JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

Perbandingan Resident Set Size dan Virtual Memory Size Algoritma Machine Learning dalam Analisis Sentimen

Reza Ardiansyah Yudhanegara[1,*], Nisrina Aliya Hana[1], Syahrizal Yonanda Mahfiridho[1], Aqwam Rosadi Kardian[2]

[1]Jurusan Kriptografi, Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara, Bogor, Indonesia
[2]Sistem Informasi, STMIK Jakarta STI&K, Jakarta Selatan, Indonesia
Email: [1,*]rezaardiansyahyudhanegara@gmail.com, [2]nisrinaahana@gmail.com, [3]syahrizalyonanda@gmail.com, [4]aqwam@staff.jak-stik.ac.id
Email Penulis Korespondensi: rezaardiansyahyudhanegara@gmail.com

Abstrak−Dalam era transformasi digital yang pesat, di mana data teks melimpah dari berbagai sumber online seperti media sosial, forum, dan ulasan produk, analisis sentimen telah menjadi komponen kritis dalam memahami dinamika opini publik dan perilaku konsumen. Analisis sentimen menggunakan machine learning, pemrosesan bahasa alami, dan linguistik komputasional untuk memahami perasaan dan opini orang lain. Algoritma machine learning yang diteliti pada tulisan ini antara lain K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naive Bayes, Iterative Dichotomiser Three (ID3), dan C4.5. Proses analisis sentimen membutuhkan sumber daya komputasi yang signifikan untuk menangani kompleksitas dan skala data. Penelitian ini bertujuan menguji perbedaan penggunaan sumber daya pada algoritma-algoritma tersebut dan menentukan algoritma manakah yang paling baik untuk digunakan dalam konteks analisis sentimen. Metode penelitian yang digunakan adalah metode kuantitatif dengan fokus pada pengumpulan dan analisis numerik pada dataset. Pengujian dilakukan dengan memanfaatkan Anaconda Library pada pemrograman berbahasa Python untuk mengukur penggunaan Resident Set Size (RSS), Virtual Memory Size (VMS), waktu eksekusi, serta akurasi program masing-masing algoritma. Hasil pengujian pada 10.652 data menunjukkan bahwa algoritma Support Vector Machine (SVM) dengan tingkat akurasi sebesar 96%, serta algoritma Naive Bayes dengan tingkat akurasi sebesar 97% adalah pilihan terbaik untuk digunakan dalam konteks analisis sentimen. Sedangkan apabila dilihat dari konteks penggunaan Resident Set Size (RSS) dan Virtual Memory Size (VMS) dalam 1 kali proses dijalankan, ID3 adalah algoritma dengan penggunaan sumber daya paling kecil dengan tingkat akurasi sebesar 92%. Rata-rata sumber daya yang digunakan oleh ID3 ialah sebesar 8.318.566,4 bytes untuk Resident Set Size (RSS) dan 7.965.900,8 bytes untuk Virtual Memory Size (VMS) dengan waktu eksekusi 2,619 detik.

Kata Kunci: Analisis Sentimen; Machine Learning; Resident Set Size; Virtual Memory Size

Abstract−In the rapidly advancing era of digital transformation, where textual data abounds from various online sources such as social media, forums, and product reviews, sentiment analysis has become a critical component in understanding public opinions and consumer behavior. Sentiment analysis employs machine learning, natural language processing, and computational linguistics to comprehend the feelings and opinions of others. The machine learning algorithms investigated in this paper include K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naive Bayes, ID3, and C4.5. The sentiment analysis process requires significant computational resources to handle the complexity and scale of data. This research aims to examine the differences in resource usage among these algorithms and determine which algorithm is best suited for sentiment analysis in this context. The research methodology employed is quantitative, focusing on the collection and numerical analysis of datasets. Testing is conducted using the Anaconda Library in the Python programming language to measure the usage of Resident Set Size (RSS), Virtual Memory Size (VMS), execution time, and the accuracy of each algorithm. The test results indicate that the Support Vector Machine (SVM) algorithm with an accuracy rate of 96% and the Naive Bayes algorithm with an accuracy rate of 97% are the best choices for use in the context of sentiment analysis. When considering the context of Resident Set Size (RSS) and Virtual Memory Size (VMS) usage in a single execution, ID3 is the algorithm with the smallest resource usage, with an accuracy rate of 92%. The average resources used by ID3 are 8.318.566,4 bytes for Resident Set Size (RSS) and 7.965.900,8 bytes for Virtual Memory Size (VMS) with an execution time of 2,619 seconds.

