

PERBANDINGAN ALGORITMA MACHING LEARNING DALAM ANALISIS SENTIMEN MOBIL LISTRIK DI INDONESIA PADA MEDIA SOSIAL TWITTER/X

Ade Yusupa¹, Victor Tarigan²

^{1,2} Teknik Elektro, Teknik, Universitas Sam Ratulangi

¹ ade@unsrat.ac.id , ² victortarigan@unsrat.ac.id

Abstrak

Twitter telah menjadi platform yang penting dalam mencermati opini publik, terutama terkait dengan topik-topik populer seperti perkembangan mobil listrik di Indonesia. Dalam konteks ini, analisis sentimen dapat digunakan untuk memahami sikap dan pandangan pengguna terhadap mobil listrik. Namun, meskipun industri mobil listrik di Indonesia mengalami pertumbuhan pesat, masih ada sejumlah masalah yang perlu diatasi, seperti infrastruktur pengisian daya yang belum memadai, harga yang masih relatif tinggi, dan kesadaran masyarakat yang perlu ditingkatkan terkait manfaat lingkungan. Penelitian sebelumnya menunjukkan bahwa teknik *machine learning*, seperti *Support Vector Machine (SVM)*, *Random Forest*, dan *Naïve Bayes* telah berhasil dalam menganalisis sentimen dari data Twitter terkait topik tertentu. Dalam penelitian ini, dapat diuji dan dibandingkan ketiga algoritma *machine learning* tersebut dalam menganalisis sentimen pengguna Twitter terhadap mobil listrik di Indonesia. Pada penelitian sudah digunakan data dari Twitter/X untuk melakukan analisis sentimen dengan memanfaatkan metrik evaluasi seperti akurasi, presisi, dan recall. Penelitian ini bertujuan untuk mengevaluasi performa relatif dari SVM, *Random Forest*, dan *Naïve Bayes* dalam mengklasifikasikan sentimen pengguna Twitter terhadap mobil listrik. Data yang digunakan bersumber dari media sosial Twitter/X dengan jumlah data yang dijadikan bahan penelitian berjumlah 1518 data dengan rincian 870 data dengan label negatif, 143 untuk label netral, dan 502 untuk label positif. Hasil dari penelitian didapatkan bahwa Nilai akurasi tertinggi didapatkan oleh algoritma *Support Vector Machine* sebesar 75,62%, Nilai presisi tertinggi didapatkan oleh algoritma *Naïve Bayes* sebesar 81,29% dan nilai recall tertinggi adalah algoritma *Support Vector Machine* dengan nilai 53,88%. Dari pengujian ini algoritma terbaik yaitu algoritma *Support Vector Machine*.

Kata kunci : Twitter, Machine Learning, Mobil Listrik, Support Vector Machine, Naive Bayes, Random Forest.

1. Pendahuluan

Twitter adalah platform media sosial yang populer di mana pengguna dapat membuat, membaca, dan berinteraksi dengan pesan singkat yang disebut "tweets". Tweets ini memiliki batasan jumlah karakter yang disebut 280 karakter (Cam et al., 2024). Twitter memiliki jangkauan yang luas di seluruh dunia dan menjadi tempat bagi masyarakat untuk berbagi pemikiran, opini, informasi, dan berita tentang berbagai topik, termasuk berita terkini, politik, olahraga, hiburan, dan teknologi (Sunitha et al., 2022). Seiring dengan banyaknya pengguna, maka semakin banyak data yang ditampung oleh twitter. Twitter menjadi sumber data yang dapat dimanfaatkan dalam berbagai kepentingan (Garcia & Berton, 2021).

Dalam bidang penelitian, data pada aplikasi twitter juga dapat digunakan untuk menganalisis opini pengguna terhadap suatu topik atau kebijakan tertentu dengan mengklasifikasikannya kedalam kelas positif, negative, atau netral (Rosenberg et al., 2023). Salah satu topik yang sering dibicarakan di media sosial twitter/X dewasa ini yaitu membahas

tentang perkembangan mobil listrik di Indonesia. Karena sifatnya yang terbuka dan cepat, Twitter menjadi platform yang ideal untuk melacak opini publik tentang mobil listrik. Opini ini dapat berupa pandangan positif tentang pengalaman berkendara mobil listrik, kekhawatiran tentang infrastruktur pengisian daya, atau perbandingan antara mobil listrik dengan mobil bensin.

Industri mobil listrik di Indonesia sedang mengalami pertumbuhan pesat. Menurut data Gabungan Industri Kendaraan Bermotor Indonesia (Gaikindo), penjualan mobil listrik di Indonesia pada tahun 2022 mencapai 3.078 unit, meningkat 201,7% dibandingkan tahun 2021 (Ruan & Lv, 2023). Pertumbuhan ini didorong oleh berbagai faktor, seperti meningkatnya kesadaran masyarakat akan pentingnya kelestarian lingkungan, serta kebijakan pemerintah yang mendukung pengembangan mobil listrik, seperti Peraturan Presiden Nomor 55 Tahun 2019 tentang Percepatan Program Kendaraan Listrik Berbasis Baterai untuk Transportasi Darat (Zola et al., 2023).

Banyak pengguna media sosial yang membagikan pengalaman mereka tentang mobil

listrik, baik positif maupun negatif. Hal ini dapat mempengaruhi persepsi masyarakat tentang mobil listrik. Sebuah studi oleh Universitas Indonesia menunjukkan bahwa 72% pengguna Twitter di Indonesia memiliki sentimen positif terhadap mobil listrik (Motz et al., 2022). Analisis sentimen merupakan salah satu teknik yang dapat digunakan untuk memahami opini publik tentang suatu topik. Analisis sentimen dapat digunakan untuk mengidentifikasi sentimen positif, negatif, dan netral dari teks. Teknik ini telah banyak digunakan untuk menganalisis opini tentang berbagai topik, seperti politik, ekonomi, dan produk (Fitri et al., 2019).

Untuk mengidentifikasi analisis sentimen media sosial, akan dimanfaatkan teknologi *machine learning* untuk proses identifikasi. *Machine learning* (ML) adalah cabang dari kecerdasan buatan (*artificial intelligence*/AI) yang berfokus pada pengembangan algoritme yang dapat secara otomatis belajar dan meningkatkan kinerjanya dari pengalaman tanpa diprogram secara eksplisit (Lickert et al., 2020).

