Perbandingan Hasil Sentimen Analysis Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbor pada Twitter

Rizky Darmawan[1*], Safrina Amini[2]

[1*,2]Sistem Informasi, Fakultas Teknologi Informasi, Universitas Budi Luhur

Email: rizdar16@gmail.com[1*], safrina.amini@budiluhur.ac.id[2]
(* : corresponding author)

Abstrak-Meningkatnya pengguna media sosial membuat data semakin banyak dan menumpuk, khususnya pada sosial media twitter, data tersebar luas dan sangat mudah didapatkan pada platform ini, oleh karena itu perlu dilakukan sebuah pengolahan data untuk memproses data-data tersebut menjadi sebuah sumber informasi yang berguna. Salah satu cara dalam mengolah data pada twitter disebut sentimen analisis, yaitu mengambil sebuah opini masyarakat yang menggunakan twitter dan mengelompokkannya kedalam kelas emosinya. Namun banyak sekali metode yang dapat digunakan untuk melakukan Sentimen Analisis, oleh karena itu munculah sebuah permasalahan dimana metode apa yang paling cocok digunakan dalam melakukan sebuah sentimen analysis. Pada pengujian ini penulis membandingkan dua algoritma yaitu Naïve Bayes dan K-Nearest Neighbor dalam melakukan sentimen analysis, dengan melakukan 3 kali percobaan menggunakan data yang didapatkan dari twitter. Dihasilkan tingkat akurasi sebesar 65%, 40%,80% untuk algoritma *Naïve* Bayes dan 55%,45%,75% untuk algoritma *K-Nearest Neighbor*. Penelitian ini berkontribusi dalam pemilihan algoritma yang digunakan dalam pembuatan sentimen analisis kedepannya.

Kata Kunci: twitter, sentimen analisis, naïve bayes, k-nearest neighbor, text mining

Comparison of Sentiment Analysis Results Using Naïve Bayes and K-Nearest Neighbor Algorithm on Twitter

Abstract*-The increase in social media users made data more and more piled up, especially on social media twitter, data is widespread and very easy to get on this platform, therefore it is necessary to do a data mining to process the data into a useful source of information. One way of data mining on Twitter is called sentiment analysis, which is to take a public opinion that uses Twitter and group it into their emotional class. However, many methods that can be used to perform Sentiment Analysis, therefore a problem arises where what method is most suitable to be used in conducting a sentiment analysis. In this study, the authors compare two algorithms, namely Naïve Bayes and K-Nearest Neighbor in conducting sentiment analysis, by conducting 3 experiments using data obtained from twitter. The Accuracy rate is 65%, 40%, 80% for the Naïve Bayes Algorithm and 55%, 45%, 75% for the K-Nearest Neighbor algorithm. This research contributes to the selection of which algorithm is used in sentiment analysis in the future.*

Keywords*: twitter, sentiment analysis, naïve bayes, k-nearest neighbor, text mining*

1. PENDAHULUAN

Twitter merupakan salah satu media sosial yang banyak diminati dan digunakan oleh masyarakat Indonesia. Berbagai macam informasi tersebar luas dalam media sosial yang satu ini. Adapula pihak yang menggunakan Twitter sebagai ajang promosi, pemberitaan, dan bisnis. Dari banyaknya data informasi pada media sosial Twitter, banyak data mengadung sebuah emosi, baik itu mencela atau memuji. Emosi ini dibagi menjadi dua, yaitu emosi Positif dan Negatif. Manusia memiliki emosi lima dasar emosi, yaitu emosi cinta, emosi senang, emosi sedih, emosi marah, dan emosi takut. Yang dimana senang dan cinta masuk kedalam kelompok positif sedangkan sedih, marah, dan takut masuk kedalam kelompok negatif

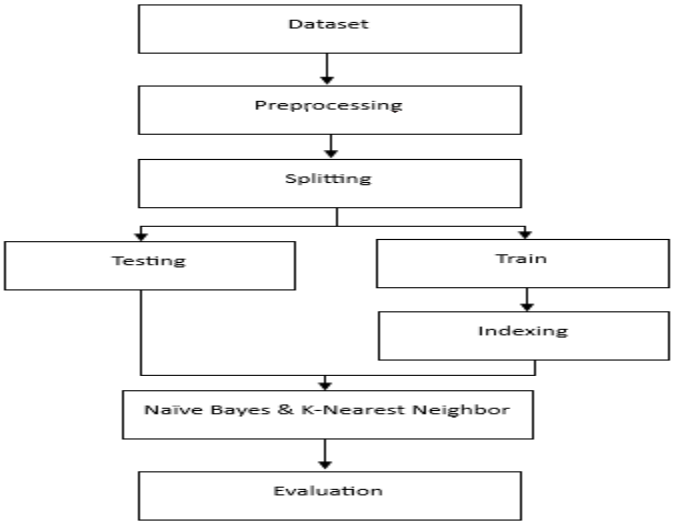
Dalam pengolahan data yang bersumber dari twitter dapat dilakukan Sentimen analisis. Sentimen analisis adalah sebuah proses untuk mengekstrak dan mengolah data dalam bentuk teks untuk mengambil informasi sentimennya secara otomatis yang terdapat dalam sebuah pendapat dan opini. Analisis sentiment dapat digunakan di banyak hal seperti ekonomi, politik, hukum dan sosial. Dari media sosial Twitter inilah data-data bisa sangat mudah didapatkan.