Keywords: Sentiment Analysis; Machine Learning; Resident Set Size; Virtual Memory Size

1. PENDAHULUAN

Analisis sentimen telah mengalami kemajuan signifikan dalam beberapa tahun terakhir. Analisis sentimen merupakan sebuah langkah untuk mengetahui opini terhadap suatu objek. Hal ini biasa dilakukan pada media sosial, seperti Twitter dan Facebook. Para peneliti telah menjelajahi berbagai dimensi analisis sentimen, termasuk klasifikasi polaritas, visualisasi sentimen, dan konstruksi leksikon sentimen khusus domain [1]. Tugas analisis sentimen telah diterapkan dalam beragam domain seperti media sosial, telemedicine, dan pendidikan, mencerminkan aplikasinya yang luas [2]–[4]. Tantangan dalam analisis sentimen bersifat kompleks, termasuk kebutuhan akan dataset berlabel besar, pengetahuan khusus domain, dan kompleksitas ekspresi sentimen dalam berbagai jenis kalimat [5]–[7].

Sudah banyak penelitian yang membahas terkait analisis sentimen ini. Beberapa penelitian telah membandingkan machine learning dan teknik analisis sentimen, menunjukkan keefektifan dari skema Ensemble Learning dan Supervised Machine Learning [8], [9]. Seperti pada penelitian yang dilakukan oleh Nasution dan Hayaty [10], dilakukan perbandingan akurasi dan waktu eksekusi algoritma K-Nearest Neighbor (K-NN) dan Support Vector Machine (SVM) pada analisis sentimen media sosial Twitter.

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

Hasil dari penelitian tersebut menunjukkan bahwa algoritma K-Nearest Neighbor memiliki kecenderungan untuk bekerja lebih baik. Penelitian lain juga dilakukan dengan membandingkan nilai akurasi antara metode Naive Bayes, K-NN, dan Decision Tree untuk melakukan penilaian analisis sentimen pada PT PAL Indonesia di sosial media Twitter dan menghasilkan kesimpulan bahwa metode Naive Bayes menjadi yang paling akurat dari dua metode lainnya dengan tingkat akurasi tertinggi [11].

Selanjutnya pada pernelitian yang dilakukan pada aplikasi Ruang Guru di Twitter menunjukkan hasil akurasi terbaik didapatkan oleh algoritma Support Vector Machine jika dibandingkan dengan algoritma Naïve Bayes dan K-Nearest Neighbour [12]. Meskipun demikian, terdapat faktor-faktor lain yang sama pentingnya dengan akurasi dan waktu eksekusi algoritma sebagai faktor pembanding dalam konteks analisis sentimen.

Analisis sentimen membutuhkan sumber daya komputasi yang signifikan untuk menangani kompleksitas dan skala data. Sumber daya komputasi yang digunakan untuk tujuan analisis sentimen menjadi krusial jika ditujukan untuk melihat efisiensi dan kinerja algoritma, kecepatan waktu eksekusi, dan memfasilitasi inferensi model dengan cepat.

Kapasitas memori yang memadai juga menjadi faktor penting, terutama dalam menyimpan parameter model dan memproses data dalam skala besar. Dalam hal penggunaan energi, pengoptimalan algoritma dapat membantu mengurangi beban daya, menciptakan analisis sentimen yang lebih efisien secara energi. Penggunaan sumber daya komputasi yang dapat didistribusikan (scalable) juga memainkan peran penting dalam meningkatkan kecepatan dan efisiensi analisis sentimen.

Pengukuran sumber daya komputasi salah satunya dapat dilakukan dengan menganalisis nilai Resident Set Size (RSS) dan Virtual Memory Size (VMS) yang digunakan oleh masing-masing algoritma saat dijalankan. Resident Set Size (RSS) merujuk pada jumlah memori fisik yang digunakan oleh suatu proses. Hal ini mencakup kode, data, dan pustaka bersama yang berada di memori. Window size yang optimal untuk berbagai aplikasi dan proses sangat krusial untuk manajemen memori yang efisien.