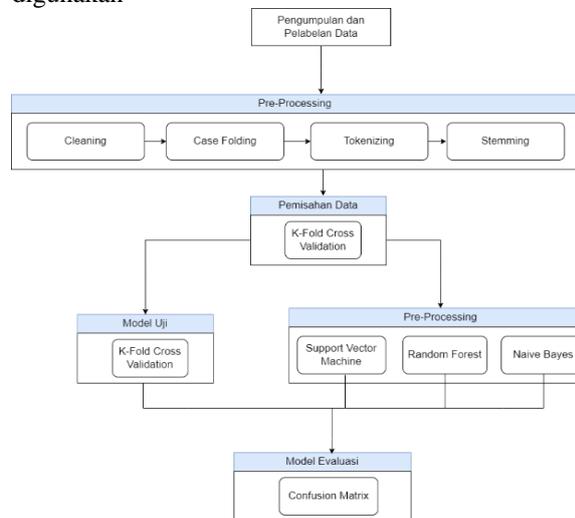
Ada banyak metode *machine learning* yang digunakan dalam analisis sentiment, seperti pada penelitian yang dilakukan oleh Rahadi Ramlan dengan judul dengan judul “Analisis Sentimen Pengguna Twitter Menggunakan *Support Vector Machine* Pada Kasus Kenaikan Harga BBM”. Dari penelitian ini menghasilkan dari hasil testing 52 data didapat hasil 43 komentar negatif dan 9 komentar positif sehingga dapat disimpulkan masyarakat lebih banyak yang tidak setuju dengan adanya kenaikan harga BBM. Penelitian yang dilakukan oleh Oceandra Audrey pada tahun 2022 dengan judul berkaitan Analisis Sentimen Twitter terhadap opini Non Fungible Token di Indonesia menggunakan algoritma Random Forest menghasilkan *confusion matrix* rata-rata sama besar yaitu 95% untuk precision, recall, f-1 score, maupun accuracy. Sedangkan penelitian yang dilakukan oleh Dedi darwis yang berjudul penerapan algoritma Naïve Bayes untuk analisis sentiment pada media sosial twitter menghasilkan uji akurasi pada metode naive bayes untuk klasifikasi yaitu 69.97%.

Berdasarkan penelitian terdahulu yang menghasilkan nilai akurasi yang cukup tinggi, penelitian ini akan menguji serta membandingkan ketiga algoritma machine learning ini terhadap sentiment masyarakat terhadap adanya mobil listrik di Indonesia yang ada di media sosial Twitter/X. Ketiga performa algoritma Vector Support Machine, Random Forest, dan Naïve Bayes akan diuji dan dihitung menggunakan persamaan *Confusion Matrix* untuk mendapatkan mendapatkan hasil berupa nilai akurasi, presisi, recall, dan f-measure sehingga dapat mengetahui performa dari masing-masing algoritma (Audrey et al., 2022).

2. Metodologi Penelitian

Penelitian ini menggunakan beberapa algoritma machine learning untuk dilakukan perbandingan

performa dalam pengklasifikasian analisis sentimen, yaitu Support Vector Machine, Random Forest, dan Naïve Bayes. Gambar 1 menunjukkan tahapan penelitian perbandingan algoritma klasifikasi yang digunakan



Gambar 1. Metodologi Penelitian

Pada Gambar 1 tahapan penelitian dimulai dengan melakukan pengumpulan dataset. Dataset yang digunakan bersumber dari media sosial Twitter/X dengan menggunakan teknik *Crawling* dengan menggunakan bahasa pemrograman Python. Setelah mendapatkan data tersebut dalam bentuk Excel, kemudian dilakukan pelabelan secara manual menjadi 2 klasifikasi, yaitu positif dan negatif. Selanjutnya dilakukan tahapan *Preprocessing*, proses ini artinya adalah mengubah data yang tidak terstruktur menjadi terstruktur sebagai bahan penelitian. Tahapan data preprocessing yaitu, *cleaning*, *case folding*, *tokenizing*, dan *stemming* (Darwis et al., 2021).

Tahap selanjutnya yaitu adalah pemisahan data menggunakan metode K-Fold Cross Validation menjadi 2 bagian, yaitu data *training* dan data *testing*. Pada data *testing* akan dilakukan pembelajaran menggunakan algoritma Support Vector Machine, Naïve Bayes, dan Random Forest dan dari proses pembelajaran akan diketahui performa dari masing-masing algoritma sehingga dapat dilakukan perbandingan kinerja. Untuk mengukur perbandingan kinerja dari algoritma-algoritma tersebut, akan digunakan teknik perhitungan *Confusion Matrix* untuk mendapatkan nilai performa akurasi, presisi, recall, dan F-Measure. Pada penelitian ini akan menggunakan Bahasa Pemrograman Python sebagai alat pengolahan data.

2.1. Pengumpulan dan Pelabelan Dataset

Dataset penelitian yang digunakan bersumber dari media sosial Twitter/X dengan menggunakan teknik *Crawling* menggunakan Bahasa pemrograman Python dengan memanfaatkan *Application Programming Interface* (API). Dataset diambil mulai

dari bulan September 2022 sampai dengan agustus 2023 dan didapatkan data sebanyak 1518 data dengan menggunakan kata kunci “mobil listrik”. Selanjutnya lakukan pelabelan untuk masing-masing data yang sudah didapatkan dengan 870 data dengan label negatif, 143 untuk label netral, dan 502 untuk label positif. Tabel 1 menunjukkan beberapa contoh dataset setelah dilakukan proses pelabelan.

Tabel 1. Contoh Dataset Penelitian

Text	Value
niat beli ev murah malah ikut dinaikin sama perintah terus rakyat jangan kasih sejahtera	negatif
cerdas orang hny dn pasar itu jln x blom da gimana kota ngecas jam br harga nih bbm naikin	negatif
antar anak sekolah antar bantu pasar klw jauh klw jebak macet abis dorong pulang dekarbonisasi eeh malah bangun pltu	negatif
cuma punya sepeda tahun	netral
moga punya produk anak bangsa aamiin allah yarobbal alamin	positif
pegawai negeri karyawan swasta cepat alih kendar ganti baterai pasti mampu baik cicil maupun kontan gantung dapat masing	positif

2.2. Preprocessing Teks

Preprocessing teks adalah serangkaian langkah yang dilakukan untuk membersihkan, mengatur, dan mempersiapkan data teks mentah agar dapat diolah lebih lanjut oleh algoritma pemrosesan bahasa alami (Natural Language Processing/NLP) atau model pembelajaran mesin lainnya. Langkah-langkah dalam preprocessing teks bervariasi tergantung pada kebutuhan dan tujuan spesifik dari analisis atau tugas yang ingin dilakukan (Chai et al., 2023). Namun, beberapa langkah umum dalam preprocessing teks meliputi (Nota et al., 2022):

1. *Cleaning*, yaitu data dilakukan pembersihan dengan tujuan untuk menghapus tanda baca, mention, hashtag, link, dan karakter lainnya pada dokumen.
2. *Case Folding*, proses dalam preprocessing teks di mana semua karakter huruf diubah menjadi huruf kecil atau huruf besar, tergantung pada konvensi yang dipilih
3. *Tokenizing*, Memisahkan teks menjadi unit-unit yang lebih kecil seperti kata-kata, frasa, atau kalimat. Ini adalah langkah dasar dalam pemrosesan teks.
4. *Stemming*, proses untuk mengubah kata yang berimbuhan menjadi kata dasar.