Dalam melakukan sebuah sentimen analisis banyak metode-metode dan algoritma yang bisa digunakan. Pada umumnya Naïve Bayes dan K-Nearest Neighbor digunakan dalam metode klasifikasi karena tingkat akurasi dari kedua metode relatif tinggi. Studi mengenai perbandingan algoritma naïve bayes dan K-nearest neighbor ini sudah pernah dilakukan dengan judul "Analisa Perbandingan Metode Naïve Bayes Classifier Dan K-Nearest Neighbor Terhadap Klasifikasi Data" [1] dan "Perbandingan Algoritma K-Nearest Neighbor, Decision Tree, Dan Naive Bayes Untuk Menentukan Kelayakan Pemberian Kredit" [2] . Penelitian yang bersumber pada *Twitter* dengan judul "Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbors dan Random Forest untuk Klasifikasi

Sentimen Terhadap BPJS Kesehatan pada Media Twitter" [3], dari penelitian terdahulu masih sedikit sekali yang menggunakan *Bag Of Word* dalam metode penyimpanan data dalam melakukan sebuah sentimen analisis

Penelitian ini bertujuan untuk mengetahui algoritma apa yang lebih cocok untuk digunakan dalam melakukan sebuah sentimen analisis yang bersumber dari twitter, dan menjadi acuan dalam pemilihan metode yang akan digunakan dalam melakukan sebuah sentimen analisis kedepannya.

2. METODE PENELITIAN



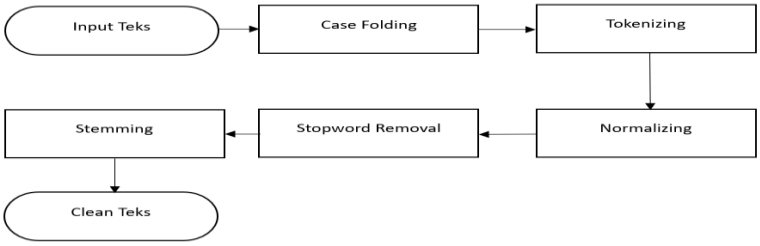
Gambar 1. Tahapan Penelitian

2.1 Dataset

Dataset yang digunakan dalam penelitian ini sebanyak 300 tweets yang dibagi kedalam 3buah dataset dengan topik yang berbeda beda, dan dikumpulkan sejak bulan Juni-Juli 2022. Pengumpulan dataset menggunakan program *python* untuk menarik data secara otomatis dengan menggunakan *keyword* yang ditentukan. Untuk pembuatan program penarikan ini harus memiliki kode API twitter yang sudah terigistrasi pada web developer twitter. Kemudian data yang sudah sudah diambil akan disimpan dalam format CSV. Dataset tersebut akan dilakukan pelabelan secara manual berdasarkan kelas sentimennya yaitu positif dan negatif. Pelabelan secara manual ini untuk pengetesan ketepatan prediksi yang akan dilakukan oleh algortima Naïve Bayes dan K-Nearest Neighbor.

2.2 Preprocessing

Preproceesing data sangat penting dilakukan sebelum diklasifikasikan. *Text Preprocessing* memiliki peranan dalam mengubah data teks yang tidak terstruktur dan mimiliki ambiguitas kata menjadi lebih terstruktur dan mudah diproses karena hasil yang lebih konsistem



