Comparison Of Naïve Bayes And Support Vector Machines In Classifying Sentiment On Twitter About Artificial Intelligence Development

Iqbal Maulana[1*], Roland Vincent Sitanggang[2], Oman Komarudin[3]

[1,2,3]Universitas Singaperbangsa Karawang
Email: *iqbal.maulana@staff.unsika.ac.id

Abstrak. Analisis sentimen merupakan bagian dari data mining yang digunakan untuk mengolah dan memproses teks dengan tujuan untuk mengetahui bagaimana opini atau pandangan masyarakat tentang suatu isu atau masalah tertentu. Metode klasifikasi yang digunakan untuk melakukan analisis sentimen pada data berupa teks, diantaranya *Naive Bayes* dan *Support Vector Machine* (SVM). Dalam mengevaluasi performa model klasifikasi yang telah dibuat, biasanya akan diukur nilai akurasinya. Oleh karena itu, penelitian ini bertujuan untuk membandingkan performa dari model klasifikasi sentimen yang menggunakan metode *Naive Bayes* dan SVM, dengan TF-IDF dan *CountVectorizer* sebagai ekstraksi fitur serta *Information Gain* sebagai seleksi fitur. Selain itu, digunakan juga N-gram sebagai upaya untuk dapat meningkatkan akurasi model klasifikasi. Penelitian ini menggunakan dataset berupa cuitan pengguna Twitter tentang perkembangan *Artificial Intelligence*. Data tersebut nantinya dikategorikan menjadi dua kelas, yaitu positif dan negatif, serta akan diolah dengan menggunakan tahapan *knowledge discovery in databases* (KDD). Hasil penelitian menunjukkan bahwa model hasil *Naive Bayes* mendapatkan akurasi tertinggi saat menggunakan ekstraksi fitur *CountVectorizer*, sedangkan model hasil SVM mendapatkan akurasi tertinggi saat menggunakan TF-IDF. Selain itu, penggunaan *Information Gain* ternyata dapat meningkatkan nilai akurasi model hasil *Naive Bayes* sebesar 12% menggunakan *CountVectorizer* dengan N-gram. Namun penggunaan *Information Gain* justru menurunkan nilai akurasi model hasil SVM sebesar 0,73% menggunakan TF-IDF dengan N-gram.

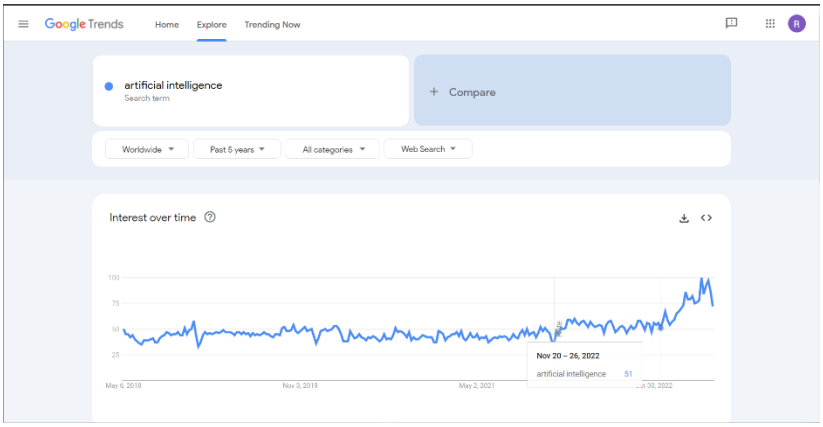
Kata kunci: *Naive Bayes; Support Vector Machine; Information Gain; Ekstraksi Fitur.*

1 Pendahuluan

Perkembangan *Artificial Intelligence* saat ini ternyata tidak selalu memberikan kesan positif bagi masyarakat. Pada media sosial, tidak sedikit pengguna yang beranggapan bahwa *Artificial Intelligence* dapat membawa bencana di masa mendatang [1]. Misalkan dengan adanya aplikasi ChatGPT dan Midjourney, banyak di antara masyarakat yang merasa terancam dalam profesinya, seperti programmer dan seniman yang kedepannya mungkin dapat digantikan oleh AI atau mengalami penurunan pendapatan [2]. Bahkan sejak November 2022 telah marak komentar di media sosial yang disebabkan oleh

Diterima 25 Mei 2024 Direvisi 28 Mei 2024 Diterima untuk publikasi 31 Mei 2024

popularitas teknologi *Artificial Intelligence* ini. Hal tersebut dapat dilihat berdasarkan data pada Google Trends yang menunjukkan peningkatan pencarian seputar *Artificial Intelligence* sejak perkiraan tanggal 27 November 2022 dalam rentang waktu 5 tahun terakhir seperti yang terlihat pada Gambar 1.