Window size merujuk pada sejumlah data atau area memori yang diperhitungkan dalam pengukuran atau manajemen memori fisik. Beberapa penelitian telah menunjukkan bahwa window size optimal bervariasi tergantung pada aplikasi atau beban kerja tertentu. Sebagai contoh, dalam konteks prediksi variabel struktural hutan berbasis penginderaan jauh, window size optimal ditemukan sebesar 400 m^2 untuk beberapa variabel tertentu [13]. Christou dkk. mengeksplorasi dampak window size yang berbeda pada akurasi klasifikasi sinyal EEG, dan menemukan bahwa ukuran jendela yang besar sekitar 21 detik berdampak positif pada akurasi 4 metode klasifikasi machine learning [14].

Sehingga saat menggunakan model machine learning untuk menganalisis sentimen dalam data teks sekuensial, pemilihan window size dapat mempengaruhi sejauh mana konteks dipertimbangkan. Selain itu, window size yang lebih besar atau lebih kecil dapat mempengaruhi konsumsi memori dan kinerja keseluruhan aplikasi, termasuk RSS.

Virtual Memory Size (VMS) digunakan untuk memantau penggunaan memori di sistem komputer dan merupakan salah satu aspek kritis dalam menetapkan kinerja sistem yang efisien. Sistem operasi Windows mendukung memori virtual yang memungkinkannya untuk menangani lebih banyak proses daripada sistem yang hanya menggunakan RAM, sehingga lebih efisien dalam mengelola aplikasi berukuran besar [15].

Manajemen memori virtual dalam sistem machine learning sangat penting untuk mengoptimalkan alokasi sumber daya dan mengatasi masalah terkait memori. Aplikasi machine learning seringkali membutuhkan manajemen memori yang efisien untuk menjamin kinerja tinggi dan keandalan. Selain itu, penyebaran framework machine learning yang dirancang untuk unit pemrosesan neural dalam memori (in-memory neural processing units), seperti yang dijelaskan oleh Jeon dkk. [16], dan optimalisasi sirkuit regresi linear dalam memori dengan array memristor, seperti yang ditunjukkan oleh Wang dkk. [17], menunjukkan peningkatan fokus pada pemanfaatan sumber daya memori untuk tugas-tugas machine learning.

Penelitian ini bertujuan untuk menentukan algoritma machine learning terbaik berdasarkan akurasi yang dikaitkan pada penggunaan sumber daya komputasi dan waktu eksekusi dalam konteks analisis sentimen pada sistem operasi Windows. Penelitian ini memanfaatkan bahasa pemrograman Python, yang merupakan salah satu bahasa paling populer untuk komputasi ilmiah [18], serta menggunakan pustaka Anaconda dalam pengembangan model. Algoritma machine learning yang digunakan dalam penelitian ini antara lain K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naive Bayes, Iterative Dichotomiser Three (ID3), dan C4.5.

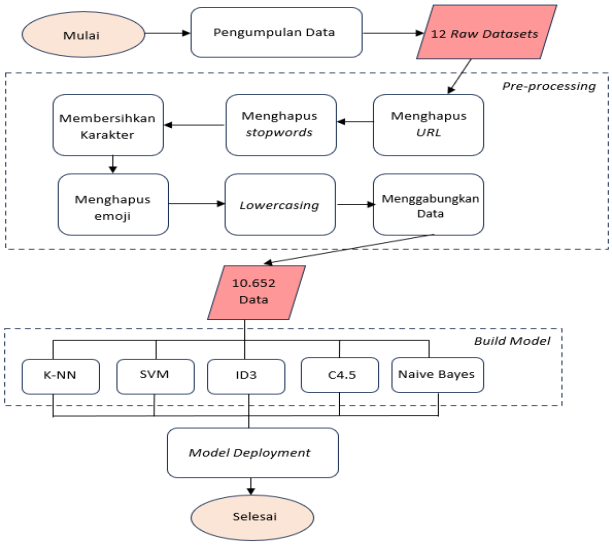
2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahap-tahapan penelitian dirancang seperti yang ditunjukkan pada Gambar 1. Berikut merupakan penjelasan mengenai tahapan penelitian yang dilakukan untuk mencapai tujuan penelitian.

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201



Gambar 1. Diagram Metode Penelitian

Secara garis besar, tahapan utama dalam metode penelitian yang dilakukan tersusun dari pengumpulan data, persiapan data (pre-processing), build model, dan model deployment. Terdapat beberapa proses yang diliputi oleh tahapan-tahapan tersebut.

2.2 Pengumpulan Data

Penelitian diawali dengan melakukan pengumpulan data. Sebanyak 12 dataset dipilih dari sumber-sumber internet yang relevan. Proses ini melibatkan analisis terperinci terhadap struktur dan variabel dalam setiap dataset, memastikan kualitas data yang diunduh. Dataset yang digunakan memiliki rincian jumlah data seperti dijabarkan pada Tabel 1.