2.3. Cross Validation

Cross-validation adalah teknik yang umum digunakan dalam pembelajaran mesin dan statistik untuk mengevaluasi kinerja model prediksi atau klasifikasi. Tujuan utamanya adalah untuk mengukur seberapa baik model akan memprediksi data baru yang tidak terlihat saat dilatih. Proses cross-

validation melibatkan pemisahan data menjadi subset yang saling bersinggungan, yang kemudian digunakan untuk melatih dan menguji model secara bergantian. Dalam teknik ini, dataset dibagi menjadi sejumlah 10 buah partisi yang terdiri dari data latih dan data uji yang dilakukan secara acak oleh system.

2.4. Model Machine Learning

Model machine learning yang digunakan ada 3, yaitu *Support Vector Machine*, *Random Forest*, dan *Naïve Bayes*.

1. *Support Vector Machine*, Support Vector Machine (SVM) adalah salah satu jenis algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. SVM awalnya dikembangkan untuk tugas klasifikasi, tetapi dapat diperluas untuk menangani masalah regresi dan deteksi anomaly (Romero et al., 2021).
2. *Random Forest*, *Random Forest* adalah salah satu jenis algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi, regresi, dan pemilihan fitur. Ini merupakan bagian dari keluarga algoritma yang dikenal sebagai ensemble learning, di mana model-model lemah dikombinasikan untuk membentuk model yang lebih kuat (Aria et al., 2021).
3. *Naïve Bayes*, *Naïve Bayes* adalah Naive Bayes adalah salah satu algoritma klasifikasi yang populer dalam pembelajaran mesin. Algoritma ini berdasarkan pada teorema Bayes dengan asumsi bahwa setiap fitur dalam data adalah independen satu sama lain, meskipun ini seringkali tidak benar dalam situasi dunia nyata (Tarigan et al., 2023).

2.5. Evaluasi Model

Untuk mengukur evaluasi dari performa ketiga algoritma machine learning, teknik yang digunakan adalah teknik *Confusion Matrix*. *Confusion Matrix*, adalah cara tabel untuk memvisualisasikan kinerja model prediksi pada pembelajaran supervised learning. Setiap data dari masing-masing kelas dalam tabel confusion matrix menunjukkan jumlah prediksi yang dibuat guna untuk mengklasifikasikan kelas yang benar atau salah [18]. Teknik ini digunakan untuk menghitung nilai akurasi, presisi, dan recall. Tabel *Confusion Matrix* dapat dilihat pada Tabel 2.

Tabel 2. Confusion Matrix

Aktual	Prediksi	
	True	False
True	TP	FP
False	FN	TN

Keterangan:

- *True Positive (TP)*, ini adalah jumlah dari satu kelas TRUE yang bisa di prediksi dengan benar pada kelas TRUE.

- *True Negative* (TN), adalah jumlah dari satu kelas FALSE yang bisa di prediksi dengan benar pada kelas FALSE
- *False Positive* (FP), ini adalah kondisi dimana kelas TRUE yang prediksinya salah pada kelas FALSE, sedangkan
- *False Negatif* (FN), adalah dimana kondisi pada kelas FALSE yang di prediksi salah pada kelas TRUE.

1. *Accuracy*, ini adalah ukuran kinerja yang akan memberikan tingkat keakuratan dari keseluruhan model atau dalam penjelasan lain adalah menghitung semua prediksi yang benar dari total jumlah data. Berikut persamaan *accuracy*:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \times 100 \quad (1)$$

2. *Presisi*, adalah ukuran kinerja yang akan memberi informasi dari prediksi sebagai kelas positif yang sebenarnya positif. Berikut persamaan presisi :

$$Presisi = \frac{TP}{(TP + FP)} \times 100 \quad (2)$$

3. *Recall*, adalah ukuran kinerja yang memberi informasi dari prediksi kelas positif yang di prediksi negatif. Berikut persamaan *recall*:

$$Recall = \frac{TP}{(TP + FN)} \times 100 \quad (3)$$

3. Hasil dan Pembahasan

A. Perhitungan Manual

Pada subbab akan dicari proses perhitungan ketiga algoritma secara manual dengan menggunakan sampel pengujian beriktu ini. Teks yang akan diuji adalah sebagai berikut:

- Teks 1: "niat beli ev murah"
- Teks 2: "cerdas orang dan pasar"
- Teks 3: "dekarbonisasi bangun pltu"
- Teks 4: "pegawai negeri alih kendaraan"

Label yang diberikan untuk masing-masing teks adalah sebagai berikut : [0, 0, 1, 1] (0 untuk sentimen negatif, 1 untuk sentimen positif). Di bawah ini akan dilakukan proses perhitungan secara manual dengan menggunakan ketiga algoritma yang akan diuji. Tetapi sebelum proses pengujian, akan dilakukan proses preprosesing untuk mendapatkan hasil yang optimal dengan menggunakan metode TF-I

- Langkah 1: Praproses teks
 Lowercasing: Semua teks sudah dalam huruf kecil.
 Tokenisasi: Memecah teks menjadi kata-kata individu.
- Langkah 2: Membuat Vocabulary

Ekstraksi kata-kata unik dari dataset untuk membuat vocabulary. Vocabulary: ["niat", "beli", "ev", "murah", "cerdas", "orang", "dan", "pasar", "dekarbonisasi", "bangun", "pltu", "pegawai", "negeri", "alih", "kendaraan"]

- Langkah 3: Hitung Term Frequency (TF)
 - Hitung kemunculan setiap kata dalam setiap dokumen teks 1

Tabel 3. Kemunculan Kata di Dokumen Teks 1

Kata	Jumlah Kemunculan
niat	1
beli	1
env	1
murah	1
cerdas	0
orang	0
dan	0
pasar	0
dekarbonisasi	0
bangun	0
pltu	0
pegawai	0
negeri	0
alih	0
kendaraan	0

- Hitung kemunculan setiap kata dalam setiap dokumen teks 2

Tabel 3. Kemunculan Kata di Dokumen Teks 2

Kata	Jumlah Kemunculan
niat	0
beli	0
env	0
murah	0
cerdas	1
orang	1
dan	1
pasar	1
dekarbonisasi	0
bangun	0
pltu	0
pegawai	0
negeri	0
alih	0
kendaraan	0

- Hitung kemunculan setiap kata dalam setiap dokumen teks 3

Tabel 4. Kemunculan Kata di Dokumen Teks 3

Kata	Jumlah Kemunculan
niat	0
beli	0
env	0
murah	0
cerdas	0
orang	0
dan	0
pasar	0
dekarbonisasi	1
bangun	1
pltu	1
pegawai	0
negeri	0
alih	0
kendaraan	0

- Hitung kemunculan setiap kata dalam setiap dokumen teks 4

Tabel 4. Kemunculan Kata di Dokumen Teks 4

Kata	Jumlah Kemunculan
niat	0
beli	0
env	0
murah	0
cerdas	0
orang	0
dan	0
pasar	0
dekarbonisasi	0
bangun	0
pltu	0
pegawai	1
negeri	1
alih	1
kendaraan	1

4. Hitung Inverse Document Frequency (IDF)

Total jumlah dokumen (N = 4)