Gambar 1. Grafik Peningkatan Pencarian Tentang *Artificial Intelligence* Kurun Waktu 5 Tahun di Google Trends

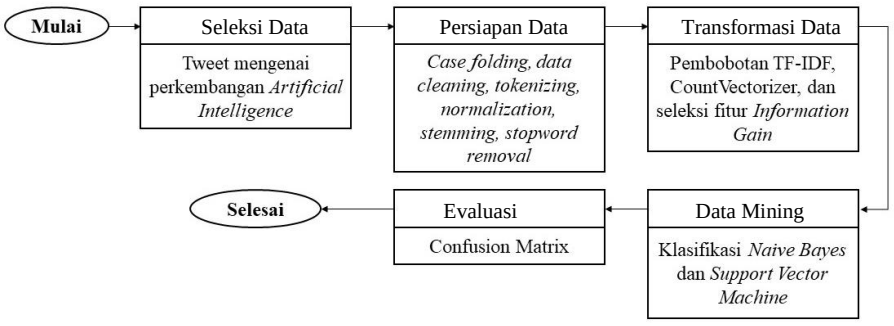
Kehadiran aplikasi yang dapat berkomunikasi seperti ChatGPT membuat topik *Artificial Intelligence* (AI) menjadi ramai dibicarakan di media sosial salah satunya twitter [3]. Dengan adanya opini yang beragam di Twitter tentang perkembangan AI tersebut, maka perlu dilakukan analisis sentimen untuk mengetahui bagaimana perkembangan *Artificial Intelligence* dipandang oleh masayarakat. Menurut Arsi dan Waluyo [4] analisis sentimen digunakan untuk mengklasifikasi antara kategori positif dan negatif dari sebuah opini. Klasifikasi ini dapat menggunakan beberapa metode seperti *Naive Bayes* (NB), *Support Vector Machine* (SVM), *Decision Tree*, dan *K-Nearest Neighbour* (K-NN) [5]. Metode klasifikasi tersebut memiliki kelebihan dan kekurangannya masing-masing, sehingga cocok dibandingkan performanya untuk mengetahui metode terbaik pada kasus data yang berbeda.

Penelitian sebelumnya pernah dilakukan oleh Ahmad dan Wahyu [6] yang mengimplementasikan metode *Naive Bayes* dan *Support Vector Machine* dalam menganalisis sentimen tentang pembobolan dan kebocoran data di Twitter. Hasil penelitian tersebut menunjukkan bahwa metode *Support Vector Machine* cukup akurat dan stabil dalam mengklasifikasikan data *tweet* dengan cara mencari *hyperline* paling baik untuk memisahkan data. Namun meskipun metode SVM menunjukkan hasil yang akurat dan stabil, tetapi dari segi hitungan metode *Naive Bayes* lebih unggul dalam proses klasifikasi. Dari hasil perhitungan *precision* dan *recall* dari *Naive Bayes* masing-masing sebesar 97%, dibandingkan dengan SVM yang hanya diperoleh *precision* 80% dan *recall* 93%. Pada penelitian kali ini akan digunakan metode klasifikasi *Naive Bayes* dan *Support Vector Machine* yang dikombinasikan dengan ekstraksi fitur

CountVectorizer dan TF-IDF, serta *Information Gain* sebagai seleksi fiturnya. Penelitian ini akan menggunakan dataset berupa cuitan (*tweet*) pengguna Twitter terhadap perkembangan *Artificial Intelligence* di Indonesia, yang nantinya akan dikategorikan ke dalam kelas positif dan negatif. Hasil penelitian sebelumnya telah membuktikan bahwa metode *Naive Baye*s dan *Support Vector Machine* cukup efektif dalam mengklasifikasikan kalimat opini di Twitter. Namun, dengan menggunakan seleksi fitur, tingkat akurasi klasifikasi dapat ditingkatkan [7]. Penelitian ini bertujuan untuk membandingkan performa *Naive Bayes* dan *Support Vector Machine* pada data dengan dua kelas yaitu positif dan negatif. Selain itu, penelitian ini juga akan mengukur performa *Naive Bayes* dan *Support Vector Machine* dengan dan tanpa seleksi fitur *Information Gain* serta menggunakan ekstraksi fitur TF-IDF dan *CountVectorizer*. Pada ekstraksi fitur juga akan menggunakan N-gram sebagai perbandingan untuk mengetahui performa dari ekstraksi fitur dengan menggunakan N-gram dan yang tanpa menggunakan N-gram.

2 Metode Penelitian

Penelitian ini mengikuti alur kerja *Knowledge Discovery in Database* (KDD) yang terdiri dari lima tahap, yaitu seleksi data, persiapan data, transformasi data, *data mining*, dan evaluasi seperti yang dapat dilihat pada Gambar 2. Pada tahap transformasi data, digunakan empat metode ekstraksi fitur, yaitu TF-IDF, TF-IDF dengan N-gram, *CountVectorizer*, dan *CountVectorizer* dengan N-gram. Pada penelitian ini akan diterapkan juga seleksi fitur *Information Gain* untuk meningkatkan akurasi dengan cara mengurangi dimensi fitur. Pada tahap *data mining*, digunakan dua algoritma klasifikasi, yaitu *Naive Bayes* dan *Support Vector Machine*, untuk mengklasifikasikan data opini yang telah dikumpulkan.



Gambar 2. Tahapan Penelitian Metode KDD

2.1 Seleksi Data

Dalam tahap ini dilakukan teknik *scraping* dari Twitter dengan tujuan melakukan pengambilan data. Data dari twitter tersebut diambil pada tanggal 1 November 2022 sampai 3 Mei 2023. Kata kunci yang digunakan yaitu

'Perkembangan AI', 'Perkembangan *Artificial Intelligence'*, 'Teknologi AI', dan 'Teknologi *Artificial Intelligence'*. Selanjutnya, data yang sudah diperoleh dan diseleksi, kemudian akan dilabeli dengan memanfaatkan ChatGPT dan BingChat.

