines (SVM), Dual Problem and Kernel Functions," *Medium*, Jun. 09, 2020. https://towardsdatascience.com/multiclass-classification-with-support-vector-machines-svm-kernel-trick-kernel-functions-f9d5377d6f02 (accessed May 16, 2023).

[11] A. Rohanah, B. A. Dermawan, and I. Purnamasari, "Klasifikasi Ulasan Pengguna Zoom Cloud Meetings Menggunakan Metode Information Gain dan Naïve Bayes Classifier," *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 2, pp. 348–357, Jun. 2021, doi: 10.32493/informatika.v6i2.10728.

[12] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi," *Jurnal Teknoinfo*, vol. 14, no. 2, p. 115, Jul. 2020, doi: 10.33365/jti.v14i2.679.

[13] A. B. P. Negara, H. Muhardi, and I. M. Putri, "Analisis Sentimen Maskapai Penerbangan Menggunakan Metode Naive Bayes dan Seleksi Fitur Information Gain," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 3, pp. 599–606, Jun. 2020, doi: 10.25126/jtiik.202071947.

[14] H. S. Utama, D. Rosiyadi, D. Aridarma, and B. S. Prakoso, "Sentimen Analisis Kebijakan Ganjil Genap Di Tol Bekasi Menggunakan Algoritma Naive Bayes dengan Optimalisasi Information Gain," *Jurnal Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 247–254, Sep. 2019, doi: 10.33480/pilar.v15i2.705.

[15] A. Dwi Pangestu, I. Ernawati, and N. Chamidah, "Analisis Sentimen Terhadap PPKM Darurat Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes Dengan Seleksi Fitur Information Gain," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya*, vol. 3, no. 2, pp. 662–671, Aug. 2022.

[16] S. Ramos *et al.*, "Data mining techniques for electricity customer characterization," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 475–488. doi: 10.1016/j.procs.2021.04.168.