Tabel 1. Jumlah Dataset

Nama Dataset	Jumlah Data
The Haven Bali Seminyak	2402
Primera Hotel Seminyak	2073
Ramada Encore by Wyndham Seminyak Bali	2073
Pelangi Bali Hotel	1201
Puri Saron Hotel	811
Kanishka Villas	559
Bali Rich Luxury Villa	387
Sense Sunset Seminyak	370
The Alea Hotel Seminyak	297
Bhavana Private Villas	235
Paragon Hotel Seminyak	171
Casa Dasa Boutique Hotel	73

2.3 Pre-Processing

Sebelum dataset diproses, langkah selanjutnya adalah melakukan pre-processing. Pada tahap ini bertujuan untuk membersihkan dataset dari noise yang mengganggu agar mudah diproses dan dianalisis. Pada tahap ini penulis melakukan pengolahan dataset. Hal yang dilakukan antara lain mengubahan semua huruf ke dalam huruf kecil (lowercasing), membersihkan karakter selain spasi dan angka, menghapus stopwords, menghapus URL, dan menghapus emoji. Hasil dari masing-masing dataset yang telah dibersihkan dijadikan menjadi satu data set dan mendapatkan total data sejumlah 10.652 buah. Penelitian ini menggunakan perangkat keras dan perangkat lunak dengan spesifikasi ditunjukkan pada Tabel 2.

Tabel 2. Spesifikasi Perangkat

Kategori	Keterangan
Prosesor	Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz, 2304 Mhz, 2 Core(s), 4 Logical Processor(s)
RAM	8GB
Hard Disk	HDD 1T
Sistem Operasi	Microsoft Windows 10 Pro Version 10.0.19045 Build 19045

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

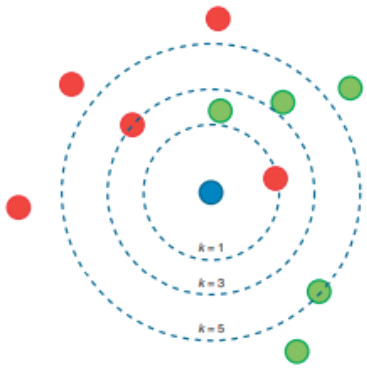
Kategori	Keterangan
Perangkat Lunak	1. Microsoft Excel 2016 2. Jupyter 3. Anaconda 3 4. Python Library meliputi scikit-learn, pandas, dan psutil. 5. Python Built-in Modules meliputi os dan time.

2.4 Build Model

Build model merupakan tahapan membuat program algoritma yang akan diuji. Build model dilakukan dengan bahasa pemrograman Python. Modul time digunakan untuk mengukur waktu eksekusi. Pustaka psutil digunakan untuk mengukur memori Resident Set Size (RSS) dan Virtual Memory Size (VMS) yang digunakan tiap algoritma. Modul os digunakan untuk menciptakan interaksi antara program yang dibuat dengan sistem operasi yang digunakan pada komputer. Sementara itu akurasi didapatkan dari pustaka scikit-learn yang juga memproses dataset pada masing-masing algoritma machine learning, dibantu oleh pusaka pandas. Algoritma machine learning yang akan diuji ialah K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naive Bayes, ID3, dan C4.5. Rumus serta cara kerja masing-masing algoritma dalam konteks analisis sentimen akan dijelaskan pada subbab selanjutnya.

2.4.1 Algoritma K-Nearest Neighbor (K-NN)

Algoritma K-Nearest Neighbors (K-NN) adalah metode klasifikasi machine learning klasik yang digunakan untuk kategorisasi data dan analisis sentimen [19]. Algoritma K-NN bekerja dengan mengidentifikasi k titik data terdekat (tetangga) dari suatu input tertentu, dan kemudian mengklasifikasikan input berdasarkan kelas mayoritas dari tetangga terdekatnya [20].



Gambar 2. Ilustrasi K-Nearest Neighbor

Gambar 2 mengilustrasikan metode klasifikasi di mana titik biru merupakan objek yang akan diprediksi kelasnya. Pada kasus ketika nilai k sama dengan 1, objek diprediksi masuk ke dalam kelas merah. Selanjutnya, dengan nilai k sama dengan 3, objek diprediksi masuk ke dalam kelas merah karena terdapat 2 tetangga merah dan 1 tetangga hijau. Ketika nilai k diperbesar menjadi 5, prediksi akan mengarah ke kelas hijau karena terdapat 3 tetangga hijau dan 2 tetangga merah [10].