2.2 Persiapan Data

Pada tahap persiapan data ini dilakukan pengurangan jumlah kata agar menghilangkan *noise*. Dalam tahap ini akan dilakukan *case folding, cleaning, stemming, tokenizing, stopword removal* dan *normalization*.

1. *Case Folding*
   Tahap awal pada *pre-processing* adalah *case folding*, yaitu mengubah huruf kapital menjadi huruf kecil.
2. *Cleaning*
   Cleaning merupakan tahap pembersihan pada data dari karakter-karakter yang tidak diperlukan, seperti *emoticon*, *username*, *hashtag*, termasuk *url*.
3. *Normalization*
   Pada tahap ini dilakukan perbaikan kata, misal kata tidak baku, kata yang disingkat atau kata yang terdapat kesalahan ketik.
4. *Tokenizing*
   Pada tahap ini dilakukan pemecahan kalimat menjadi kata per kata.
5. *Stemming*
   Pada tahap *stemming*, dilakukan penghilangan imbuhan pada setiap kata.
6. *Stopword Removal*
   Pada tahap ini kata-kata akan disaring sehingga menyisakan kata-kata penting yang memiliki nilai saja. Kata sambung seperti 'dan' atau 'atau' akan dihilangkan. Terdapat sejumlah kata *stopwords* pada *stopwords list* yang disediakan *library* Sastrawi yang dapat mempermudah menghapus kalimat *stopwords*.

2.3 Transformasi Data

Transformasi data merupakan tahap yang biasanya diimplementasikan melalui pengkodean dan memiliki beberapa fungsi lain yang digunakan untuk menemukan pola dalam proses *data mining*. Selama tahap ini, kata-kata dinilai menggunakan algoritma TF-IDF (*Term Frequency Inverse Document Frequency*) dan *CountVectorizer*. Algoritma TF-IDF digunakan untuk mengekstraksi teks dengan memberikan nilai pada setiap kata dalam teks tersebut. Penghitungan nilai dilakukan dengan cara menghitung frekuensi kemunculan kata dalam dokumen menggunakan metode TF, dan mencari tingkat kepentingan kata tersebut dalam dokumen menggunakan metode IDF [8]. Sedangkan *CountVectorizer* melakukan pembobotan berdasarkan frekuensi kemunculan masing-masing kata pada dokumen [9]. Dalam upaya meningkatkan akurasi, maka pada penelitian ini akan menggunakan N-gram

dengan model *bigram* untuk memisahkan kalimat menjadi dua kata dan *unigram* untuk memisahkan kalimat menjadi satu kata. Penggunaan N-gram dilakukan karena mempunyai keunggulan yaitu memperhatikan keterkaitan suatu kata dengan kata sebelumnya dan sesudahnya dalam suatu kalimat [10], sehingga dapat meningkatkan akurasi dalam proses klasifikasi. Sebagai perbandingan tingkat akurasi dengan dan tanpa seleksi fitur, dilakukan proses seleksi fitur menggunakan *Information Gain* agar meningkatkan performa klasifikasi dengan mengurangi dimensi *dataset*.

2.4 Data Mining

Dalam tahapan ini dilakukan proses klasifikasi terhadap *dataset* dua kelas yaitu sentimen positif dan negatif menggunakan algoritma *Naive Bayes* dan *Support Vector Machine*. Pembagian data menjadi dua bagian yaitu *data traning* dan *data testing* perlu dilakukan sebelum proses klasifikasi dikerjakan. Untuk memperoleh hasil yang lebih akurat, pada penelitian ini digunakan tiga rasio perbandingan data latih dan data uji yaitu [11]:

1. 90% sebagai *data traning* dan 10% sebagai *data testing*.
2. 80% sebagai *data traning* dan 20% sebagai *data testing*.
3. 70% sebagai *data traning* dan 30% sebagai *data testing*.

Setelah dilakukan pembagian rasio *data traning* dan *data testing*, berikutnya mencari nilai *accuracy, precision, recall, f-measure* dari tiap rasio untuk menentukan rasio mana dengan nilai yang terbaik.

2.5 Evaluasi

Hasil evaluasi akan menunjukkan perbandingan performa dari *Naive Bayes* dan SVM dengan menggunakan empat metode ekstraksi fitur yang berbeda, yaitu TF-IDF, TF-IDF dengan N-gram, *CountVectorizer*, dan *CountVectorizer* dengan N-gram. Selain itu, hasil evaluasi juga akan melihat seberapa pengaruh penggunaan *Information Gain* dalam meningkatkan akurasi pada metode *Naive Bayes* dan SVM. Dengan demikian, penelitian ini dapat memberikan rekomendasi tentang metode klasifikasi dan ekstraksi fitur yang paling optimal untuk mengklasifikasikan data opini pengguna Twitter terhadap perkembangan AI.