Gambar 2. Tahapan Preprocessing

Dari gambar diatas dilihatkan cara pengerjaan pada proses preprocessing dimana tiap inputan mentah akan dioleh dengan menghilangkan dan mengganti tiap katanya menjadi sebuah kalimat yang bersih dan konsisten. Tahapan-tahapannya yaitu:

a. Case Folding

Case Folding merupakan proses pengubahan karakter teks menjadi bentuk standar, setiap karakter dalam text diubah menjadi huruf-huruf kecil. Contohnya "Naïve Bayes" diposisi awal, tengah atapun akhir maka diubah menjadi "naïve bayes". Sistem akan menganggap sebagai *string* yang berbeda dan bisa berakibat kalimatnya terhitung berbeda(menjadi ganda)

b. Tokenizing

Tokenizing merupakan sebuah proses pemecahan setiap kalimat mejadi kata independen. Kalimat diubah kedalam sekumpulan kata tunggal dan dihilangkannya tanda baca. Dengan bantuan *Regular Expression* symbol-simbol dan angka juga dapat dihilangkan

c. Normalization

Normalization berguna untuk mengubah kata-kata singkatan dan Bahasa gaul yang berada dalam dokumen menjadi kata yang sebenarnya. Bertujuan untuk membuat data lebih bersih dan bisa dipahami oleh system

d. Stopword Removal

*Stopwords Remova*l berfungsi untuk menghapus setiap kata yang tidak ada maknanya, seperti kata sambung dan kata umum yang tidak memiliki nilai. *Stopword Removal* bisa dibuat secara manual dan diimport kedalam system, Adapun cara lain dengan memanggil *liblary* yang sudah ada dari python.

e. Stemming

Stemming adalah suatu proses pengolahan kata yang memiliki imbuhan menjadi kata dasar untuk mengurangi ambiguitas dan kesalahan pada perhitungan frekuensi kemunculan kata. Disini penulis menggunakan liblary stemmer factory dari algoritma Nazied & Adriani.

2.3 Splitting

Splitting merupakan proses dimana dibagikannya *dataset* menjadi dua kelompok, yaitu *Testing* dan *Training.*Data *Training* merupakan sebuah data yang berguna untuk membangun dan melatih model pengklasifikasian. Sedangkan *Testing* merupakan data yang akan dilakukan pengujian menggunakan pengklasifikasian yang sebelumnya sudah dibuat menggunakan data *Training.*

Model pengklasifikasian data terbuat dari kumpulan data *Training*, kemudian performa pengklasifikasiannya diukur berdasarkan data *Testing*. Perbandingan antara data *Training* dan data *testing* pada umumnya adalah 80:20 (80% adalah *Training* 20% adalah *testing*) , adapula 50:50 (dimana 50% adalah *training* dan 50% *testing*). Hasil yang optimal pada pengklasifikasian bergantung pada data *training*, jika data *training* dapat mencakup sebagian besar data yang dibutuhkan dalam pengujian data *testing* maka hasilnya yang didapatkan akan maksimal.

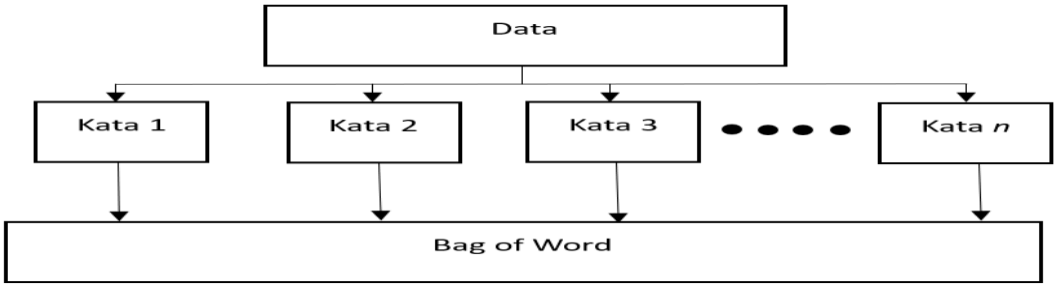
$$\text{Jumlah } Train = \text{proporsi data } Train \times N$$
$$\text{Jumlah } Test = N - \text{total data } Train \quad (1)$$

Dengan :

N = jumlah seluruh data (data *training* + data *testing*)

2.4 Indexing (Bag of Word)

Bag of Word (BoW) merupakan representasi sederhana yang digunakan pada *Natural Language Processing* (NLP) dan *Information Retrieval* (IR). Didalamnya, kalimat dalam teks digambarkan menjadi kantung(bag) dari kata-kata yang berada, dan tidak melihat dari urutan serta posisinya, tetapi perbedaannya tetap ada. *Bag of Words* menyederhanakan representasi kalimat sebagai sekumpulan kata serta mengabaikan tata bahasa dan posisi tiap kalimat. *Bag of Word* menyimpan semua kalimat kedalam sebuah *array* dan memodelkan tiap dokumen dengan cara menghitung kemunculan setiap katanya.