Dalam konteks penelitian ini, data teks yang telah diperoleh akan diubah menjadi vektor. Dalam menghitung jarak antar vektor, metode yang umumnya digunakan adalah cosine similarity. Penggunaan metode ini dijelaskan sebagai cara yang efektif pada data teks yang bersifat sparse, karena memudahkan interpretasi pada data yang memiliki keberagaman kata-kata. Persamaan cosine similarity adalah:

$$\text{Scosine}(x, y) = \frac{x^T y}{\|x\|\|y\|} \quad (1)$$

x dan y adalah dua vektor dalam ruang yang memiki l dimensi, sedangkan x^T adalah transpos dari vektor x. Sedangkan untuk panjang dari vektor dapat ditulis sebagai berikut [21]:

$$\|x\| = \sqrt{\sum_{i=1}^{l} x_i^2} \quad (2)$$

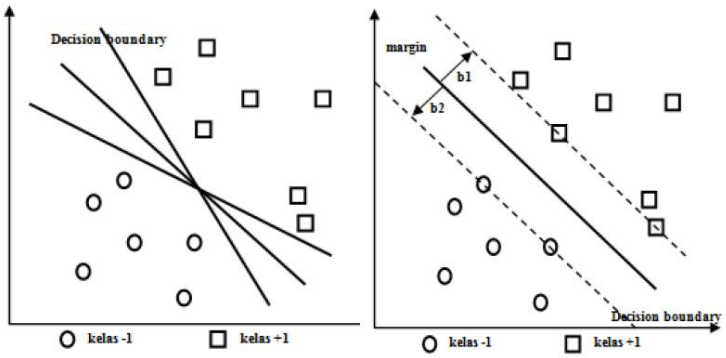
2.4.2 Algoritma Support Vector Machine (SVM)

Support Vector Machine (SVM) dapat memisahkan antara teks dengan sentimen positif dan negatif dengan mencari hyperplane (bidang pembatas) terbaik yang memisahkan kelas-kelas data. SVM juga dapat mengatasi masalah klasifikasi non-linear dan kompleks dengan menggunakan fungsi kernel untuk mentransformasikan data ke dimensi yang lebih tinggi. Dalam sebuah studi lain yang melakukan analisis sentimen di Twitter mengenai

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

penggunaan transportasi umum darat di kota-kota dengan menggunakan Support Vector Machine, hasil dari uji coba tersebut mencapai akurasi sebesar 78.12% [22].



Gambar 3. Ilustrasi Support Vector Machine

Gambar 3 memperlihatkan konsep dasar klasifikasi di mana pada diagram sebelah kiri terdapat dataset yang terbagi menjadi dua kelas, yaitu lingkaran sebagai kelas -1 dan kotak sebagai kelas +1. Beberapa hyperplane yang mungkin untuk mengklasifikasikan data juga ditampilkan dalam gambar tersebut. Pada diagram sebelah kanan, diperlihatkan hyperplane yang memiliki margin maksimal. Proses perhitungan hyperplane ini melibatkan pengukuran jarak margin dengan data terdekat dari masing-masing kelas. Data yang berperan sebagai titik pendukung dalam menentukan hyperplane disebut sebagai Support Vector. Prinsip utama dari metode SVM adalah mencari hyperplane paling optimal di antara berbagai kemungkinan yang ada [10].

Persamaan hyperplane dalam SVM dijelaskan oleh persamaan linear berikut. f(x) adalah fungsi keputusan dimana jika $f(x) > 0$, maka titik tersebut diklasifikasikan ke kelas positif. Sebaliknya, jika $f(x) < 0$, titik tersebut diklasifikasikan ke kelas negatif. Vektor bobot (weight) w dikalikan dengan vektor fitur input x lalu ditambahkan bias b akan menghasilkan persamaan hyperplane sebagai berikut:

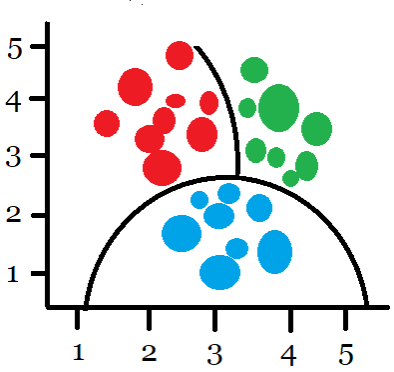
$$f(x) = w \cdot x + b \quad (3)$$

Rumus untuk menghitung margin maksimal antara hyperplane optimal dan hyperplane yang melewati support vector dapat dijabarkan sebagai berikut [10]:

$$\text{Margin} = \frac{2}{\|\vec{w}\|} \quad (4)$$

2.4.3 Algoritma Naive Bayes

Naive Bayes adalah algoritma klasifikasi yang menghitung probabilitas suatu set fitur tertentu termasuk dalam kelas tertentu. Algoritma ini didasarkan pada Teorema Bayes dan asumsi 'naif' tentang independensi kondisional antara setiap pasang fitur dengan nilai variabel kelas [23]. Algoritma ini dikenal karena kesederhanaan dan efektivitasnya, terutama dalam data berdimensi tinggi, karena mengestimasi probabilitas setiap fitur untuk bersifat independen [24]. Karena itu Naive Bayes tepat untuk diterapkan pada sentimen, yang merupakan jenis data berdimensi tinggi. Ilustrasi pengklasifikasian 3 kelompok kelas dengan algoritma Naive Bayes dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi Naive Bayes

Pada penelitian ini penulis menggunakan model Multinomial Naive Bayes. Sampel (vektor fitur) mencerminkan frekuensi di mana peristiwa tertentu dihasilkan oleh sebuah multinomial $p = (p_1, ..., p_n)$ dengan p_1 adalah probabilitas bahwa peristiwa i terjadi. $p(c \mid d)$ merupakan probabilitas data d berada di kelas c. $P(c)$

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

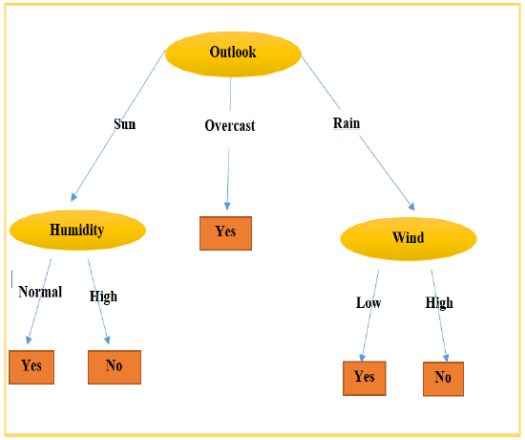
JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

merupakan prior probability suatu data berada di kelas c. Sebuah vektor fitur x = (x_1, ..., x_n) merupakan sebuah histogram, dengan x_i menghitung berapa kali peristiwa i diamati. $p(x_i|c)$ merupakan probabilitas bersyarat term x_i berada di dokumen pada kelas c. Ini adalah model peristiwa yang umumnya digunakan untuk klasifikasi dokumen, di mana peristiwa mewakili jumlah kemunculan kata dalam dokumen [25].

$$p(c\,|d\,) \propto p(c)\prod_{i=1}^{n} p(x_i|c) \quad (5)$$

2.4.4 Algoritma Iterative Dichotomiser 3 (ID3)

Algoritma ID3 (Iterative Dichotomiser 3) yang dikembangkan oleh Ross Quinlan, merupakan algoritma fundamental dalam bidang machine learning dan konstruksi tree decision. Algoritma ini menjadi dasar bagi algoritma C4.5, versi yang diperbarui dan dikembangkan oleh Quinlan pada tahun 1993[26]. ID3 beroperasi dengan membagi dataset secara rekursif berdasarkan atribut paling informatif di setiap simpul pohon. Pemilihan atribut ini dilakukan menggunakan pengukuran keuntungan informasi, yang bertujuan untuk memaksimalkan homogenitas variabel target dalam setiap partisi [27]. Algoritma ID3 dikenal karena kemampuannya untuk menangani atribut tidak numerik dan pencarian dari atas ke bawah dalam pembentukan pohon keputusan [28].