3 Hasil dan Pembahasan

Penelitian ini akan menghasilkan model klasifikasi terhadap data opini pengguna Twitter tentang perkembangan *artificial intelligence* di Indonesia, menggunakan algoritma *Naive Bayes* dan *Support Vector Machine* dengan empat metode ekstraksi fitur. Penelitian ini diharapkan dapat memberikan rekomendasi tentang metode klasifikasi dan ekstraksi fitur mana yang paling optimal untuk mengklasifikasikan data opini pengguna Twitter terhadap perkembangan *artificial intelligence.*

3.1 Seleksi Data

Dalam menjalankan proses pengumpulan data, opini-opini yang dibutuhkan diambil menggunakan alat Twint dengan teknik *scraping* yang diprogram menggunakan bahasa pemrograman Python. Data-data yang berhasil dikumpulkan kemudian disimpan dalam format CSV. Rentang waktu pengambilan data adalah dari tanggal 1 November 2022 sampai 3 Mei 2023 dengan total jumlah data mentah sebanyak 2.579 data. Data yang telah dikumpulkan tersebut, kemudian akan melalui tahap seleksi untuk diklasifikasikan menjadi sentimen positif dan negatif menggunakan aplikasi ChatGPT dan BingChat. Kedua aplikasi tersebut dibangun dengan menggunakan model GPT, namun BingChat menunjukkan akurasi yang lebih baik dibandingkan dengan ChatGPT. Berdasarkan hasil pelabelan menggunakan model GPT, hanya 1.375 data saja yang berhasil dilabeli, terdiri dari 536 data dengan sentimen negatif dan 839 data dengan sentimen positif. Adapun sisa datanya nantinya tidak akan digunakan dan akan dihapus. Hal itu dikarenakan adanya kesalahan seperti kalimat yang tidak lengkap, menggunakan bahasa daerah, atau ChatGPT dan BingChat tidak dapat memahami kalimat tersebut.

3.2 Persiapan Data

Setelah data dilabeli, berikutnya dilakukan tahap *pre-processing* yang bertujuan untuk menghilangkan kata yang tidak diperlukan dalam proses klasifikasi.

1. *Case Folding:* tahap dimana data tweet diubah dengan cara mengubah huruf kapital menjadi huruf kecil.
2. *Cleaning:* tahap dimana data yang berupa tweet akan dibersihkan dengan cara menghilangkan kata tidak penting seperti *hashtag*, *usertag*, *url*, *emoticon*, angka, dan simbol-simbol yang tidak diperlukan pada proses data mining.
3. *Normalization*: tahap dimana dilakukan perbaikan terhadap kalimat-kalimat yang mengandung kata-kata tidak baku, singkatan, dan kesalahan ejaan. Proses ini dilakukan dengan pengkodean menggunakan Python. Untuk memperbaiki kata tersebut, digunakan juga daftar koreksi kata yang sudah dibuat berdasarkan kata-kata yang salah dalam dataset.
4. *Tokenizing*: tahap dimana data *tweet* akan dipecah menjadi kata per kata.
5. *Stemming*: tahap dimana setiap kalimat akan dihilangkan imbuhannya. Pada proses ini menggunakan modul sastrawi untuk menggunakan *Stemmer Factory*. Sebagai contoh, kata “berkembang” akan diubah menjadi “kembang”.
6. *Stopwords Removal*: tahap dimana kata-kata yang tidak bernilai informatif dalam proses *data mining* nantinya akan dihapus. Kata-kata yang dihapus dapat berupa kata sambung, sebagai contoh, kata “dan”, “dengan”, dan “atau” akan dihapus.

3.3 Transformasi Data

Pada tahap transformasi data, digunakan ekstraksi fitur *CountVectorizer* dan TF-IDF untuk ekstraksi fitur, dan *Information Gain* untuk seleksi fitur. Penelitian ini akan menerapkan juga N-gram pada kedua ekstraksi fitur dengan model *unigram* dan *bigram*. Selanjutnya, dilakukan ekstraksi fitur dan seleksi fitur pada masing-masing kombinasi percobaan. Percobaan dilakukan dengan menerapkan empat metode ekstraksi fitur, yaitu TF-IDF, TF-IDF dengan N-gram, *CountVectorizer*, dan *CountVectorizer* dengan N-gram. Percobaan juga dilakukan dengan menggunakan seleksi fitur *Information Gain* untuk mengurangi dimensi fitur. Dengan demikian, terdapat 8 kombinasi metode ekstraksi dan seleksi fitur yang digunakan dalam penelitian ini, seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Kombinasi Percobaan *Data Mining*

Percobaan	TF-IDF	*CountVectorizer*	N-gram	*Information Gain*
1	✓			
2		✓		
3	✓		✓	
4		✓	✓	
5	✓			✓
6		✓		✓
7	✓		✓	✓
8		✓	✓	✓

Setelah dilakukan pemrosesan menggunakan metode TF-IDF dan *CountVectorizer*, maka diperoleh sebuah matriks yang terdiri dari 1375 dokumen dan 4862 kata unik dengan bobot yang sesuai. Sedangkan hasil dari proses menggunakan N-gram menghasilkan sebuah matriks yang terdiri dari 1375 dokumen dan 23.542 kombinasi unik dari kata-kata dalam bentuk 1 dan 2 kata, serta bobot dari ekstraksi fitur tersebut.

Selanjutnya, hasil-hasil tersebut nantinya akan diproses menggunakan seleksi fitur *Information Gain* untuk mengurangi fitur yang tidak diperlukan dengan cara mencari nilai *k* (jumlah fitur) yang sesuai.

3.4 Data Mining

Setelah tahap transformasi, langkah selanjutnya dalam proses data mining adalah melakukan klasifikasi menggunakan metode *Naive Bayes* dan *Support Vector Machine* (SVM). Sebelum dilakukan klasifikasi, data yang telah diubah bobotnya akan dibagi menjadi *data training* dan *data testing*. Pembagian data dilakukan dengan rasio 90:10, 80:20, dan 70:30.

