

## PENERAPAN *TEXT MINING* DAN ANALISIS SENTIMEN PADA KOMENTAR *YOUTUBE* “*DIRTY VOTE*” MENGGUNAKAN ALGORITMA KLASIFIKASI

Ahmad Farul Azis<sup>1\*</sup>, Mardi Hardjianto<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: <sup>1\*</sup>ahmadfahrula779@gmail.com, <sup>2</sup>mardi.hardjianto@budiluhur.ac.id  
(\* : corresponding author)

**Abstrak-**Kecurangan dalam pemilu merupakan salah satu masalah yang ada dalam konteks demokrasi karena dapat mengancam integritas proses demokratis dan kedaulatan rakyat. Jenis kecurangan seperti politik uang, politik identitas, intimidasi pemilih, dan manipulasi hasil pemilu dapat merusak kepercayaan masyarakat dan demokrasi secara keseluruhan. Isu kecurangan ini sering kali menjadi perbincangan hangat di masyarakat dan memicu perdebatan yang intens. Pada Februari 2024 lalu terdapat sebuah *film* yang berjudul “*Dirty Vote*” yang menjadi perbincangan hangat masyarakat di berbagai platform media sosial terutama *YouTube*. Komentar terhadap film tersebut berisi pro dan kontra dari masyarakat. Namun, dengan banyaknya komentar masyarakat dan tidak memungkinkan untuk menghitung satu persatu maka diperlukan analisis sentimen menggunakan pendekatan *text mining*. Tujuan penelitian ini untuk memahami sentimen masyarakat dalam memperoleh gambaran yang jelas berupa positif, negatif atau netral terhadap isu yang diangkat dalam film *Dirty Vote* menggunakan metode *naïve bayes*, K-NN dan *decision tree* sekaligus mengetahui algoritma manakah yang terbaik. Algoritma *naïve bayes* menghasilkan nilai akurasi sebesar 81% menggunakan pelabelan manual dan akurasi 70% menggunakan pelabelan *lexicon*. Algoritma K-NN menghasilkan akurasi 80% menggunakan pelabelan manual dan akurasi 61% menggunakan pelabelan *lexicon*. Algoritma *decision tree* menghasilkan akurasi 74% menggunakan pelabelan manual dan akurasi 68% menggunakan pelabelan *lexicon*. Sehingga dari ketiga pengujian tersebut algoritma *naïve bayes* dengan pelabelan manual menghasilkan tingkat akurasi sebesar 81% dengan presisi sebesar 81%, *recall* sebesar 100% dan *F1-Score* sebesar 90% menjadi algoritma terbaik pada penelitian ini.

**Kata Kunci:** Kecurangan pemilu, *YouTube*, analisis sentimen, *Text Mining*, *Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree*

## APPLICATION OF TEXT MINING AND SENTIMENT ANALYSIS ON YOUTUBE COMMENT “DIRTY VOTE” USING CLASSIFICATION ALGORITHM

**Abstract-** Electoral fraud is one of the problems that exist in the context of democracy because it can threaten the integrity of the democratic process and popular sovereignty. Types of fraud such as money politics, identity politics, voter intimidation and manipulation of election results can undermine public trust and democracy as a whole. The issue of fraud often becomes a hot topic of discussion in society and triggers intense debate. In February 2024, there was a movie entitled “*Dirty Vote*” which became a hot topic of public discussion on various social media platforms, especially *YouTube*. Comments on the movie contained pros and cons from the public. However, with so many public comments and it is not possible to count them one by one, sentiment analysis using a *Text Mining* approach is needed. The purpose of this research is to understand public sentiment in obtaining a clear picture in the form of positive, negative or neutral towards the issues raised in the *Dirty Vote* movie using the *Naïve Bayes*, K-NN and *Decision Tree* methods while knowing which algorithm is the best. The *Naïve Bayes* algorithm produces an accuracy value of 81% using manual labeling and 70% accuracy using *lexicon* labeling. K-NN algorithm produces 80% accuracy using manual labeling and 61% accuracy using *lexicon* labeling. *Decision Tree* algorithm produces 74% accuracy using manual labeling and 68% accuracy using *lexicon* labeling. So from the three tests, the *Naïve Bayes* algorithm with manual labeling produces an accuracy rate of 81% with 81% precision, 100% recall and *F1-Score* of 90% being the best algorithm in this study.