IDF dihitung dengan rumus:

$$IDF(T) = \log\left(\frac{N}{df(t)}\right) \tag{4}$$

Perhitungan untuk IDF setiap kata

Tabel 5. Perhitungan IDF

Kata	df	IDF
niat	0	$\log\left(\frac{4}{1}\right) = \log(4)$
beli	0	$\log\left(\frac{4}{1}\right) = \log(4)$
env	0	$\log\left(\frac{4}{1}\right) = \log(4)$
murah	0	$\log\left(\frac{4}{1}\right) = \log(4)$
cerdas	1	$\log\left(\frac{4}{1}\right) = \log(4)$
orang	1	$\log\left(\frac{4}{1}\right) = \log(4)$
dan	1	$\log\left(\frac{4}{1}\right) = \log(4)$
pasar	1	$\log\left(\frac{4}{1}\right) = \log(4)$
dekarbonisasi	0	$\log\left(\frac{4}{1}\right) = \log(4)$
bangun	0	$\log\left(\frac{4}{1}\right) = \log(4)$
pltu	0	$\log\left(\frac{4}{1}\right) = \log(4)$
pegawai	0	$\log\left(\frac{4}{1}\right) = \log(4)$
negeri	0	$\log\left(\frac{4}{1}\right) = \log(4)$
alih	0	$\log\left(\frac{4}{1}\right) = \log(4)$
kendaraan	0	$\log\left(\frac{4}{1}\right) = \log(4)$

Untuk semua kata yang ada dalam vocabulary kita, nilai IDF adalah:

$$IDF = \log(4) = 1,386$$

d) Hitung TF-IDF

TF-IDF dihitung dengan mengalikan TF dengan IDF untuk setiap kata dalam dokumen.

a) Dokumen 1: "niat beli ev murah"

niat : $\frac{1}{4} \times 1,386 = 0,3465$

beli : $\frac{1}{4} \times 1,386 = 0,3465$

ev : $\frac{1}{4} \times 1,386 = 0,3465$

murah : $\frac{1}{4} \times 1,386 = 0,3465$

kata lainnya : 0

b) Dokumen 2: "cerdas orang dan pasar"

cerdas : $\frac{1}{4} \times 1,386 = 0,3465$

orang : $\frac{1}{4} \times 1,386 = 0,3465$

dan : $\frac{1}{4} \times 1,386 = 0,3465$

pasar : $\frac{1}{4} \times 1,386 = 0,3465$

kata lainnya : 0

c) Dokumen 3: "dekarbonisasi bangun pltu"

dekarbonisasi : $\frac{1}{3} \times 1,386 = 0,462$

bangun : $\frac{1}{3} \times 1,386 = 0,462$

pltu : $\frac{1}{3} \times 1,386 = 0,462$

kata lainnya : 0

d) Dokumen 3: "pegawai negeri alih kendaraan"

pegawai : $\frac{1}{4} \times 1,386 = 0,3465$

negeri : $\frac{1}{4} \times 1,386 = 0,3465$

alih : $\frac{1}{4} \times 1,386 = 0,3465$

kendaraan : $\frac{1}{4} \times 1,386 = 0,3465$

kata lainnya : 0

Dibawah ini rekap dari hasil akhir TF-IDF untuk masing-masing dokumen

a. Hitung TF-IDF untuk dokumen teks 1

Tabel 6. Perhitungan TF-IDF dokumen teks 1

Kata	Jumlah Kemunculan
niat	0.347
beli	0.347
env	0.347
murah	0.347
cerdas	0
orang	0
dan	0
pasar	0
dekarbonisasi	0
bangun	0
pltu	0
pegawai	0
negeri	0
alih	0
kendaraan	0

b. Hitung kemunculan setiap kata dalam setiap dokumen teks 2

Tabel 7. Perhitungan TF-IDF dokumen teks 2

Kata	Jumlah Kemunculan
niat	0
beli	0
env	0
murah	0
cerdas	0.347
orang	0.347
dan	0.347
pasar	0.347
dekarbonisasi	0
bangun	0
pltu	0
pegawai	0
negeri	0
alih	0
kendaraan	0

c. Hitung kemunculan setiap kata dalam setiap dokumen teks 3

Tabel 8. Perhitungan TF-IDF dokumen teks 3

Kata	Jumlah Kemunculan
niat	0
beli	0
env	0
murah	0
cerdas	0
orang	0
dan	0

pasar	0
dekarbonisasi	0.462
bangun	0.462
pltu	0.462
pegawai	0
negeri	0
alih	0
kendaraan	0

d. Hitung kemunculan setiap kata dalam setiap dokumen teks 2

Tabel 9. Perhitungan TF-IDF dokumen teks 4

Kata	Jumlah Kemunculan
niat	0
beli	0
env	0
murah	0
cerdas	0
orang	0
dan	0
pasar	0
dekarbonisasi	0
bangun	0
pltu	0
pegawai	1
negeri	1
alih	1
kendaraan	1

Setelah mendapatkan hasil TF-IDF untuk masing-masing dokumen, akan dilakukan pengujian untuk masing-masing algoritma:

1. Algoritma Support Vector Machine

Untuk kesederhanaan, akan digunakan contoh dua dimensi (dua fitur) untuk menggambarkan konsep dasar SVM. Misalkan kita hanya menggunakan dua fitur pertama (niat dan beli):

Tabel 10. Algoritma SVM

Dokumen	Niat	Beli
Teks 1	0.347	0.347
Teks 2	0	0
Teks 3	0	0
Teks 4	0	0

Dengan dua fitur ini, bisa dibedakan Teks 1 dari yang lainnya. Misalkan kita memiliki dua kelas (negatif = 0, positif = 1), kita bisa menggambar garis (hiperplane) yang memisahkan Teks 1 dari dokumen lainnya. Untuk perhitungan manual SVM, langkah-langkahnya melibatkan:

- a) Pilih fungsi kernel (misalnya, linear kernel).
- b) Hitung margin maksimal antara dua kelas.
- c) Menentukan *support vectors* yang berada di tepi margin.

Untuk menghitung klasifikasi dengan SVM secara manual, kita menggunakan rumus:

$$f(x) = w \cdot x + b \tag{5}$$

Di mana:

- w adalah vektor bobot.
- x adalah vektor fitur.
- b adalah bias.

Misalkan akan digunakan SVM dengan linear kernel, kita perlu menghitung w dan b. Misalkan kita menemukan bahwa bobot w dan bias b adalah sebagai berikut (untuk contoh sederhana):

$$w = [1.1]$$

$$b = -0.5$$

Hitung Keputusan untuk setiap dokumen:

- 1) Teks 1:
 $f(x_1) = (1 \times 0.347) + (1 \times 0.347) - 0.5 = 0.194$
Hasil: Positif(1)
- 2) Teks 2:
 $f(x_2) = (1 \times 0) + (1 \times 0) - 0.5 = -0.5$
Hasil: Negatif(0)
- 3) Teks 3:
 $f(x_2) = (1 \times 0) + (1 \times 0) - 0.5 = -0.5$
Hasil: Negatif(0)
- 4) Teks 3:
 $f(x_2) = (1 \times 0) + (1 \times 0) - 0.5 = -0.5$
Hasil: Negatif(0)

2. Algoritma Random Forest

Pelatihan dan pembuatan pohon keputusan sangat kompleks untuk dilakukan secara manual dan harus menggunakan bahasa komputasi untuk menyelesaikan masalah tersebut namun ada langkah konseptual untuk mengerjakan perhitungan manual sederhana yang terjadi dalam Random Forest:

- a) Bootstrap Sampling: Mengambil sampel acak dari data pelatihan dengan penggantian (sampling with replacement).
- b) Pembangunan Pohon: Untuk setiap sampel bootstrap, membangun pohon keputusan dengan menggunakan subset acak dari fitur
- c) Voting: Menggabungkan prediksi dari semua pohon untuk menentukan kelas akhir.