Selain itu, data tersebut juga akan diuji menggunakan metode validasi silang 10-*fold cross-validation*.

Pada percobaan pertama hingga keempat dilakukan tanpa menggunakan seleksi fitur *Information Gain*. Data tersebut akan diuji pada masing-masing rasio pembagian data uji dan latih. Kemudian hasil dari percobaan tersebut akan dievaluasi berdasarkan tingkat akurasi dan disajikan dalam bentuk tabel, seperti dapat dilihat pada Tabel 2.

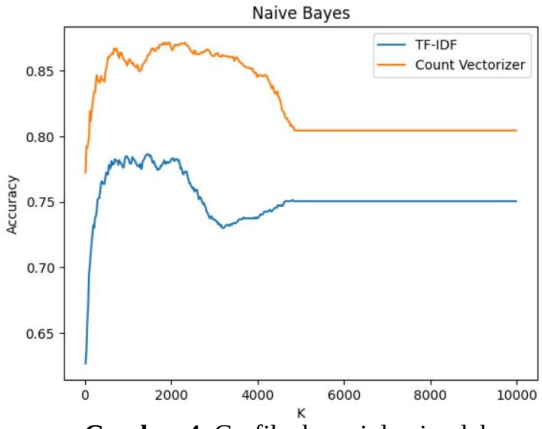
Tabel 2. Tabel hasil percobaan 1-4

Metode	Ekstraksi Fitur	Rasio			Rata-rata
		70:30	80:20	90:10	
Naive Bayes	TF-IDF	0.748	0.80	0.797	0.782
SVM	TF-IDF	0.857	0.857	0.862	0.859
Naive Bayes	CV	0.818	**0.833**	0.804	0.818
SVM	CV	0.792	0.804	0.768	0.788
Naive Bayes	TF-IDF (N-gram)	**0.685**	**0.742**	**0.783**	0.736
SVM	TF-IDF (N-gram)	0.835	0.865	**0.877**	0.859
Naive Bayes	CV (N-gram)	**0.818**	**0.811**	**0.732**	0.787
SVM	CV (N-gram)	0.838	0.836	0.804	0.826

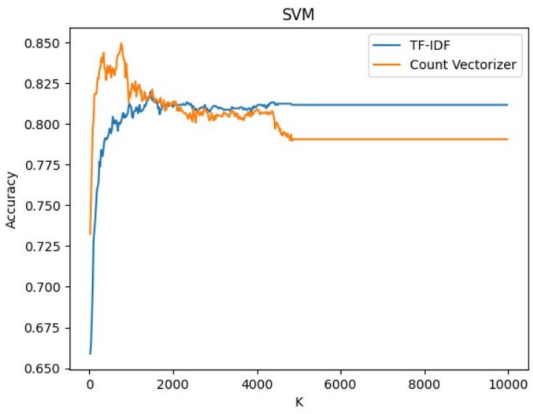
Hasil yang tertera pada Tabel 2 menunjukkan bahwa akurasi TF-IDF saat menggunakan N-gram cenderung meningkat seiring dengan jumlah data latih yang digunakan. Namun, pada *CountVectorizer* dengan N-gram, akurasi justru menurun seiring dengan peningkatan jumlah data latih. Pada data tersebut akurasi tertinggi tercapai saat menggunakan SVM yaitu **0.877** yang menggunakan TF-IDF dan N-gram, sedangkan akurasi tertinggi untuk *Naive Bayes* yaitu **0.833** yang tercapai saat menggunakan *CountVectorizer*. Dari Tabel 2 juga menunjukkan bahwa model hasil *Naive Bayes* lebih optimal ketika menggunakan *CountVectorizer*, sedangkan model hasil SVM lebih optimal ketika menggunakan TF-IDF.

Selanjutnya percobaan kelima sampai kedelapan dilakukan dengan menerapkan *Information Gain* sebagai seleksi fitur. Penentuan jumlah fitur (*k*) yang dipakai, akan dilakukan pada percobaan kali ini. Langkah pertama yaitu melakukan pembobotan dengan menggunakan *Information Gain* untuk menentukan nilai bobot kepentingan kata. Selanjutnya hasil pembobotan diurutkan dan kemudian jumlah fitur ditentukan dari nilai bobot terbesar ke yang terkecil. Jumlah fitur (*k*) ini ditentukan dengan cara membuat grafik akurasi berdasarkan jumlah fitur yang dipakai. Pada penelitian ini, grafik dicoba mulai dengan jumlah fitur *k*=20 sampai *k*=10.000, yang mana memiliki jarak 20 untuk setiap langkahnya. Kemudian berdasarkan grafik tersebut nantinya akan dipilih nilai *k* dengan akurasi yang paling tinggi.