Gambar 3. Proses Bag of Word

Pada Gambar bisa diketahui bahwa tiap kata dalam sebuah kalimat yang berada pada data akan disimpan kedalam *Bag of Word* dan dilakukan perhitungan berapa kali kemunculan katanya pada tiap kalimat yang ada didalam data

2.5 Bernoulli Naïve Bayes

Bernoulli Naïve Bayes adalah suatu metode dalam bidang pengklasifikasian yang diperoleh berdasar pada konsep Bayes, berguna dalam menentukan suatu probabilitas dalam sebuah kejadian dengan cara mempertimbangkan tiap kemungkina peristiwa yang telah terjadi. *Bernoulli Naïve Bayes* hanya menggunakan

binary (True/False, Yes/No, Success / Failure, 0/1)sebagai valuenya.[6]menyatakan bahwa dalam sebuah teks klasifikasi, probabilitas sebuah dokumen 'd' yang tedapat dalam kategori 'c' memiliki persamaan yaitu:

$$P(c|d) \propto P(c) \prod_{1 \le i \le M} P(d|c) \tag{2}$$

Dengan *p(d|c)* merupakan *conditional probability* sedangkan *p(c)* adalah *prior probability* dokumen yang terletak di kategori 'c'[6]. Dalam sebuah klasifikasi *Bernoulli Naïve Bayes*, sebuah kategori yang mempunyai nilai maksimal (*maximum a posterior*) *class Cmap* merupakan kategori yang terbaik

$$Cmap = \arg max\, p(c|d) \tag{3}$$

Rumus *prior probability* nya adalah:

$$P(c) = \frac{N_c}{N} \tag{4}$$

'N_c' adalah total keseluruhan dokumen yang terdapat pada katergori c, dan '*N*' adalah total dokumen pada semua kategori. *Conditional probability* dihitung menggunakan dokumen *training* yang terdapat pada kategori 'c' dangan *term* (berada pada sembarang posisi dan beberapa kali kemunculan) menggunakan rumus :

$$P(d|c) = \prod_{1 \le i \le M} (B_{it} P(e_i|c) + ((1 - B_{it})(1 - P(e_i|C))) \tag{5}$$

dengan $p(e_i|c)$ adalah probabilitas sebuah dokumen yang terletak pada kategori c, *term* akan muncul (berada pada sembarang posisi dan beberapa kali kemunculan). B_{it}memiliki nilai 0 jika *term* tidak ada dan 1 jika *term* muncul pada dokumen tersebut. Ketidak munculan *term* dalam model ini menjadi factor yang diperhitungkan karna jika ada value 0 dalam sebuah persamaan maka hasilnya akan 0 dan akan terjadi *zero probability*

$$P(e_i|c) = \frac{df_i + 1}{df_i + 2} \tag{6}$$

df' merupakan total keseluruhan dokumen yang terletak pada kategori 'c' dan df_i merupakan total dokumen *Training* dalam kategori 'c' yang terdapat *term*. '2' adalah sebuah nilai tetap yang dalam yang digunakan bila *term* muncul ataupun tidak. '1' berguna untuk *Laplace smoothing* supaya tidak terjadi *Zero probability* [6].

2.6 K-Nearest Neighbor

*K-Nearest Neighbor (*K-NN*)* merupakan sebuah metode untuk memperoleh titik hasil yang baru diklasifikasikan berdasar kelas terbanyak. Tidak ada pemodelan apapaun dalam klasifikasi ini dan berdasar pada memori yang menggunakan jumlah terbanyak diantara klasifikasi dari nilai K sebagai prediksi dari titik yang terbaru dekatnya jarak titik dengan tetangganya bisa dihitung dengan menggunakan *euclidien distance* yang direpresentasikan sebagai berikut :

$$d(x_{ik}, x^*_{jk}) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x^*_{jk})^2} \tag{7}$$

Dimana :

$d(X_{ik}, X^*_{jk})$ = Jarak Euclidean *training* ke-*I* dengan data *testing* ke-*j*
X_{ik} = Nilai variebel bebas ke-*k* dari data *training* ke-*I,i=1,2,…,n*
X^*_{jk} = Nilai variable bebas ke-*k* dari data *testing* *i=1,2,…,n**
P = banyaknya variable bebas

Pada metode ini K dinyatakan sebagai hasil terdekat yang terlibat dalam proses menentuka prediksi kelas yang diuji. Tiap K yang tergolong kedalam kelompok terdekat kemudian dilakukan pemilihan kelas dari nilai K tersebut. Suara yang memiliki jumlah tetangga terbanyak akan dijadikan sebagai hasil perdiksi kelas pada data *Train*nya

2.7 Evaluation

Hasil kedua algoritma akan dievaluasi dengan *Confussion matrix*. *Confusion matrix* menampilkan sejumlah informasi hasil pengklasifikasian yang sudah dilakukan dengan hasil yang sebenarnya. *Confusion matrix* digambarkan dengan bentuk tabel yang membandingkan hasil performa model pada klasifikasi dengan nilai data yang sebenarnya.