Untuk contoh manual ini, akan diprediksi kelas dari dokumen baru berdasarkan hasil voting mayoritas dari beberapa pohon keputusan. Berikut adalah contoh bagaimana kita bisa melakukannya dengan data sederhana (dengan hanya dua pohon):

Contoh Penggunaan Dua Pohon Keputusan

Pohon 1:

- a) Menggunakan fitur [niat, beli]
- b) Jika niat > 0,2 dan beli > 0,2, prediksi adalah 0 (negatif), sebaliknya 1 (positif).

Pohon 2:

- a) Menggunakan fitur [dekarbonisasi, bangun]
- b) Jika dekarbonisasi > 0,3 dan bangun > 0,3, prediksi adalah 1 (positif), sebaliknya 0 (negatif).

Misalkan kita ingin memprediksi kelas untuk dokumen berikut: "niat beli ev murah"

- a. Prediksi dari Pohon 1:
Fitur: niat = 0.347, beli = 0.347
Keputusan: niat > 0.2 dan beli > 0.2, maka prediksi = 0 (negatif)
- b. Prediksi dari Pohon 2:
Fitur: dekarbonisasi = 0, bangun = 0
Keputusan: dekarbonisasi <= 0.3 atau bangun <= 0.3, maka prediksi = 0 (negatif)

Voting Hasil:

Pohon 1: Negatif (0)
Pohon 2: Negatif (0)
Mayoritas: Negatif (0)
Sehingga, dokumen "niat beli ev murah" diklasifikasikan sebagai negatif (0).

3. Algoritma Naive Bayes

Langkah-langkah perhitungan manual sederhana dengan menggunakan algoritma Naive Bayes adalah sebagai berikut:

a. Menghitung Probabilitas Prior

Probabilitas prior adalah probabilitas dari setiap kelas dalam dataset:

$$P(0) = \text{Jumlah dokumen negatif} / \text{Total dokumen} = 2/4 = 0.5$$

$$P(1) = \text{Jumlah dokumen positif} / \text{Total dokumen} = 2/4 = 0.5$$

b. Menghitung Probabilitas Kondisional

Untuk setiap kata dalam vocabulary, kita perlu menghitung probabilitas kondisional bahwa kata tersebut muncul dalam dokumen positif atau negatif. dapat dihitung probabilitas kondisional untuk kata "niat" sebagai contoh:

Probabilitas Kondisional untuk kelas 0 (Negatif):
 $P(\text{niat}|0) = (\text{Jumlah kemunculan "niat" dalam dokumen negatif} + 1) / (\text{Total kata dalam dokumen negatif} + \text{Jumlah kata dalam vocabulary})$

$$P(\text{niat}|0) = (1 + 1) / (4 + 15) = 2 / 19 = 0.105$$

Probabilitas Kondisional untuk kelas 1 (Positif):
 $P(\text{niat}|1) = (\text{Jumlah kemunculan "niat" dalam dokumen positif} + 1) / (\text{Total kata dalam dokumen positif} + \text{Jumlah kata dalam vocabulary})$

$$P(\text{niat}|1) = (0 + 1) / (4 + 15) = 1 / 19 = 0.053$$

Lakukan perhitungan ini untuk setiap kata dalam vocabulary untuk kedua kelas.

c. Menghitung Probabilitas Posterior

Dengan menggunakan nilai yang sudah dihitung sebelumnya (misalkan semua nilai sama dengan nilai "niat"):

$$P(0|d) \propto 0.5 \times 0.105 \times 0.105 \times 0.105 \times 0.105$$

$$P(0|d) \propto 0.5 \times (0.105)^4$$

$$P(0|d) \propto 0.5 \times 1.22 \times 10^{-5}$$

$$P(0|d) \propto 6.1 \times 10^{-6}$$

Probabilitas Posterior untuk kelas 1 (Positif):

$$P(1|d) \propto P(1) \times P(\text{niat}|1) \times P(\text{beli}|1) \times P(\text{ev}|1) \times P(\text{murah}|1)$$

Dengan menggunakan nilai yang sudah dihitung sebelumnya (misalkan semua nilai sama dengan nilai "niat"):

$$P(0|d) \propto 0.5 \times 0.053 \times 0.053 \times 0.053 \times 0.053$$

$$P(0|d) \propto 0.5 \times (0.053)^4$$

$$P(0|d) \propto 0.5 \times 7.9 \times 10^{-7}$$

$$P(0|d) \propto 3.95 \times 10^{-7}$$

d. Menentukan Klasifikasi

Bandingkan probabilitas posterior untuk setiap kelas dan pilih kelas dengan probabilitas tertinggi.

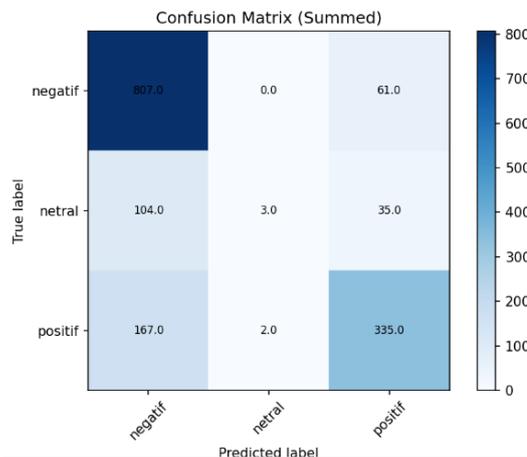
$$P(0|d) = 6.1 \times 10^{-6}$$

$$P(1|d) = 3.95 \times 10^{-7}$$

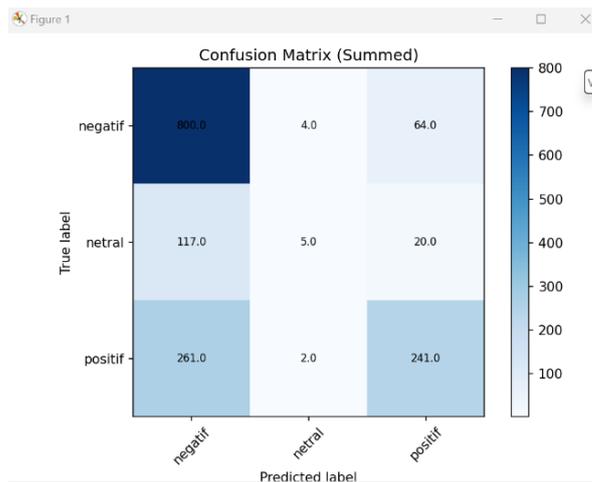
Sehingga, dokumen "niat beli ev murah" diklasifikasikan sebagai negatif