**Keywords:** Election fraud, *YouTube*, sentiment analysis, *Text Mining*, *Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree*

### 1. PENDAHULUAN

Kecurangan dalam pemilu merupakan salah satu masalah yang ada dalam konteks demokrasi karena dapat mengancam integritas proses demokratis dan kedaulatan rakyat. Jenis kecurangan seperti politik uang, politik identitas, intimidasi pemilih, dan manipulasi hasil pemilu dapat merusak kepercayaan masyarakat dan demokrasi secara keseluruhan. Hal ini juga dapat menyebabkan pemerintahan yang kurang representatif dan demokratis. Isu

kecurangan ini sering kali menjadi perbincangan hangat di masyarakat dan memicu perdebatan yang *intens*. Pada Februari 2024 lalu terdapat sebuah film yang berjudul “*Dirty Vote*” yang menjadi perbincangan hangat masyarakat di berbagai platform media sosial. “*Dirty Vote*” adalah sebuah film dokumenter politik yang menggambarkan kisah tentang kecurangan dalam pemilihan umum pada tahun 2024. Film ini menggambarkan bagaimana politik dapat menjadi kotor dan penuh intrik, serta bagaimana kekuasaan dapat dimanfaatkan untuk kepentingan pribadi. Setiap komentar yang terdapat pada video tersebut mencerminkan bagaimana pendapat serta penilaian setiap individu masyarakat terhadap film “*Dirty Vote*”.

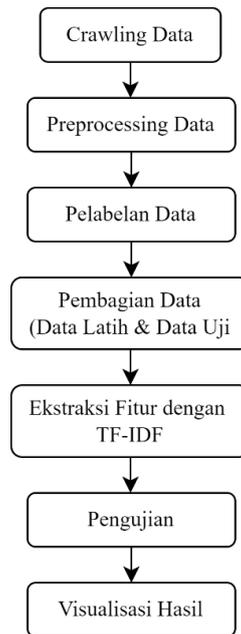
Media sosial, seperti *YouTube*, adalah *platform* penting bagi masyarakat untuk berbagi kritik, ide inovatif, dan pendapat tentang berbagai isu seperti ketidakadilan, politik, ekonomi, dan lainnya. *YouTube*, sebagai salah satu *platform* terbesar, memungkinkan pengguna untuk berinteraksi melalui komentar yang mencerminkan pandangan mereka terhadap konten *video*. Analisis sentimen, yang memproses teks digital untuk mengelompokkan emosi menjadi positif, negatif, dan netral, menjadi alat penting untuk memahami reaksi pengguna. Proses ini melibatkan berbagai langkah mulai dari pemrosesan teks hingga pengklasifikasian sentimen, dan berbagai algoritma *text mining* dapat digunakan, masing-masing dengan kelebihan dan kekurangannya. Pemilihan algoritma yang tepat sangat penting untuk mendapatkan hasil yang akurat dan informatif, karena setiap algoritma memiliki tingkat akurasi yang berbeda tergantung pada jenis data dan kompleksitas tugas. Dalam *machine learning*, algoritma dievaluasi berdasarkan metrik seperti akurasi, presisi, *recall*, dan skor F1 untuk menentukan efektivitasnya dalam analisis sentimen pada komentar *video YouTube*.

Penelitian sebelumnya yang dilakukan pada [1] melakukan perbandingan algoritma *naïve bayes* dan *SVM* dengan hasil akurasi tertinggi oleh algoritma *SVM* dengan nilai akurasi sebesar 86,82%. Penelitian yang dilakukan oleh [2] melakukan analisis perbandingan algoritma yang sama berdasarkan data yang diambil dari *Twitter* dengan hasil akurasi tertinggi yang didapatkan oleh *SVM* sebesar 88,80%. Penelitian yang dilakukan oleh [3] melakukan analisis perbandingan *naïve bayes* dan K-NN dengan hasil akurasi yang tidak terlalu berbeda jauh namun K-NN memiliki hasil tertinggi sebesar 85,00%. Penelitian lainnya yang dilakukan oleh [4] dalam analisis komentar masyarakat terhadap produk Nokia menggunakan perbandingan algoritma *decision tree* dan *random forest*. Algoritma *decision tree* memiliki akurasi yang sedikit lebih tinggi yaitu 89,40% dibandingkan dengan algoritma *random forest* yaitu 88,20%.