Langkah kesatu yaitu mencari nilai *k* untuk metode *Naive Bayes* dengan TF-IDF dan *CountVectorizer* sebagai ekstraksi fitur, dan tanpa menerapkan N-gram. Karena semua dataset digunakan, dalam arti tidak ada penentuan rasio *data training* dan *data testing*, maka teknik 10-*fold cross validation* akan digunakan dan diperoleh grafik seperti yang dapat dilihat pada Gambar 4. Berdasakan grafik tersebut memperlihatkan bahwa pada TF-IDF akurasi tertinggi diperoleh saat *k*=1440 dengan nilai akurasi sebesar 0.786, sedangkan pada CV akurasi tertinggi diperoleh saat *k*=1820 dengan nilai akurasi sebesar 0.871. Langkah kedua yaitu mencari nilai *k* untuk metode SVM dengan TF-IDF dan *CountVectorizer* sebagai ekstraksi fitur, dan tanpa menerapkan N-gram. Dengan cara yang sama menggunakan 10-*fold cross validation* diperoleh grafik yang dapat dilihat pada Gambar 5. Berdasakan grafik tersebut memperlihatkan bahwa pada TF-IDF akurasi tertinggi diperoleh saat *k*=1460 dengan nilai akurasi sebesar 0.820, sedangkan pada CV akurasi tertinggi diperoleh saat *k*=760 dengan nilai akurasi sebesar 0.849.



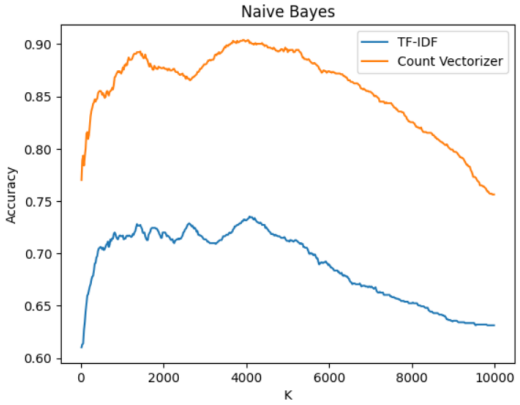
Gambar 4. Grafik akurasi dan jumlah fitur *Naive Bayes*



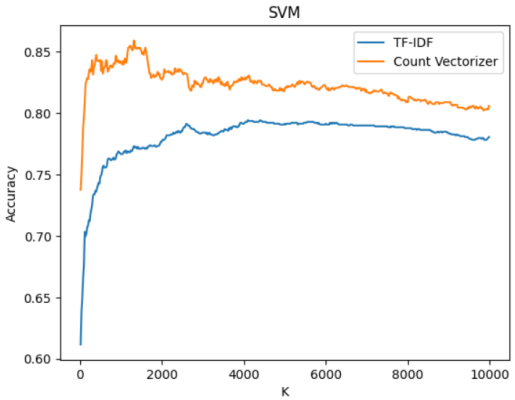
Gambar 5. Grafik akurasi dan jumlah fitur SVM

Langkah ketiga yaitu mencari nilai *k* untuk *Naive Bayes* dengan TF-IDF dan *CountVectorizer* sebagai ekstraksi fitur, dan dengan menerapkan N-gram. Dengan cara yang sama menggunakan 10-*fold cross validation* diperoleh grafik yang dapat dilihat pada Gambar 6. Berdasakan grafik tersebut memperlihatkan bahwa pada TF-IDF akurasi tertinggi diperoleh saat *k*=4080 dengan nilai akurasi sebesar 0.735, sedangkan pada CV akurasi tertinggi diperoleh saat *k*=3920 dengan nilai akurasi sebesar 0.904. Langkah keempat yaitu mencari nilai *k* untuk SVM dengan TF-IDF dan *CountVectorizer* sebagai ekstraksi fitur, dan dengan menerapkan N-gram. Dengan cara yang sama menggunakan 10-*fold cross validation* diperoleh

grafik yang dapat dilihat pada Gambar 7. Berdasakan grafik tersebut memperlihatkan bahwa pada TF-IDF akurasi tertinggi diperoleh saat *k*=4100 dengan nilai akurasi sebesar 0.794, sedangkan pada CV akurasi tertinggi diperoleh saat *k*=1320 dengan nilai akurasi sebesar 0.859.



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



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

Berdasarkan data yang diproses tersebut, maka diperoleh nilai *k* untuk masing-masing kombinasi, yang selanjutnya akan dibut tabel nilai akurasi berdasarkan nilai *k* yang sudah diperoleh tersebut. Tabel itu nantinya akan digunakan untuk menentukan mana rasio terbaik seperti yang disajikan pada Tabel 3 berikut.

Tabel 3. Tabel hasil percobaan 5-8 dengan *Information Gain*

Metode	Ekstraksi Fitur	IG Nilai *k*	Rasio			Rata-rata
			70:30	80:20	90:10	
Naive Bayes	TF-IDF	1440	0.797	0.825	0.818	0.814
SVM	TF-IDF	1460	0.845	0.862	**0.869**	0.859
Naive Bayes	CV	1820	0.891	0.916	0.913	0.907
SVM	CV	760	0.826	0.869	0.826	0.840
Naive Bayes	TF-IDF (N-gram)	4080	0.717	0.774	0.797	0.763
SVM	TF-IDF (N-gram)	4100	0.814	0.851	**0.869**	0.845
Naive Bayes	CV (N-gram)	3920	0.896	**0.931**	0.920	0.916
SVM	CV (N-gram)	1320	0.859	0.854	0.848	0.854

Berdasarkan hasil pada Tabel 3, dapat dilihat bahwa seleksi fitur *Information Gain* cukup baik dalam meningkatkan nilai akurasi dari metode *Naive Bayes* dan SVM dibandingkan saat tidak menerapkan seleksi fitur. Berdasarkan data pada Tabel 3, diperoleh akurasi tertinggi dari metode *Naive Bayes* adalah **0.931** menggunakan *CountVectorizer* dengan N-gram, sedangkan akurasi tertinggi dari metode SVM adalah **0.869** menggunakan TF-IDF dengan atau tanpa N-gram. Selain itu penerapan *Information Gain* juga ternyata cukup berpengaruh pada metode *Naive Bayes*, hal ini dapat dilihat dengan adanya peningkatan nilai akurasi sebanyak 12% saat menggunakan *CountVectorizer*

dengan N-gram. Dari Tabel 3 tersebut juga menunjukkan bahwa rasio 80:20 cukup baik dibandingkan dengan rasio yang lain.