		ACTUAL	
		POSITIVE	NEGATIVE
PREDICT	POSITIVE	TP (TRUE POSITIVE)	FP (FALSE POSITIVE)
	NEGATIVE	FN (FALSE NEGATIVE)	TN (TRUE NEGATIVE)

Gambar 4. Confussion Matrix

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP): Data yang sebenarnya Positif dan diprediksi Positif; *True Negative* (TN): Data yang sebenarnya Negatif dan diprediksi Negatif; *False Positive* (FP): Data yang sebenarnya Negatif dan diprediksi Positif*; False Negative* (FN): Data yang sebenarnya Positif dan diprediksi Negatif

Berdasar pada *confusion matrix diatas*, dapat dihasilkan beberapa rumus pengukuran performa algoritma, dengan rumus :

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$precision = \frac{TP}{TP + FP} \quad (9)$$

$$recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1\ Score = \frac{2 * (Precission * Recall)}{Precission + recall} \quad (11)$$

3. HASIL DAN PEMBAHASAN

Hasil percobaan ini berupa nilai akurasi pada setiap algoritme dalam memprediksi kelas pada data *Testing.* Hasil akurasi ini didapatkan setelah melalui tiap langkah yang sudah ditampilkan pada gambar(1). Dimulai dengan pengambilan data dari twitter dan dilabelkan sentimennya secara manual.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	id	sentimen	tweets															
2	1	0	b'RT @ezash: Sebenernya win-win dari WFH x WFO ini adalah hybrid-policy, imho\n\nLagi males commute ke kantor? Ya kerja di rumah silahkan\n\nLagi\xe2\x80\xa6'															
3	2	0	b'@mineandpurple Bener wkwkwk.\nPas WFH setahun, malah mengexplore diri dgn keanehan yg belum pernah dicoba \xf0\x9f\x98\x82'															
4	3	0	b'RT @siKirun: @ezash setuju sih sama ini, bukan berarti WFH gak produktif, tidak juga WFO itu mandatory wajib. WFO lembur mulu, WFH, tengah\xe2\x80\xa6'															
5	4	1	b'Pengennya bisa WFA atau hybrid. \nKdg merasa wasting time and energy ke kantor. kdg jenuh WFH bisa ke kantor (WFO),\xe2\x80\xa6 https://t.co/krRRJtkOIn'															
6	5	0	b'Bersyukurnya kantor ku udh nerapin ini, jadi yg mau WFH monggo, mau ke kantor juga silakan, aman dan nyamannya karyawan \xe2\x80\xa6 https://t.co/4EESkEt7fx'															
7	6	1	b'tim gasuka wfh karena jaringan internet di rumah tidak memadai\xf0\x9f\xa5\xb2 https://t.co/kOmbYLckUg'															
8	7	1	b'Seminggu ini, karna pelatihan jadi wfh. Jujur gak siap buat ke kantor besok.\nTakut banget'															
9	8	0	b'SEPAKAT. Gue ke kantor cuma 1x seminggu di hari selasa. Jujur jd happy nunggu-nunggu saat ke kantor, di sisi lain W\xe2\x80\xa6 https://t.co/izyJwkfOZz'															
10	9	1	b'@ezash Dulu zoom cuma dipake buat wfh. Sekarang udah ngabisin waktu di jalan, nyampe kantor malah zoom pula \xf0\x9f\xa5\xb2'															
11	10	1	b'@ezash WFH tapi meeting terus diluar jam kantor juga buat apa. WFO tapi lembur tiap hari juga buat apa \xf0\x9f\x99\x88'															
12	11	0	b'Sjak implementasi wfh jd ktauan tu kerjaan mana aja yg bisa tetep optimal klo di rumah mana yg ngga.\n\nKampeni jd bi\xe2\x80\xa6 https://t.co/MisAKuaqJB'															
13	12	0	b'Bener udh nyaman double gini\nAlhamdulillah dpt kantor yg msh rutin schedule wfo+wfh jd gak bosen kecapeam ke kantor\xe2\x80\xa6 https://t.co/y2vc66NTMG'															
14	13	1	b'@chrysanthelaa fix bentar lagi dia gangguin gw wfh\xf0\x9f\x98\x94'															
15	14	1	b'Kalo saya, semenjak bisa WFH, balik ke sistem WFO malah bikin kerjaan kurang efisien\n\nContoh kecilnya :\nWFH : Bangu\xe2\x80\xa6 https://t.co/q7BKB6acVc'															
16	15	0	b'@Baratheeon gak usah ngadi2, gue lagi menikmati kenikmatan wfh'															
17	16	0	b'Bsk senin dan wfh, haaaaa leganya ga ketemu orang2 kantor di tempat magang.'															
18	17	0	b'@ezash Setuju. Kadang bosen kalo kelamaan WFH ataupun WFO, yg paling pas ya sesuai kebutuhan'															
19	18	1	b'Yallah kalo mau cari loker tp wfh apa masih ada yaa :"(('															
20	19	0	b'@worksfess Aku yg selalu wfh, nonton drakor/film/yt, baca novel/au gt aja nder. Kadang random juga telfonan sama temen'															
21	20	1	b'Semalam punya penat gara gara besok wfo &amp															
22	21	0	b'besok wfh ceunahh, agak senang dikitt'															