Penelitian ini memanfaatkan media sosial *YouTube* sebagai acuan sumber data yang akan diteliti. Data yang diambil berdasarkan komentar pada *video YouTube “Dirty Vote”*. Data tersebut diproses menggunakan teknik *text mining* untuk menganalisis sentimen menggunakan beberapa algoritma. Berdasarkan dari penelitian sebelumnya, perbandingan antar algoritma masih diperlukan untuk memberikan gambaran yang lebih komprehensif tentang kinerja masing-masing metode dalam berbagai kondisi dan jenis data. Oleh karena itu, penelitian ini dilakukan untuk mengatasi keterbatasan dalam penelitian sebelumnya yang hanya menguji menggunakan satu algoritma saja atau melakukan perbandingan dua algoritma saja. Penelitian ini akan memperluas analisis dengan membandingkan tiga algoritma klasifikasi berbeda, yaitu *naïve bayes*, K-NN, dan *decision tree* untuk mengidentifikasi algoritma yang paling optimal untuk klasifikasi data pada komentar *YouTube*. Pendekatan ini diharapkan dapat memberikan wawasan lebih mendalam mengenai keunggulan dan kelemahan relatif dari setiap algoritma, serta mencari algoritma yang terbaik berdasarkan nilai akurasi yang tertinggi.

## 2. METODE PENELITIAN

Metode penelitian merupakan tahapan krusial yang menjelaskan secara rinci bagaimana metode yang telah dipilih akan diimplementasikan. Dalam penelitian ini untuk membangun aplikasi sentimen analisis terdapat beberapa langkah. Pada Gambar 1 berisi tahapan metodologi penelitian.



Gambar 1. Langkah Metodologi Penelitian

## 2.1 Crawling Data

Pada penelitian ini untuk tahap *Crawling* data menggunakan bahasa pemrograman *python*, ID *video* dan API *youtube* yang disediakan oleh *google* sebagai prasyarat untuk melakukan *crawling* data. Target *video* yang akan dijadikan sebagai dataset dalam penelitian ini adalah komentar pada *video* yang berjudul “*Dirty vote*”. Dataset yang digunakan pada penelitian ini didapatkan pada tanggal 11 Februari hingga 3 April 2024 yang berjumlah 2.436 komentar. Data tersebut akan disimpan kedalam berkas CSV dengan atribut *publisedAt*, *authorDisplayName* dan *textDisplay*.

## 2.2 Pre-processing Data

*Preprocessing* atau prapemrosesan adalah sebuah tahapan dalam mengolah teks yang tidak teratur atau tidak terstruktur untuk membuatnya lebih teratur. Tahap ini memungkinkan pengolahan teks untuk mengumpulkan informasi dan pola teks yang spesifik untuk dokumen tertentu [5]. Dalam *Preprocessing* memiliki beberapa langkah yang bertujuan untuk meningkatkan kualitas *teks* yang akan di proses. Langkah-langkah dalam melakukan *preprocessing* adalah:

- Cleansing*, Proses membersihkan data berbentuk teks seperti menghapus karakter dari data komentar, termasuk html tags, tanda baca (*punctuations*), sebagian besar angka (*numbers*), spasi berlebih, ruang kosong, dan data duplikat [6].
- Case Folding*, sebuah langkah untuk mengubah seluruh data berbentuk teks yang memiliki huruf besar dirubah menjadi huruf kecil secara keseluruhan [5]. Tahap *case folding* perlu dilakukan dalam *preprocessing* teks untuk memastikan konsistensi dalam analisis teks.
- Tokenizing*, sebuah proses untuk mengubah suatu kalimat yang memiliki beberapa kata menjadi sebuah kata sendiri secara terpisah. Misalnya pada kalimat “sedang mengerjakan skripsi” maka akan menjadi “sedang”, “mengerjakan”, “skripsi” dengan menggunakan metode yang bernama *tokenizing* [5].
- Normalisasi, sebuah proses mengubah teks pada sebuah dokumen dari kata yang tidak relevan atau kata singkatan seperti bahasa gaul (*slang word*) menjadi kata yang memiliki arti. Misalnya pada kata “dgn” menjadi “dengan”, “yg” menjadi “yang” dan lain sebagainya [5].
- StopWord*, proses untuk mengurangi beberapa kata yang dianggap tidak penting dalam dokumen dengan menghapus kata-kata yang kurang deskriptif atau kata penghubung. Contoh dari kata *stopword* adalah “dengan”, “dan”, “yang” dan sebagainya [5]. Proses *Stopword* dilakukan untuk menghapus kata dalam sebuah komentar yang memiliki kesamaan dalam kumpulan kata (*corpus*). Pada penelitian [7] membuat sebuah

dataset *corpus* untuk *stopword* yang berasal dari analisis frekuensi kata dalam Bahasa Indonesia. Percobaan dilakukan pada korpus Bahasa Indonesia yang dibangun dari berita utama harian *online* Kompas selama satu tahun (Januari 2001 hingga Desember 2001), dengan total 3160 dokumen. Korpus ini terdiri dari 50.000 kata unik setelah menghilangkan nama orang, kota, organisasi, negara, dll.

- f. *Stemming*, sebuah proses untuk mengubah setiap kata menjadi sebuah kata dasar dengan menghapus kata yang memiliki imbuhan. Contoh dari langkah *stemming* ini seperti “melakukan” menjadi “lakukan” dan sebagainya [5].