B. Pengujian Menggunakan Algoritma Python

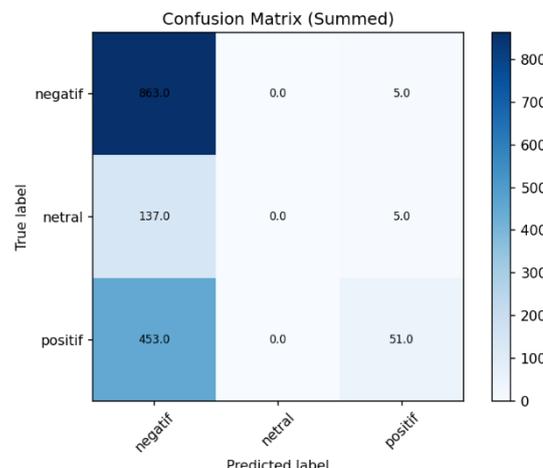
Pengujian algoritma bahasa pemrograman Python. Gambar 4, 5 & 6 menunjukkan hasil performa ketiga algoritma. Hasil performa ditampilkan dalam bentuk grafik dengan menggunakan library matplotlib di python.



Gambar 4. Hasil Performa Algoritma SVM



Gambar 5. Hasil Performa Algoritma Random Forest



Gambar 6. Hasil Performa Algoritma Naive Bayes

Berdasarkan Gambar 4, Gambar 4 menunjukkan hasil yang didapatkan oleh algoritma Support Vector Machine yang dapat dijelaskan di bawah ini:

a. Kelas "Negatif":

- 1) 807 sampel yang sebenarnya negatif diklasifikasikan dengan benar sebagai negatif (True Negatives).
 - 2) 0 sampel yang sebenarnya negatif diklasifikasikan salah sebagai netral.
 - 3) 61 sampel yang sebenarnya negatif diklasifikasikan salah sebagai positif
- b. Kelas "Netral":
- 1) 104 sampel yang sebenarnya netral diklasifikasikan salah sebagai negatif.
 - 2) 3 sampel yang sebenarnya netral diklasifikasikan dengan benar sebagai netral
 - 3) 35 sampel yang sebenarnya netral diklasifikasikan salah sebagai positif.
- c. Kelas "positif":
- 1) 167 sampel yang sebenarnya positif diklasifikasikan salah sebagai negatif.
 - 2) 2 sampel yang sebenarnya positif diklasifikasikan salah sebagai netral.
 - 3) 335 sampel yang sebenarnya positif diklasifikasikan dengan benar sebagai positif.

Dari hasil penjelasan di atas algoritma ini cukup baik dalam mengklasifikasikan sampel "negatif" dengan 807 prediksi benar dan hanya 61 prediksi salah sebagai "positif". Sedangkan dalam mengklasifikasikan sampel "netral" algoritma ini kurang baik dengan hanya 3 prediksi benar, sementara terdapat banyak kesalahan klasifikasi ke "negatif" (104) dan "positif" (35).

Untuk Gambar 5 menunjukkan hasil confusion Matrix untuk model algoritma Random Forest. Penjelasan dari Gambar 5 dapat dijelaskan sebagai berikut ;

- a. Kelas "Negatif":
- 1) 800 sampel yang sebenarnya negatif diklasifikasikan dengan benar sebagai negatif (True Negatives).
 - 2) 4 sampel yang sebenarnya negatif diklasifikasikan salah sebagai netral.
 - 3) 64 sampel yang sebenarnya negatif diklasifikasikan salah sebagai positif
- b. Kelas "Netral":
- 1) 117 sampel yang sebenarnya netral diklasifikasikan salah sebagai negatif.
 - 2) 5 sampel yang sebenarnya netral diklasifikasikan dengan benar sebagai netral
 - 3) 20 sampel yang sebenarnya netral diklasifikasikan salah sebagai positif.
- c. Kelas "positif":
- 1) 261 sampel yang sebenarnya positif diklasifikasikan salah sebagai negatif.
 - 2) 2 sampel yang sebenarnya positif diklasifikasikan salah sebagai netral.
 - 3) 241 sampel yang sebenarnya positif diklasifikasikan dengan benar sebagai positif.

Dari hasil penjelasan di atas algoritma ini cukup baik dalam mengklasifikasikan sampel "negatif"

dengan 800 prediksi benar dan hanya 68 prediksi salah sebagai "positif". Sedangkan dalam mengklasifikasikan sampel "netral" algoritma ini kurang baik dengan hanya 5 prediksi benar, sementara terdapat banyak kesalahan klasifikasi ke "negatif" (117) dan "positif" (20). Sedangkan untuk sentiment "positif" prediksi benar menghasilkan 241, namun ada cukup banyak kesalahan klasifikasi ke "negatif" (261) dan sedikit ke "netral" (2).

Gambar 6 menunjukkan hasil confusion matrix untuk algoritma Naïve Bayes. Penjelasan dari Gambar 6 dapat dijelaskan sebagai berikut:

- a. Kelas "Negatif":
- 1) 863 sampel yang sebenarnya negatif diklasifikasikan dengan benar sebagai negatif (True Negatives).
 - 2) 0 sampel yang sebenarnya negatif diklasifikasikan salah sebagai netral.
 - 3) 5 sampel yang sebenarnya negatif diklasifikasikan salah sebagai positif
- b. Kelas "Netral":
- 1) 137 sampel yang sebenarnya netral diklasifikasikan salah sebagai negatif.
 - 2) 0 sampel yang sebenarnya netral diklasifikasikan dengan benar sebagai netral
 - 3) 5 sampel yang sebenarnya netral diklasifikasikan salah sebagai positif.
- c. Kelas "positif":
- 1) 453 sampel yang sebenarnya positif diklasifikasikan salah sebagai negatif.
 - 2) 0 sampel yang sebenarnya positif diklasifikasikan salah sebagai netral.
 - 3) 51 sampel yang sebenarnya positif diklasifikasikan dengan benar sebagai positif.