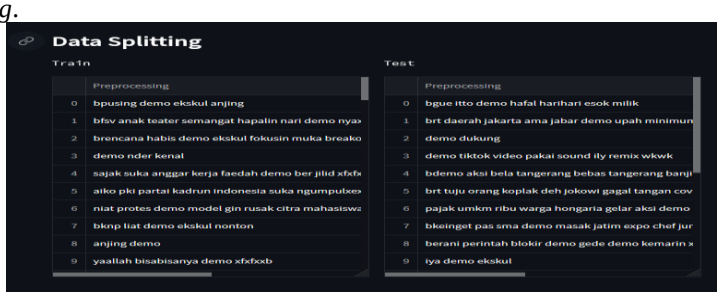
Gambar 5. Dataset

Data yang baru ditarik dari twitter masih banyak simbol simbol dan link. Oleh karena itu diharuskan untuk masuk ke tahap *Preprocessing* untuk menghilangkan simbol,link dan membuat tiap tweet lebih konsisten.

	id	sentiment	tweets	Preprocessing
0	1	positif	Dalam Pilkada DKI, AHY juga mampu menunjukkar	pilkada dki ahy kualitas pimpin muda jiwa ksatria ahy dunia
1	2	positif	Terima kasih mas @AgusYudhoyono, karenamu an	terima kasih mas karena anak muda berani suara ubah sama
2	3	positif	KETEMU DI ISTIQLAL Bangga Pernah Jadi Bagian D	ketemu istiqlal bangga murni
3	4	positif	Tersingkir dari kancah pilkada DKI, rupanya ada ca	singkir kancah pilkada dki angkat dunia
4	5	positif	usai pilkada DKI AHY keliling Nusantara merajut Pe	pilkada dki ahy keliling nusantara rajut satu jakarta
5	6	positif	(12) AHY justru menitipkan pentingnya menjaga pl	ahy titip jaga pluralisme bangsa satu saing pilkada satu
6	7	negatif	Buat pendukung . AHY mari rapatkan barisan ke pa	dukung ahy mari rapat baris paslon buang issu miring paslon pimpin dakwa simul
7	8	negatif	Harusnya rakyat menangis melihat . AHY KALAH. B	rakyat menang ahy kalah aspek ahy pribadi penting rakyat tetapahy
8	9	negatif	Maksud hati caper dari ahy/silvy, eh malah kena pr	maksud hati caper ahysilvy eh kena protes face with stuckout tounges rolling on tl
9	10	positif	Ingat, fokus dan menangkan #AHY #AgusYudhoyon	fokus menang
10	11	positif	Langkah "Kuda" Agus Yudhoyono -> https://goo.gl/	langkah kuda agus yudhoyono
11	12	positif	fDatang tampak muka pulang tampak punggung. T	fdatang muka pulang punggung terima kasih kepri
12	13	negatif	@ZaraZettiraZZ @NetizenTofa @ndoro__putri ada	putri ahoker ngebully program face with tears of joy face with tears of joy face with
13	14	negatif	AHY kemungkinan besar tersingkir <Persevering Fa	ahy singkir persevering face persevering face pilkadadki dilanjuntukan putar ahok

Gambar 6. Preprocessing

Setelah semua tweet dilakukan preprocessing maka akan menghasilkan data yang bersih untuk dapat masuk keproses selanjutnya, yaitu *Splitting Data*. Data dibagi menjadi 2 yaitu data *Training* dimana setiap katanya akan disimpan kedalam *Bag of Word* sebagai data latih. Data *Testing* akan menjadi bahan uji tiap algoritma dalam memprediksi kelas sentimennya dan menggunakan data *Training* yang sudaha disimpan menjadi acuan dalam memprediksi kelasnya. Menggunakan mode 80:20 dimana 80% dari total data merupakan data *Training* dan 20% merupakan data *Testing*.