3.5 Evaluasi

Tahap setelah dilakukan *data mining* adalah tahap *evaluation* untuk mengukur kinerja dari metode *Naive Bayes* dan *Support Vector Machine* dengan menerapkan ekstraksi fitur TF-IDF dan *CountVectorizer* serta seleksi fitur *Information Gain*. Dapat dilihat pada Tabel 4 berikut merupakan nilai akurasi dari hasil *data mining* dengan metode *Naive Bayes* dan SVM.

Tabel 4. Rekapan Nilai Akurasi Metode *Naive Bayes* dan SVM

Metode	Ekstraksi Fitur	Seleksi Fitur	Rasio			Rata-Rata
			70:30	80:20	90:10	
Naive Bayes	TF-IDF	-	0.748	0.80	0.797	0.782
SVM	TF-IDF	-	0.857	0.857	0.862	0.859
Naive Bayes	CV	-	0.818	0.833	0.804	0.819
SVM	CV	-	0.792	0.804	0.768	0.788
Naive Bayes	TF-IDF (N-gram)	-	0.685	0.742	0.783	0.736
SVM	TF-IDF (N-gram)	-	0.835	0.865	**0.877**	0.859
Naive Bayes	CV (N-gram)	-	0.818	0.811	0.732	0.787
SVM	CV (N-gram)	-	0.838	0.836	0.804	0.826
Naive Bayes	TF-IDF	IG	0.797	0.825	0.819	0.814
SVM	TF-IDF	IG	0.845	0.862	0.869	0.859
Naive Bayes	CV	IG	0.891	0.916	0.913	0.907
SVM	CV	IG	0.826	0.869	0.826	0.840
Naive Bayes	TF-IDF (N-gram)	IG	0.717	0.775	0.797	0.763
SVM	TF-IDF (N-gram)	IG	0.814	0.851	0.869	0.845
Naive Bayes	CV (N-gram)	IG	0.896	**0.931**	0.920	0.916
SVM	CV (N-gram)	IG	0.859	0.855	0.848	0.854

Dari tabel tersebut, terlihat bahwa *Naive Bayes* dengan *Information Gain* memiliki peningkatan akurasi yang signifikan daripada *Naive Bayes* tanpa *Information Gain*. SVM dengan *Information Gain* juga memiliki peningkatan akurasi, tetapi tidak sebesar *Naive Bayes*. Dari tabel tersebut juga terlihat bahwa *Naive Bayes* dengan *CountVectorizer* memiliki performa yang lebih baik daripada *Naive Bayes* dengan TF-IDF, sedangkan SVM dengan TF-IDF memiliki performa yang lebih baik daripada SVM dengan *CountVectorizer* jika tidak menerapkan seleksi fitur *Information Gain*. Penerapan N-gram juga ternyata dapat meningkatkan nilai akurasi untuk *CountVectorizer*, tetapi sebaliknya dapat menurunkan nilai akurasi untuk TF-IDF jika menggunakan algoritma *Naive Bayes*. Akurasi tertinggi tanpa seleksi fitur *Information Gain* dicapai oleh SVM dengan TF-IDF dengan N-gram sebesar **0.877**, sedangkan akurasi tertinggi dari seleksi fitur *Information Gain* dicapai oleh *Naive Bayes* menggunakan *CountVectorizer* dan N-gram, yaitu **0.931**. Hasil penelitian juga menunjukkan bahwa rasio pembagian data 80:20 dan 90:10 memberikan hasil akurasi yang cukup baik dibandingkan rasio pembagian data 70:30.