### 2.3 Pelabelan Data

Pelabelan data adalah proses memberikan label atau kategori pada teks berdasarkan karakteristik atau konten tertentu yang telah ditentukan sebelumnya. Proses ini mencakup perumusan kategori, persiapan data, pemilihan metode pelabelan (baik manual maupun otomatis), pelatihan model jika menggunakan pembelajaran mesin, evaluasi dan validasi model, dan penggunaan model untuk memberi label pada teks untuk mengklasifikasikan teks ke dalam kategori yang relevan. Analisis sentimen, deteksi spam, dan pengelompokan topik adalah beberapa contoh aplikasinya yang bertujuan untuk mengorganisasi dan menganalisis data teks secara efektif. Pada tahap ini untuk menentukan kelas menggunakan pelabelan secara manual dan berdasarkan kamus *lexicon* dari laman *GitHub* (<https://github.com/fajri91/InSet>) untuk mengklasifikasikan setiap kata dalam komentar sebagai positif, negatif, atau netral [6]. Pelabelan menggunakan kamus *lexicon* dilakukan dengan menghitung nilai *polarity score*. Nilai *polarity* didapatkan dengan menjumlahkan bobot setiap kata yang terdapat pada kamus *lexicon* positif dan *lexicon* negatif. *Polarity score* yang bernilai  $> 0$  maka memiliki sentimen positif,  $< 0$  memiliki sentimen negatif dan  $= 0$  memiliki sentimen netral [8].

### 2.4 Pembagian Data

Pembagian data yaitu proses membagi data menjadi dua bagian kelompok yaitu, data latih dan data uji. Data latih adalah data yang membantu membangun dan melatih model dalam pengklasifikasian, dan data uji adalah data yang akan diuji menggunakan pengklasifikasian yang telah dibuat sebelumnya dengan data latih [9]. Proses pembagian tersebut akan dibagi dengan skala 80% untuk data latih dan 20% untuk data uji.

### 2.5 Ekstraksi Fitur TF-IDF

TF-IDF merupakan singkatan dari *Term Frequency Inverse Document Frequency* adalah metode yang berguna untuk menghitung nilai bobot, atau berat, setiap kata yang ada dalam dokumen. Nilai berat menunjukkan betapa pentingnya sebuah kata dalam dokumen, dan semakin besar bobotnya, semakin penting peran kata tersebut. *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) adalah konsep perhitungan bobot yang digabungkan untuk membentuk TF-IDF [5]. TF adalah sebuah banyaknya kemunculan kata (*frequency*) pada setiap dokumen. Rumus TF dihitung dengan membagi berapa kali istilah tersebut muncul dalam dokumen ( $f_{t,d}$ ) dengan jumlah total istilah dalam dokumen. Sedangkan, IDF didefinisikan untuk mengukur kepentingan kata di seluruh kumpulan dokumen. Rumus IDF dihitung dengan membagi jumlah dokumen (N) dengan jumlah dokumen yang memuat istilah tersebut [10].

$$tf(t, d) = \frac{f_{t,d}}{F_{t,d}} \quad (1)$$

$$idf(t) = \log_{10} \left( \frac{N}{1 + n_t} \right) + 1 \quad (2)$$

$$tfidf(t, d) = tf_{(t,d)} * idf_{(t)} \quad (3)$$

### 2.6 Pengujian

Tahap pengujian merupakan proses kinerja model yang telah dibangun menggunakan algoritma tertentu dalam *text mining*. Pada penelitian pengujian menggunakan algoritma *naïve bayes*, K-NN, dan *decision tree*. Dalam tahap ini data yang telah diberi label (data *training*) akan dilakukan proses pelatihan menggunakan masing-masing algoritma dan kemudian diuji menggunakan data uji untuk mengukur akurasi, presisi, recall, dan metrik lainnya.

- a. Metode *naïve bayes* adalah metode statistik yang digunakan untuk mengestimasi probabilitas. Esensi metode ini terletak pada gagasan bahwa setiap atribut dalam *set* data berfungsi secara independen untuk

mempengaruhi peluang hasil klasifikasi. Metode ini menilai peluang setiap kelas berdasarkan distribusi atribut dalam sampel pelatihan, dan kemudian menggunakan peluang ini untuk mengklasifikasikan entri baru dalam set data uji. *Naïve bayes* menjadi salah satu metode yang paling disukai untuk klasifikasi teks karena kinerjanya yang memuaskan dalam kumpulan data yang luas dan dengan persiapan data awal yang cukup [11].