Dari hasil penjelasan di atas algoritma ini cukup baik dalam mengklasifikasikan sampel "negatif" dengan 863 prediksi benar dan hanya 5 prediksi salah sebagai "positif". Sedangkan dalam mengklasifikasikan sampel "netral" algoritma ini kurang baik dengan hanya 5 prediksi benar, sementara terdapat banyak kesalahan klasifikasi ke "negatif" (137) dan "positif" (5). Sedangkan untuk sentiment "positif" prediksi benar menghasilkan 51 prediksi benar dan kesalahan negative sebanyak 453

Setelah percobaan selesai, langkah selanjutnya adalah melakukan evaluasi performa dengan mencari nilai akurasi, presisi, dan recall masing-masing algoritma. Hasil evaluasi ketiga algoritma dapat dilihat pada perhitungan di bawah ini

1. Algoritma Support Vector Machine

a. Perhitungan Akurasi

$$\text{Jumlah prediksi benar} = 807 \text{ (negatif)} + 3 \text{ (netral)} + 335 \text{ (positif)} = 1145$$

$$\text{Total jumlah sampel} = 807 + 0 + 61 + 104 + 3 + 35 + 167 + 2 + 335 = 1514$$

$$\text{Akurasi} = \frac{1145}{1514} = 0.756 * 100\% = 75.6$$

b. Perhitungan Presisi dan Recall

Kelas Negatif:

True Positif = 807
 False Positif = 167 + 104 = 271
 False Negatif = 61

$$Presisi(Negatif) = \frac{807}{807 + 271} = 0.748$$

$$Recall(Negatif) = \frac{807}{807 + 61} = 0.93$$

Kelas Netral:
 True Positif = 3
 False Positif = 2+0 = 2
 False Negatif = 104 + 35 = 139

$$Presisi(Neutral) = \frac{3}{3 + 2} = 0.6$$

$$Recall(Neutral) = \frac{3}{3 + 139} = 0.021$$

Kelas Positif:
 True Positif = 335
 False Positif = 61+35 = 96
 False Negatif = 167

$$Presisi(Positif) = \frac{335}{335 + 96} = 0.77$$

$$Recall(Positif) = \frac{335}{335 + 167} = 0.667$$

Sehingga nilai presisi dan Recall yaitu:

$$Presisi = \frac{0.748 + 0.6 + 0.7}{3} = \frac{2,125}{3} = 0.708 * 100\% = 70.8\%$$

$$Recall = \frac{0.93 + 0.021 + 0.667}{3} = \frac{1,618}{3} = 0.539 * 100\% = 53,9\%$$

2. Algoritma Random Forest

a. Perhitungan Akurasi
 Jumlah prediksi benar = 800 (negatif) + 5 (netral) + 241 (positif) = 1046
 Total jumlah sampel = 800 + 4 + 64 + 117 + 5 + 20 + 261 + 2 + 241 = 1046
 $Akurasi = \frac{1046}{1514} = 0.691 * 100\% = 69,1$

b. Perhitungan Presisi dan Recall
 Kelas Negatif :
 True Positif = 800
 False Positif = 256 + 117 = 378
 False Negatif = 64

$$Presisi(Negatif) = \frac{800}{800 + 378} = 0.679$$

$$Recall(Negatif) = \frac{800}{800 + 64} = 0.926$$

Kelas Netral :
 True Positif = 5
 False Positif = 2+4 = 6
 False Negatif = 117 + 20 = 137

$$Presisi(Neutral) = \frac{5}{5 + 6} = 0.455$$

$$Recall(Neutral) = \frac{5}{5 + 137} = 0.035$$

Kelas Positif :
 True Positif = 241
 False Positif = 64+20 = 84
 False Negatif = 261

$$Presisi(Positif) = \frac{241}{241 + 84} = 0.742$$

$$Recall(Positif) = \frac{241}{241 + 261} = 0.480$$

Sehingga nilai presisi dan Recall yaitu:

$$Presisi = \frac{0.679 + 0.455 + 0.742}{3} = \frac{1,876}{3} = 0.625 * 100\% = 62.5\%$$

$$Recall = \frac{0.926 + 0.035 + 0.48}{3} = \frac{1,441}{3} = 0.48 * 100\% = 48\%$$

3. Algoritma Naïve Bayes

a. Perhitungan Akurasi
 Jumlah prediksi benar = 863 (negatif) + 0 (netral) + 51 (positif) = 914
 Total jumlah sampel = 863 + 0 + 5 + 137 + 0 + 5 + 453 + 0 + 51 = 1514
 $Akurasi = \frac{914}{1514} = 0.603 * 100\% = 60,3$

b. Perhitungan Presisi dan Recall
 Kelas Negatif:
 True Positif = 863
 False Positif = 453 + 137 = 590
 False Negatif = 5

$$Presisi(Negatif) = \frac{863}{863 + 590} = 0.594$$

$$Recall(Negatif) = \frac{863}{863 + 5} = 0.994$$

Kelas Netral:
 True Positif = 0
 False Positif = 0+0 = 0
 False Negatif = 137 + 5 = 142
 Presisi(Neutral) = 0

$$Recall(Neutral) = \frac{0}{0 + 142} = 0$$

Kelas Positif :
 True Positif = 51
 False Positif = 5+5 = 10
 False Negatif = 453

$$Presisi(Positif) = \frac{51}{51 + 10} = 0.836$$

$$Recall(Positif) = \frac{51}{51 + 453} = 0.101$$

Sehingga nilai presisi dan Recall yaitu:

$$Presisi = \frac{0.594 + 0 + 0.836}{3} = \frac{1,43}{3} = 0.477 * 100\% = 47.7\%$$

$$Recall = \frac{0.994 + 0 + 0.101}{3} = \frac{1,095}{3} = 0.365 * 100\% = 36,5\%$$

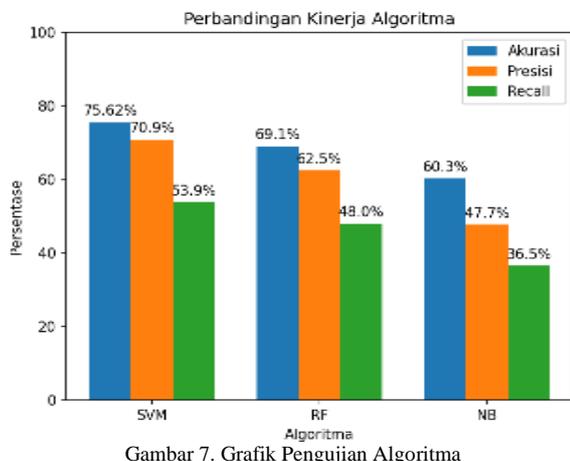
Rekap perhitungan akurasi, presisi, dan recall untuk ketiga algoritma dapat dilihat pada Tabel 3

Tabel 3. Hasil Perhitungan Confusion Matrix

Algoritma	Parameter Pengujian		
	Akurasi	Presisi	Recall
SVM	75,62%	70,9%	53,9%
RF	69,1 %	62,5%	48%
NB	60,3%	47,7%	36,5%

Dalam bentuk grafik, perbandingan masing-masing algoritma ditampilkan pada Gambar 7. Perbandingan nilai akurasi dari kinerja masing-masing algoritma yaitu Support Vector Machine sebesar 75,62%, Random Forest sebesar 69,1%, dan

Naïve Bayes sebesar 60,3%. Algoritma Support Vector Machine memiliki nilai akurasi lebih tinggi dibandingkan dengan algoritma yang lain, artinya algoritma ini mampu mengklasifikasi dengan baik dan maksimal daripada kedua algoritma lainnya.