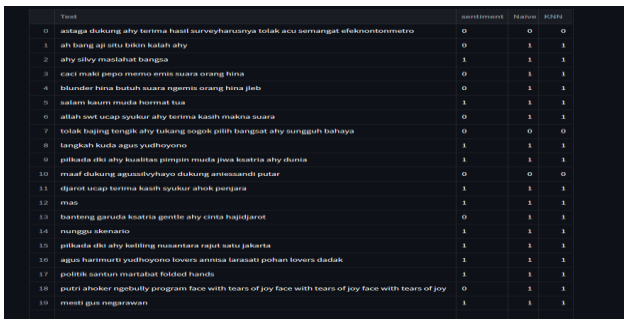


	Preprocessing
0	bpusing demo ekskul anjing
1	bfsv anak teater semangat hapalin nari demo nyaa
2	brencana habis demo ekskul fokusin muka breako
3	demo nder kenal
4	sajak suka anggar kerja faedah demo ber jilid xfxfx
5	aiko pki partai kadrun indonesia suka ngumpulxe
6	niat protes demo model gin rusak citra mahasiswa
7	bknp liat demo ekskul nonton
8	anjing demo
9	yaallah bisabisanya demo xfxfxxb

	Preprocessing
0	bgue itto demo hafal harihari esok milik
1	brt daerah jakarta ama jabar demo upah minimun
2	demo dukung
3	demo tiktok video pakai sound ily remix wkwk
4	bdemo aksi bela tangerang bebas tangerang banji
5	brt tuju orang koplak deh jokowi gagal tangan cov
6	pajak umkm ribu warga hongaria gelar aksi demo
7	bkeinget pas sma demo masak jatim expo chef jur
8	berani perintah blokir demo gede demo kemarin x
9	iya demo ekskul

Gambar 7. Splitting data

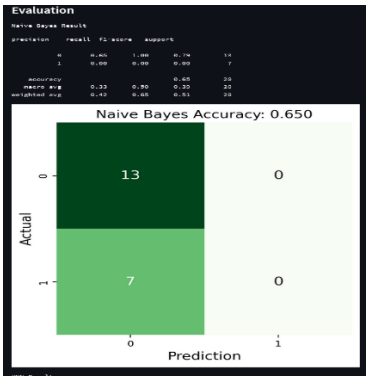
Data *Testing* ini akan diuji menggunakan algoritma Naïve Bayes dan K-Nearest Neighbor untuk memprediksi kelas sentimennya menggunakan data *Train* yang sudah disimpan menggunakan metode *Bag of Word* sebagai value tiap katanya. Hasil prediksi akan dibandingkan dengan pelabelan awal secara manual untuk melihat akurasinya



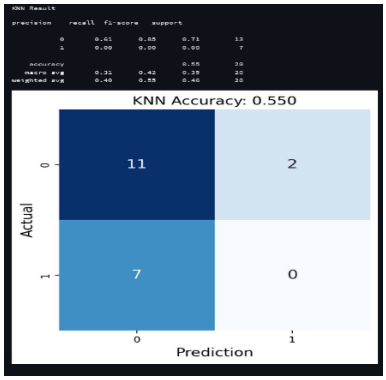
	Text	sentiment	Naive	KNN
0	astaga dukung ahy terima hasil surveyharusnya tolak acu semangat efeknontonmetro	0	0	0
1	ah bang aji situ bikin kalah ahy	0	1	1
2	ahy silvy maslahat bangsa	1	1	1
3	caci maki pepo memo emis suara orang hina	0	1	1
4	blunder hina butuh suara ngemis orang hina jleb	0	1	1
5	salam kaum muda hormat tua	1	1	1
6	allah swt ucap syukur ahy terima kasih makna suara	0	1	1
7	tolak bajing tengik ahy tukang sogok pilih bangsat ahy sungguh bahaya	0	0	0
8	langkah kuda agus yudhoyono	1	1	1
9	pilkada dki ahy kualitas pimpin muda jiwa ksatria ahy dunia	1	1	1
10	maaf dukung agussilvyhayo dukung aniessandi putar	0	0	0
11	djarot ucap terima kasih syukur ahok penjara	1	1	1
12	mas	1	1	1
13	banteng garuda ksatria gentle ahy cinta hajidjarot	0	1	1
14	nunggu skenario	1	1	1
15	pilkada dki ahy keliling nusantara rajut satu jakarta	1	1	1
16	agus harimurti yudhoyono lovers annisa larasati pohan lovers dadak	1	1	1
17	politik santun martabat folded hands	1	1	1
18	putri ahoker ngebully program face with tears of joy face with tears of joy face with tears of joy	0	1	1
19	mesti gus negarawan	1	1	1

Gambar 8. Hasil Prediksi Algoritma

Dari hasil prediksi ini dibuat menjadi sebuah *Confussion Matriks* untuk memperjelas hasil perbandingan dan menemukan akurasi dari algoritma Naïve Bayes dan K-Nearest Neighbor