$$AcP(C_k|X) = \frac{P(X|C_k)P(C_k)}{P_X} \quad (4)$$

- b. Metode K-NN klasifikasikan objek berdasarkan data pelatihan dengan menghitung jarak terdekat dari keseluruhan data. Pendekatan ini memanfaatkan prinsip bahwa dokumen yang serupa cenderung berada dalam kelompok yang sama. Metode K-NN menjalankan proses klasifikasi berdasarkan jumlah (K) paling dekat lalu diurutkan nilai yang didapatkan pada setiap objek atau data. Perhitungan melibatkan penjumlahan semua nilai kesamaan untuk satu kategori, yang akan dibandingkan dengan nilai mana pun yang lebih tinggi [3].

$$D_{(a,b)} = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (5)$$

- c. Metode *decision tree* salah satu metode pengolahan data yang menggunakan klasifikasi atau regresi model untuk membuat prediksi masa depan. Di mana setiap simpul internal menunjukkan tes atribut, masing-masing cabang menunjukkan hasilnya, dan setiap node memiliki label kelas. Simpul akar berada di node tertinggi di pohon keputusan [12]. Langkah untuk menentukan *node* akar dengan menghitung *gini impurity* dari masing-masing term pada kelas sentimen. Selanjutnya menghitung *average gini impurity* dan menghitung informasi gain untuk menentukan nilai terbesar atau terbanyak pada kelas sentimen [13].

$$gini\ impurity = 1 - \sum_{i=1}^n (p_i^2) \quad (6)$$

$$avg\ gini = \sum_i \frac{|n_i|}{|N|} * gini\ impurity\ (n) \quad (7)$$

$$Information\ Gain = Gini\ Impurity - Average \quad (8)$$

## 2.7 Evaluasi

Tahap evaluasi menggunakan *confusion matrix* atau matriks kebingungan adalah pengukuran performa kinerja untuk proses klasifikasi dalam pembelajaran mesin yang menghasilkan dua kelas atau lebih. metode terdiri dari tabel yang mengandung empat kombinasi nilai aktual dan prediksi. Metode ini memberikan hasil evaluasi berupa laporan klasifikasi (*classification report*). Laporan klasifikasi menghasilkan berbagai metrik kinerja seperti akurasi, presisi, recall, dan F1-score untuk masing-masing kelas. Serta memberikan wawasan mendalam tentang di mana model melakukan kesalahan. Rumus untuk menghitung nilai dari *confusion matrix* yaitu:

- a. *Accuracy* menunjukkan seberapa akurat model dalam mengklasifikasikan dengan benar baik negatif atau positif.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (9)$$

- b. *Precision* menunjukkan hasil perhitungan tingkat ketepatan antara jawaban model dan informasi yang diminta.

$$Precision = \frac{TP}{(TP + FP)} \quad (10)$$

- c. *Recall* (sensitifitas) menunjukkan hasil perhitungan untuk tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan oleh model.

$$Recall = \frac{TP}{(TP + FN)} \quad (11)$$

- d. *F-1 Score* menunjukkan perbandingan rata-rata precision dan *recall* yang dibobotkan.

$$F1\ Score = \frac{(2 * Recall * Precision)}{(Recall + Precision)} \quad (12)$$



kamus *Github*. Selanjutnya bobot dari setiap kata yang ditemukan dalam teks dijumlahkan untuk mendapatkan total nilai skor sentimen. Label kelas ditentukan berdasarkan hasil perhitungan skor bernilai lebih dari nol (0) maka akan dinilai positif. Pelabelan dinilai negatif jika total bobot nilainya kurang dari nol (0). Pelabelan dinilai netral jika total bobot nilainya sama dengan (0).

**Tabel 2.** Hasil Pelabelan Data

DataID	publishedAt	authorDisplayName	Teks bersih	Label Manual	Label <i>Lexicon</i>
1	April 19, 2024, 11:01 a.m.	@marciamailoa	terima kasih berani zainal arifin mochtar bivitri susanti feri amsari tim libat tayang dokumenter <i>dirty vote</i> edukatif informatif terima kasih tuhan yesus kati tolong negeri cinta	Positif	Positif
2	March 24, 2024, 9:11 p.m.	@LegendaryShorts37	pilih jokowi mari doa tim <i>dirty vote</i> sehat rezeki lancar	Positif	Positif