4 Kesimpulan

Berbeda dengan penelitian sebelumnya [12] yang hanya menerapkan metode *Naive Bayes* untuk melihat hasil sentimen pengguna twitter terhadap tools *artificial intelegence*, diperoleh nilai akurasi sebesar 79,4%. Maka dalam penelitian ini, dilakukan perbandingan kinerja antara *Naive Bayes* dan *Support Vector Machine* (SVM) dengan menerapkan ekstraksi fitur dan seleksi fitur dalam melakukan analisis sentimen. Hasil percobaan memperlihatkan bahwa metode SVM memiliki performa yang lebih baik dibandingkan dengan metode *Naive Bayes*. Nilai akurasi paling tinggi yang dihasilkan oleh SVM pada klasifikasi dua kelas sentimen sebesar 87.68% dengan menggunakan metode ekstraksi fitur TF-IDF yang menerapkan N-gram. Sementara itu, *Naive Bayes* mencapai nilai akurasi tertinggi sebesar 83.27% dengan menggunakan metode ekstraksi fitur *CountVectorizer*. Penggunaan seleksi fitur *Information Gain* juga mampu meningkatkan nilai akurasi pada model klasifikasi hasil metode *Naive Bayes* dan SVM. Pada metode *Naive Bayes*, seleksi fitur *Information Gain* mampu meningkatkan akurasi sampai 12%. Nilai akurasi pada *Naive Bayes* meningkat dari 0.811 menjadi 0.931 saat menggunakan ekstraksi fitur *CountVectorizer* dengan menerapkan N-gram. Akan tetapi nilai akurasi pada SVM menurun dari 0.877 menjadi 0.869 saat menggunakan ekstraksi fitur TF-IDF dengan N-gram. Ekstraksi fitur TF-IDF dan *CountVectorizer* sangat mempengaruhi proses klasifikasi. Perbedaan terbesar dari kedua ekstraksi fitur tersebut adalah *CountVectorizer* memperhatikan urutan kata sedangkan TF-IDF tidak. Berdasarkan penelitian ini dihasilkan bahwa metode *Naive Bayes* mendapatkan akurasi tertingginya saat menggunakan ekstraksi fitur *CountVectorizer*, sedangkan SVM mendapatkan akurasi tertingginya saat menggunakan TF-IDF. Pada saat menggunakan TF-IDF, performa dari N-gram justru menurunkan nilai akurasi dari model klasifikasi. Sedangkan pada *CountVectorizer* justru meningkatkan nilai akurasi dari model klasifikasi. Dengan demikian dapat disimpulkan bahwa N-gram lebih cocok jika digunakan pada *CountVectorizer* dibandingkan pada TF-IDF.

5 Referensi

[1] I. R. Dewi, “Ahli Warning AI Bisa Picu Bencana Setara Perang Nuklir. CNBC Indonesia,” 2022. https://www.cnbcindonesia.com/tech/20221007074709-37-377902/ahli-warning-ai-bisa-picu-bencana-setara-perang-nuklir (accessed Mar. 11, 2023).

[2] Romanti, “Artificial Intelligence (AI): Bahaya atau Dukungan untuk Pekerjaan Manusia?,” 2023. https://itjen.kemdikbud.go.id/web/artificial-intelligence-ai-bahaya-atau-dukungan-untuk-pekerjaan-manusia/ (accessed Apr. 01, 2024).

[3] S. A. Putra and A. Wijaya, “Analisis Sentimen Artificial Intelligence (Ai) Pada Media Sosial Twitter Menggunakan Metode Lexicon Based,” *JuSiTik J. Sist. dan Teknol. Inf. Komun.*, vol. 7, no. 1, pp. 21–28, 2023, doi:

10.32524/jusitik.v7i1.1042.

[4] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 1, p. 147, Feb. 2021, doi: 10.25126/jtiik.0813944.

[5] J. E. Simarmata, G.-W. Weber, and D. Chrisinta, "Performance Evaluation of Classification Methods on Big Data: Decision Trees, Naive Bayes, K-Nearest Neighbors, and Support Vector Machines," *J. Mat. Stat. Dan Komputasi*, vol. 20, no. 3, pp. 623–638, 2024, doi: 10.20956/j.v20i3.32970.

[6] A. Zy and Wahyu Hadikristanto, "Implementasi Algoritma Metode Naive Bayes dan Support Vector Machine Tentang Pembobolan dan Kebocoran Data di Twitter," *Bull. Inf. Technol.*, vol. 4, no. 1, pp. 49–56, 2023, doi: 10.47065/bit.v4i1.493.

[7] Sharazita Dyah Anggita and Ferian Fauzi Abdulloh, "Optimasi Algoritma Support Vector Machine Berbasis PSO Dan Seleksi Fitur Information Gain Pada Analisis Sentimen," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 52–57, 2023, doi: 10.52158/jacost.v4i1.524.

[8] F. L. Hakim and K. E. Dewi, "KOMPUTA : Jurnal Ilmiah Komputer dan Informatika PRODUK KECANTIKAN MENGGUNAKAN NEIGHBOR WEIGHTED K-NEAREST NEIGHBOR KOMPUTA : Jurnal Ilmiah Komputer dan Informatika," vol. 13, no. 1, pp. 1–10, 2024.

[9] A. Averina, H. Hadi, and J. Siswantoro, "Analisis Sentimen Multi-Kelas Untuk Film Berbasis Teks Ulasan Menggunakan Model Regresi Logistik," *Teknika*, vol. 11, no. 2, pp. 123–128, 2022, doi: 10.34148/teknika.v11i2.461.

[10] S. K. Dirjen, P. Riset, D. Pengembangan, R. Dikti, S. Khomsah, and A. S. Aribowo, "Terakreditasi SINTA Peringkat 2 Model Text-Preprocessing Komentar Youtube Dalam Bahasa Indonesia," *Masa Berlaku Mulai*, vol. 1, no. 3, pp. 648–654, 2017.

[11] C. B. Vista, O. M. Sihono, and A. T. Firdausi, "Analisis Sentimen Kebijakan Pembelajaran Tatap Muka Selama Pandemi Covid-19 Menggunakan Metode Support Vector Machine," *J. Inform. Polinema*, vol. 9, no. 3, pp. 259–264, 2023, doi: 10.33795/jip.v9i3.1273.

[12] I. Oktavia and A. R. Isnain, "Analisis Sentimen Opini Terhadap Tools Artificial Intelligence (AI) Berdasarkan Twitter Menggunakan Algoritma Naïve Bayes," *J. Media Inform. Budidarma* , vol. 8, no. 2, pp. 777–787, 2024, doi: 10.30865/mib.v8i2.7524.