Gambar 9. Confussion Matriks Naïve Bayes



Gambar.10 Confussion Matrix K-Nearest Neighbor

Dari hasil *Confussion Matrix* bisa dilihat bahwa pada percobaan menggunakan dataset 1 Naïve bayes memiliki akurasi sebesar 65% dan K-Nearest Neighbor memiliki akurasi sebesar 55%. Lanjutkan percobaan menggunakan dataset berikutnya. Hasil menggunakan dataset selanjutnya akan ditampilkan apda tabel dibawah

Tabel 2. Hasil Perbandingan

Data	Jumlah	NB	K-NN
Dataset 1	100	65%	55%
Dataset 2	100	40%	45%
Dataset 3	100	80%	75%

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan terhadap 3buah dataset menggunakan algoritma Naïve Bayes dan K-Nearest Neighbor hanya mempunyai sedikit perbedaan, yang membuat kedua algoritma ini memang layak digunakan untuk melakukan sebuah Sentimen Analysis. Dari percobaan pengujian ini Naïve Bayes lebih sedikit unggul dibandingkan K-Nearest Neighbor pada percobaan ke-1 dan ke-3 dengan akurasi 65% dan 80%.

Dalam pengujian perbandingan antara algoritma Naïve Bayes dan K-Nearest Neighbor dimasa mendatang diharapkan menggunakan metode pembobotan lain seperti TF-IDF untuk memperbesar tingkat keakurasiannya

DAFTAR PUSTAKA

[1] A. Indriani, "Analisa Perbandingan Metode Naïve Bayes Classifier Dan K-Nearest Neighbor Terhadap Klasifikasi Data," *Sebatik*, vol. 24, no. 1, pp. 1–7, 2020, doi: 10.46984/sebatik.v24i1.909.

[2] T. T. Muryono, A. Taufik, and I. Irwansyah, "Perbandingan Algoritma K-Nearest Neighbor, Decision Tree, Dan Naive Bayes Untuk Menentukan Kelayakan Pemberian Kredit," *Infotech J. Technol. Inf.*, vol. 7, no. 1, pp. 35–40, 2021, doi: 10.37365/jti.v7i1.104.

[3] T. A.M and A. Yaqin, "Perbandingan Algoritma Naïve Bayes, K-Nearest Neighbors dan Random Forest untuk Klasifikasi Sentimen Terhadap BPJS Kesehatan pada Media Twitter," *InComTech J. Telekomun. dan Komput.*, vol. 12, no. 1, p. 01, 2022, doi: 10.22441/incomtech.v12i1.13642.

[4] R. N. Devita, H. W. Herwanto, and A. P. Wibawa, "Perbandingan Kinerja Metode Naive Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa indonesia," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, p. 427, 2018, doi: 10.25126/jtiik.201854773.

[5] A. D. Afifaturahman and F. MSN, "Perbandingan Algoritma K-Nearest Neighbour (KNN) dan Naive Bayes pada Intrusion Detection System (IDS)," *Innov. Res. Informatics*, vol. 3, no. 1, pp. 17–25, 2021, doi: 10.37058/innovatics.v3i1.2852.

[6] V. Novalia, R. Goejantoro, and Sifriyani, "Perbandingan Metode Klasifikasi Naive Bayes dan K-Nearest Neighbor (Studi Kasus : Status Kerja Penduduk Di Kabupaten Kutai Kartanegara Tahun 2018)," *J. EKSPONENSIAL*, vol. 11, pp. 159–166, 2020.

[7] F. N. Hasan, N. Hikmah, and D. Y. Utami, "Perbandingan Algoritma C4.5, KNN, dan Naive Bayes untuk Penentuan Model Klasifikasi Penanggung jawab BSI Entrepreneur Center," *J. Pilar Nusa Mandiri*, vol. 14, no. 2, p. 169, 2018, doi: 10.33480/pilar.v14i2.908.

[8] I. Saputra and D. Rosiyadi, "Perbandingan Kinerja Algoritma K-Nearest Neighbor, Naïve Bayes Classifier dan Support Vector Machine dalam Klasifikasi Tingkah Laku Bully pada Aplikasi Whatsapp," *Fakt. Exacta*, vol. 12, no. 2, p. 101, 2019, doi: 10.30998/faktorexacta.v12i2.4181.

[9] A. Sumiah and N. Mirantika, "Perbandingan Metode K-Nearest Neighbor dan Naive Bayes untuk Rekomendasi Penentuan Mahasiswa Penerima Beasiswa pada Universitas Kuningan," *Buffer Inform.*, vol. 6, no. 1, pp. 1–10, 2020.

[10] P. L. Prasanna, S. Manogni, P. Tejaswini, K. T. Kumar, and K. Manasa, "Document classification using KNN with fuzzy bags of word representation," *Int. J. Recent Technol. Eng.*, vol. 7, no. 6, pp. 631–634, 2019.