### 3.4 Tahap Pembagian Data

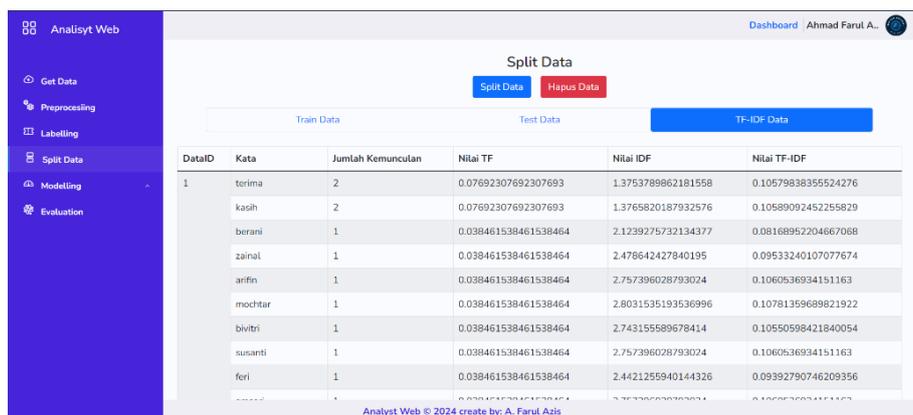
Tahap pembagian data dilakukan dari dataset yang sudah dilakukan labelling. Pembagian data dilakukan dengan skala 80% data latih dan 20% data uji sehingga dari 2.145 total data dalam dataset setelah dilakukan *preprocessing*, didapatkan 1.716 data latih dan 429 untuk data uji. Data komentar yang telah di lakukan pembagian data digunakan untuk tahap ekstraksi fitur.

**Tabel 3.** Pembagian Data

Jenis Data	Persentase	Jumlah Data
Data Latih ( <i>Training Data</i> )	80%	1.716
Data Uji ( <i>Test Data</i> )	20%	429
Total	100%	2.145

### 3.5 Tahap Ekstraksi Fitur TF-IDF

Tahap selanjutnya yaitu ekstraksi fitur menggunakan TF-IDF untuk memproses seluruh data latih dan data uji. Langkah awal yang dilakukan mencari nilai TF berdasarkan jumlah frekuensi setiap kata yang muncul pada 1 dokumen dibagi dengan total kata pada dokumen tersebut. Setelah nilai TF dari masing kata sudah didapatkan langkah selanjutnya menghitung IDF dengan rumus jumlah kemunculan kata pada seluruh dokumen dibagi dengan total dokumen. Selanjutnya menghitung nilai TF-IDF dengan mengkalikan nilai TF yang didapat sebelumnya dengan nilai IDF.



DataID	Kata	Jumlah Kemunculan	Nilai TF	Nilai IDF	Nilai TF-IDF
1	terima	2	0.07692307692307693	1.3753789862181558	0.1057983835524276
	kasih	2	0.07692307692307693	1.3765820187932576	0.10589092452255829
	berani	1	0.038461538461538464	2.1239275732134377	0.08168952204667068
	zainal	1	0.038461538461538464	2.478642427840195	0.09533240107077674
	arifin	1	0.038461538461538464	2.757396028793024	0.1060536934151163
	mochtar	1	0.038461538461538464	2.8031535193536996	0.10781359689821922
	bivitri	1	0.038461538461538464	2.743155589678414	0.10550598421840054
	susanti	1	0.038461538461538464	2.757396028793024	0.1060536934151163
	feri	1	0.038461538461538464	2.4421255940144326	0.09392790746209356

**Gambar 3.** Hasil TF-IDF

### 3.6 Pengujian dan Evaluasi

Tahap pengujian dilakukan dari data latih yang telah dilakukan pembobotan kata menggunakan TF-IDF. Pengujian dilakukan masing-masing pada data label manual dan label otomatis *lexicon* untuk diuji menggunakan algoritma *naïve bayes*, KNN, dan *decision tree*. Sehingga setiap algoritma untuk label manual dan label otomatis menghasilkan *confusion matrix* untuk mendapatkan nilai akurasi, presisi, *F1-score* dan *recall*.

a. *Naïve Bayes*

## 1) Label Manual

**Tabel 4.** Hasil *Naïve Bayes* dengan Label Manual

	Presisi	Recall	F1-Score	Support
Positif	0.81	1.00	0.90	348
Negatif	1.00	0.04	0.07	26
Netral	0.00	0.00	0.00	55
Accuracy			0.81	429
Macro Avg	0.60	0.35	0.32	429
Weight Avg	0.72	0.81	0.73	429

Dari Tabel 4 hasil pengujian untuk algoritma *naïve bayes* dengan menggunakan pelabelan secara manual didapatkan akurasi sebesar 81%.