Gambar 5. Ilustrasi ID3

Pseudocode dari algoritma ini sangat sederhana. Diberikan suatu set atribut tidak target (C1, C2, ..., Cn), C sebagai atribut target, dan suatu himpunan S sebagai rekaman pembelajaran. Anggaplah kita ingin menggunakan algoritma ID3 untuk menentukan apakah saatnya untuk bermain bola. Selama dua minggu, data dikumpulkan untuk membantu membangun pohon keputusan ID3. Klasifikasi dari atribut target adalah 'haruskah kita bermain bola?' yang dapat berupa 'ya' atau 'tidak'. Atribut cuaca meliputi outlook, temperatur, kelembapan, dan kecepatan angin. Maka pohon keputusan algoritma ID3 akan terlihat seperti pada Gambar 5 [29].

Pada penelitian ini, penulis menggunakan algoritma ID3 dengan DecisionTreeClassifier dari modul pustaka pemrograman Python scikit-learn dengan kriteria entropy. Rumus matematis untuk perhitungan entropi (entropy) dalam ID3 adalah sebagai berikut:

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i) \quad (6)$$

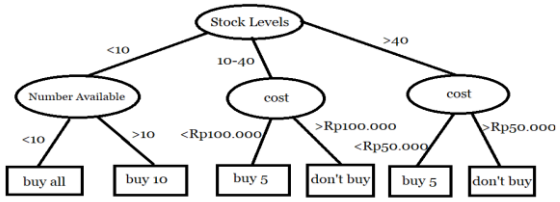
Di mana S adalah himpunan data dan p_i adalah proporsi jumlah sampel yang termasuk dalam kelas i terhadap total sampel, dengan jumlah kelas sebanyak c. Entropi adalah ukuran dari ketidakmurnian atau ketidakpastian dalam sebuah himpunan data. Dalam konteks klasifikasi data, entropi digunakan untuk mengukur seberapa tidak pastinya kelas-kelas dalam sebuah himpunan data. Semakin tinggi entropi, semakin tidak pasti atau tidak murni himpunan data tersebut. Entropi sering digunakan dalam menentukan bagaimana node bercabang di pohon keputusan [30]. Selain itu, information gain digunakan sebagai kriteria pemisah dalam ID3, yang terkait dengan konsep entropi [31].

2.4.5 Algoritma C4.5

Algoritma C4.5 adalah algoritma machine learning yang dikategorikan sebagai algoritma tree decision. Algoritma ini banyak digunakan untuk tugas klasifikasi dan dikenal karena kemampuannya untuk menangani data yang bersifat kontinu maupun diskrit, serta nilai atribut yang hilang. Algoritma ini bekerja dengan cara membagi data secara rekursif berdasarkan atribut untuk membuat struktur pohon, di mana setiap node internal mewakili uji pada suatu atribut, setiap cabang mewakili hasil uji, dan setiap node daun mewakili label kelas [32]. Selain itu, algoritma C4.5 sering digunakan dalam metode hibrida, seperti dikombinasikan dengan optimisasi gerombol partikel untuk menghasilkan tingkat kesiapan kelas yang tinggi dan akurat bagi mahasiswa [33]. Sebuah studi terkait analisis sentimen pandemi COVID-19 pada media sosial Twitter menunjukkan algoritma pohon keputusan seperti ID3 dan C4.5 memiliki akurasi yang lebih tinggi dibandingkan beberapa algoritma lainnya [34]. Gambar 6 mengilustrasikan metode pohon keputusan algoritma C4.5 terkait pembelian stok.

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201



Gambar 6. Ilustrasi C4.5

Pada penelitian ini, penulis menggunakan algoritma C4.5 dengan DecisionTreeClassifier dari modul pustaka pemrograman Python scikit-learn dengan kriteria gini. Rumus matematis untuk perhitungan gini dalam C4.5 adalah sebagai berikut:

$$\text{Gini}(D) = 1 - \sum_{i=1}^{K}(p_i)^2 \quad (7)$$

D adalah node dalam pohon keputusan, K adalah jumlah kelas, dan p_i adalah proporsi jumlah sampel yang termasuk dalam kelas i terhadap total sampel.

2.5 Model Deployment

Pada tahap model deployment, model setiap algoritma yang telah dibuat pada tahapan build model masing-masing dijalankan dengan input sebanyak 10.652 data yang didapatkan dari tahapan pre-processing. Tahapan ini akan menghasilkan nilai Resident Set Size, Virtual Memory Size, waktu eksekusi program dan akurasi dari masing-masing algoritma.

3. HASIL DAN PEMBAHASAN

Bagian ini menunjukkan hasil dari masing-masing program algoritma K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naive Bayes, ID3, dan C4.5.