Gambar 7. Grafik Pengujian Algoritma

Hal ini berlaku juga untuk nilai recall, algoritma Support Vector Machine lebih unggul dari Naïve Bayes dan Random Forest. Algoritma Support Vector Machine memiliki nilai recall sebesar 53,9%, Random Forest memiliki nilai recall sebesar 48%, dan Naïve Bayes memiliki nilai recall sebesar 36,5%.

Hal ini berlaku juga untuk nilai recall, algoritma Support Vector Machine lebih unggul dari Naïve Bayes dan Random Forest. Algoritma Support Vector Machine memiliki nilai recall sebesar 70,9%, Random Forest memiliki nilai recall sebesar 62,5%, dan Naïve Bayes memiliki nilai recall sebesar 47,7%.

4. Kesimpulan

Dalam penelitian ini telah dilakukan 3 eksperimen perbandingan kinerja algoritma machine learning, yaitu algoritma Support Vector Machine, algoritma Random Forest, dan algoritma Naïve Bayes dalam melakukan analisis sentiment tentang keberadaan mobil Listrik di Indonesia dikarenakan terjadi perbedaan pemahaman yang terjadi di kalangan Masyarakat terkait keberadaan mobil Listrik di Indonesia. Data yang digunakan bersumber dari media sosial Twitter/X dengan jumlah data yang dijadikan bahan penelitian berjumlah 1518 data dengan rincian 870 data dengan label negatif, 143 untuk label netral, dan 502 untuk label positif. Hasil dapat diamati bahwa Nilai akurasi tertinggi didapatkan oleh algoritma Support Vector Machine sebesar 75,62%, Nilai presisi tertinggi didapatkan oleh algoritma Support Vector Machine sebesar 70,9% dan nilai recall tertinggi adalah algoritma Support Vector Machine dengan nilai 53,88%. Dari data ini dapat disimpulkan algoritma terbaik yaitu Algoritma Support Vector Machine

Daftar Pustaka:

Aria, M., Cuccurullo, C., & Gnasso, A. (2021). A

comparison among interpretative proposals for Random Forests. *Machine Learning with Applications*, 6(June), 100094. <https://doi.org/10.1016/j.mlwa.2021.100094>

Audrey, O., Ratnawati, D. E., & Arwani, I. (2022). Analisis Sentimen Pengguna Twitter Terhadap Opini Non Fungible Token di Indonesia Menggunakan Algoritma Random Forest Classifier. 6(12), 5889–5897. <http://j-ptiik.ub.ac.id>

Cam, H., Cam, A. V., Demirel, U., & Ahmed, S. (2024). Sentiment analysis of financial Twitter posts on Twitter with the machine learning classifiers. *Heliyon*, 10(1), e23784. <https://doi.org/10.1016/j.heliyon.2023.e23784>

Chai, X., Xu, S., Li, S., & Zhao, J. (2023). The Process and Algorithm Analysis of Text Mining System Based on Artificial Intelligence. *Procedia Computer Science*, 228, 574–581. <https://doi.org/10.1016/j.procs.2023.11.066>

Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional. *Jurnal Tekno Kompak*, 15(1), 131. <https://doi.org/10.33365/jtk.v15i1.744>

Fitri, V. A., Andreswari, R., & Hasibuan, M. A. (2019). Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm. *Procedia Computer Science*, 161, 765–772. <https://doi.org/10.1016/j.procs.2019.11.181>

Garcia, K., & Berton, L. (2021). Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Applied Soft Computing*, 101, 107057. <https://doi.org/10.1016/j.asoc.2020.107057>

Lickert, H., Wewer, A., Dittmann, S., Bilge, P., & Dietrich, F. (2020). Selection of Suitable Machine Learning Algorithms for Classification Tasks in Reverse Logistics. *Procedia CIRP*, 96(March), 272–277. <https://doi.org/10.1016/j.procir.2021.01.086>

Motz, A., Ranta, E., Calderon, A. S., Adam, Q., Alzhouri, F., & Ebrahimi, D. (2022). Live Sentiment Analysis Using Multiple Machine Learning and Text Processing Algorithms. *Procedia Computer Science*, 203, 165–172. <https://doi.org/10.1016/j.procs.2022.07.023>

Nota, G., Postiglione, A., & Carvello, R. (2022). Text mining techniques for the management of predictive maintenance. *Procedia Computer Science*, 200, 778–792. <https://doi.org/10.1016/j.procs.2022.01.276>

Ramlan, R., Satyahadewi, N., & Andani, W. (2023). Analisis Sentimen Pengguna Twitter Menggunakan Support Vector Machine Pada Kasus Kenaikan Harga BBM. *Jambura Journal of Mathematics*, 5(2), 431–445.

- <https://doi.org/10.34312/jjom.v5i2.20860>
Romero, P. E., Rodriguez-Alabanda, O., Molero, E., & Guerrero-Vaca, G. (2021). Use of the support vector machine (SVM) algorithm to predict geometrical accuracy in the manufacture of molds via single point incremental forming (SPIF) using aluminized steel sheets. *Journal of Materials Research and Technology*, *15*, 1562–1571.
<https://doi.org/10.1016/j.jmrt.2021.08.155>
- Rosenberg, E., Tarazona, C., Mallor, F., Eivazi, H., Pastor-Escuredo, D., Fuso-Nerini, F., & Vinuesa, R. (2023). Sentiment analysis on Twitter data towards climate action. *Results in Engineering*, *19*(July), 101287.
<https://doi.org/10.1016/j.rineng.2023.101287>
- Ruan, T., & Lv, Q. (2023). Public perception of electric vehicles on Reddit and Twitter: A cross-platform analysis. *Transportation Research Interdisciplinary Perspectives*, *21*(January), 100872.
<https://doi.org/10.1016/j.trip.2023.100872>
- Sunitha, D., Patra, R. K., Babu, N. V., Suresh, A., & Gupta, S. C. (2022). Twitter sentiment analysis using ensemble based deep learning model towards COVID-19 in India and European countries. *Pattern Recognition Letters*, *158*, 164–170.
<https://doi.org/10.1016/j.patrec.2022.04.027>
- Tarigan, V. (2023). Pembuatan Aplikasi Data Mining Untuk Memperediksi Masa Studi Mahasiswa Menggunakan Algoritma Naive Bayes. *Informatika*, *11*(1), 54–62.
<https://doi.org/10.36987/informatika.v11i1.3847>
- Tarigan, V., Syahputra, R., Saputra, P. H., & Yusupa, A. (2023). *Seleksi Fitur Dengan Menggunakan Metode Entropy Pada Algoritma Klasifikasi Naive Bayes Untuk Penyakit Diabetes*. *8*(2), 66–77.
- Zola, G., Nugraheni, S. D., Rosiana, A. A., & ... (2023). Inovasi Kendaraan Listrik Sebagai Upaya Meningkatkan Kelestarian Lingkungan Dan Mendorong Pertumbuhan Ekonomi Hijau Di *Journal of Public ...*, *11*(3), 159–170.
<https://journal.student.uny.ac.id/index.php/joppar/article/view/20712%0Ahttps://journal.student.uny.ac.id/index.php/joppar/article/viewFile/20712/18383>

Halaman ini sengaja dikosongkan