 2) Label Otomatis *Lexicon*
**Tabel 5.** Hasil *Naïve Bayes* dengan Label *Lexicon*

	Presisi	Recall	F1-Score	Support
Positif	0.66	0.78	0.75	215
Negatif	0.78	0.60	0.67	186
Netral	0.00	0.00	0.00	28
Accuracy			0.70	429
Macro Avg	0.48	0.49	0.48	429
Weight Avg	0.67	0.70	0.67	429

Dari Tabel 5 hasil pengujian untuk algoritma *Naïve Bayes* dengan menggunakan pelabelan secara otomatis *lexicon* didapatkan akurasi sebesar 70%.

 b. *K-Nearest Neighbor*

## 1) Label Manual

**Tabel 6.** Hasil K-NN dengan Label Manual

	Presisi	Recall	F1-Score	Support
Positif	0.89	0.90	0.90	348
Negatif	0.30	0.50	0.37	26
Netral	0.56	0.35	0.43	55
Accuracy			0.80	429
Macro Avg	0.58	0.58	0.56	429
Weight Avg	0.81	0.80	0.80	429

Dari Tabel 6 hasil pengujian untuk algoritma K-NN dengan menggunakan pelabelan secara manual didapatkan akurasi sebesar 80%.

 2) Label Otomatis *Lexicon*
**Tabel 7.** Hasil K-NN dengan Label *Lexicon*

	Presisi	Recall	F1-Score	Support
Positif	0.74	0.51	0.60	215
Negatif	0.58	0.78	0.67	186
Netral	0.23	0.25	0.24	28
Accuracy			0.61	429
Macro Avg	0.52	0.52	0.51	429
Weight Avg	0.64	0.61	0.61	429

Dari Tabel 7 hasil pengujian untuk algoritma K-NN dengan menggunakan pelabelan secara otomatis *lexicon* didapatkan akurasi sebesar 61%.

 c. *Decision Tree*

## 1) Label Manual

**Tabel 8.** Hasil *Decision Tree* dengan Label Manual

	Presisi	Recall	F1-Score	Support
Positif	0.98	0.81	0.89	348
Negatif	0.19	0.81	0.31	26
Netral	0.50	0.27	0.35	55

<i>Accuracy</i>			0.74	429
<i>Macro Avg</i>	0.56	0.63	0.52	429
<i>Weight Avg</i>	0.87	0.74	0.79	429

Dari Tabel 8 hasil pengujian untuk algoritma *decision tree* dengan menggunakan pelabelan secara manual didapatkan akurasi sebesar 74%.

## 2) Label Otomatis *Lexicon*

**Tabel 9.** Hasil *Decision Tree* dengan Label *Lexicon*

	Presisi	Recall	F1-Score	Support
Positif	0.76	0.64	0.70	215
Negatif	0.63	0.80	0.70	186
Netral	0.64	0.25	0.36	28
<i>Accuracy</i>			0.68	429
<i>Macro Avg</i>	0.67	0.56	0.59	429
<i>Weight Avg</i>	0.69	0.68	0.68	429

Dari Tabel 9 hasil pengujian untuk algoritma *decision tree* dengan menggunakan pelabelan secara otomatis *lexicon* didapatkan akurasi sebesar 68%.

## 4. KESIMPULAN

Berdasarkan hasil penelitian untuk analisis sentimen masyarakat terhadap isu kecurangan pemilu yang terdapat film *Dirty Vote* didapatkan hasil akurasi yang berbeda-beda pada setiap algoritma. Hasil pengujian algoritma *naïve bayes* menggunakan pelabelan manual menghasilkan tingkat akurasi sebesar 81% dan pelabelan otomatis *lexicon* menghasilkan akurasi sebesar 70%. Pengujian algoritma K-NN menggunakan pelabelan manual menghasilkan akurasi sebesar 80% dan pelabelan otomatis menghasilkan akurasi sebesar 61%. Pengujian algoritma *decision tree* menggunakan pelabelan manual menghasilkan akurasi sebesar 74% dan pelabelan otomatis menghasilkan akurasi sebesar 68%. Dari hasil pengujian algoritma *naïve bayes*, K-NN, dan *decision tree*, dapat disimpulkan bahwa algoritma *naïve bayes* dengan pelabelan manual menjadi algoritma terbaik dengan menghasilkan tingkat akurasi sebesar 81% dengan presisi sebesar 81%, *recall* sebesar 100% dan F1-Score sebesar 90%. Hasil akurasi pada *naïve bayes* pelabelan manual menghasilkan akurasi yang lebih tinggi dibandingkan dengan pelabelan otomatis menggunakan kamus *lexicon* dari GitHub. Hal ini menunjukkan bahwa data yang diberi label secara manual cenderung lebih akurat karena proses ini melibatkan pemeriksaan inti sentimen secara lebih mendalam, sedangkan pelabelan otomatis dengan *lexicon* cenderung kurang akurat karena bergantung pada bobot kata yang mungkin tidak sepenuhnya menangkap konteks sentimen dari data tersebut. Dalam penelitian ini yang melakukan perbandingan menggunakan tiga algoritma yaitu *naïve bayes*, K-NN, dan *decision tree*, dimasa yang akan datang diharapkan menggunakan algoritma *text mining* lainnya dan melakukan pembagian data (*split data*) dengan rasio yang lebih beragam.

## DAFTAR PUSTAKA

- [1] I. Kurniawan, A. Lia Hananto, S. Shofia Hilabi, A. Hananto, B. Priyatna, and A. Yuniar Rahman, "Perbandingan Algoritma Naive Bayes Dan SVM Dalam Sentimen Analisis Marketplace Pada Twitter," *J. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 1, pp. 731–740, 2023, [Online]. Available: <http://jurnal.mdp.ac.id>
- [2] S. Styawati, A. R. Isnain, N. Hendrastuty, and L. Andraini, "Comparison of Support Vector Machine and Naïve Bayes on Twitter Data Sentiment Analysis," *J. Inform. J. Pengemb. IT*, vol. 6, no. 1, pp. 56–60, 2021, doi: 10.30591/jpit.v6i1.3245.
- [3] A. Prayogo, F. Fauziah, and W. Winarsih, "Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbor Pada Klasifikasi Judul Artikel Pada Jurnal Ilmiah," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 8, no. 4, pp. 1327–1338, 2023, doi: 10.29100/jupi.v8i4.4141.
- [4] M. AUFAR, R. Andreswari, and D. Pramesti, "Sentiment Analysis on Youtube Social Media Using Decision Tree and Random Forest Algorithm: A Case Study," *2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020*, 2020, doi: 10.1109/ICoDSA50139.2020.9213078.
- [5] H. Ashari, D. Arifianto, H. Azizah, and A. Faruq, "Perbandingan Kinerja Algoritma Multinomial Naive Bayes (MNB, Multivariate Bernoulli dan Rocchio Algoritim Dalam Klasifikasi Konten Berita Hoax Berbahasa Indonesia Pada Media Sosial," *Http://Repository.Unmuhjember.Ac.Id*, pp. 1–12, 2020.
- [6] R. Savira, A. Solichin, I. Imelda, and M/ Syafrullah, "Analisis Sentimen Pada Twitter Terhadap Kenaikan BBM 2022 Dengan Lexicon Dan Support Vector Machine," *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi*

- (SENAFTI) KE-2, vol. 2, no. 1, April 2023, pp. 211–218. [Online]. Available: <https://senafti.budiluhur.ac.id/index.php/senafti/article/view/564>
- [7] F. Z. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” *M.Sc. Thesis, Append. D*, vol. pp, pp. 39–46, 2003.
- [8] R. D. Wahyuni and A. N. Utomo, “Penggunaan Metode Lexicon Untuk Analisis Sentimen pada Ulasan Aplikasi KAI Access di Google Play Store,” *J. Rekayasa Inf.*, vol. 11, no. 2, pp. 134–145, 2022.
- [9] R. Darmawan and S. Amini, “Perbandingan Hasil Sentimen Analysis Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbor pada Twitter Comparison of Sentiment Analysis Results Using Naïve Bayes and K-Nearest Neighbor Algorithm on Twitter,” *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, vol. 1, no. 1, September 2022, pp. 495–501. [Online]. Available: <https://senafti.budiluhur.ac.id/index.php/senafti/article/view/210>
- [10] C. A. Misrun, E. Haerani, M. Fikry, and E. Budianita, “Analisis Sentimen Komentar Youtube Terhadap Anies Baswedan Sebagai Bakal Calon Presiden 2024 Menggunakan Metode Naive Bayes Classifier,” *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 4, no. 1, pp. 207–215, 2023, doi: 10.37859/coscitech.v4i1.4790.
- [11] Y. Hariyanti, S. Kacung, and B. Santoso, “Analisis Sentimen Terhadap Putusan Mahkamah Konstitusi Tentang Batasan Umur Capres Dan Cawapres Menggunakan Metode Naïve Bayes,” *Multidiscip. Indones. Cent. J.*, vol. 1, no.1, pp. 517–525, 2024.
- [12] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiyari, “Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE,” *Aiti*, vol. 18, no. 2, pp. 173–184, 2021, doi: 10.24246/aiti.v18i2.173-184.
- [13] R. Metivianis, D. E. Ratnawati, and B. Rahayudi, “Analisis Sentimen pengguna Twitter terhadap Vaksinasi Sinovac dan AstraZeneca menggunakan Algoritma CART,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, pp. 1913–1920, 2022.