3.1 Pengujian dan Perhitungan

Penulis menjalankan 5 algoritma yang sudah disebutkan sebelumnya untuk melihat berapa banyak nilai Resident Set Size dan Virtual Memory Size masing-masing algoritma. Selain itu, pengujian juga melibatkan perhitungan waktu eksekusi program dan akurasi yang diberikan oleh algoritma. Perhitungan Resident Set Size dan Virtual Memory Size dimulai pada saat program melakukan pemisahan train data dan test data, menjalankan algoritma kepada train data, dan melakukan tes kepada test data serta menampilkan hasil perhitungan alokasi memori dan akurasi dari algoritma.

Hasil dari 10 kali pengujian yang dilakukan pada algoritma K-NN menunjukkan hasil seperti yang ditunjukkan pada Tabel 3. Dapat disimpulkan bahwa akurasi yang didapatkan dari algoritma tersebut dalam melakukan analisis sentimen pada dataset menunjukkan nilai yang tidak tinggi, yaitu hanya sebesar 88%. Sementara itu penggunaan sumber daya terbilang cukup besar karena menyentuh angka di atas 10.000.000 bytes pada setiap iterasi. Namun waktu yang dibutuhkan untuk eksekusi program sangat cepat karena hanya membutuhkan waktu sekitar 2-2,5 detik.

Tabel 3. Hasil Algoritma K-NN

Percobaan Ke-	RSS (bytes)	VMS (bytes)	Waktu Eksekusi (detik)	Akurasi (%)
1	15.736.832	15.556.608	2,424	88
2	12.648.448	12.480.512	2,161	
3	11.096.064	11.284.480	2,053	
4	11.694.080	11.513.856	2,033	
5	12.926.976	12.890.112	2,048	
6	10.907.648	11.161.600	2,055	
7	13.008.896	12.427.264	2,060	
8	11.046.912	11.198.464	2,055	
9	13.590.528	13.561.856	2,034	
10	12.328.960	12.353.536	2,063	

Hasil dari 10 kali pengujian yang dilakukan pada algoritma SVM menunjukkan hasil seperti yang ditunjukkan pada Tabel 4. Dapat disimpulkan bahwa akurasi yang didapatkan dari algoritma tersebut dalam melakukan analisis sentimen pada dataset menunjukkan nilai yang tinggi, yaitu sebesar 96%. Sementara itu penggunaan sumber daya terbilang cukup besar karena pada iterasi pertama dan kedua berada pada angka di atas 10.000.000 bytes baik untuk RSS maupun VMS. Waktu yang dibutuhkan untuk eksekusi program juga sangat lambat karena membutuhkan waktu sekitar 19,6-22,7 detik.

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

Tabel 4. Hasil Algoritma SVM

Percobaan Ke-	RSS (bytes)	VMS (bytes)	Waktu Eksekusi (detik)	Akurasi (%)
1	12.283.904	12.427.264	19,898	96
2	12.926.976	12.828.672	19,816	
3	8.527.872	8.716.288	19,920	
4	9.293.824	9.650.176	20,632	
5	8.491.008	9.986.048	22,668	
6	9.773.056	12.337.152	19,773	
7	8.761.344	10.915.840	19,673	
8	8.929.280	7.925.760	19,632	
9	8.630.272	9.560.064	19,746	
10	8.605.696	10.727.424	19,678	

Hasil dari 10 kali pengujian yang dilakukan pada algoritma Naive Bayes menunjukkan hasil seperti yang ditunjukkan pada Tabel 5. Dapat disimpulkan bahwa akurasi yang didapatkan dari algoritma tersebut dalam melakukan analisis sentimen pada dataset menunjukkan nilai yang tinggi yaitu sebesar 97%. Sementara itu penggunaan sumber daya terbilang tidak terlalu besar karena sama sekali tidak menyentuh angka di atas 10.000.000 bytes. Waktu yang dibutuhkan untuk eksekusi program juga terbilang sangat cepat karena hanya membutuhkan waktu di bawah 1 detik.

Tabel 5. Hasil Algoritma Naive Bayes

Percobaan Ke-	RSS (bytes)	VMS (bytes)	Waktu Eksekusi (detik)	Akurasi (%)
1	9.269.248	8.712.192	0,87	97
2	8.544.256	7.258.112	0,753	
3	8.527.872	8.314.880	0,780	
4	8.888.320	8.392.704	0,78	
5	8.208.384	7.536.640	0,742	
6	8.908.800	9.043.968	0,77	
7	8.810.496	8.830.976	0,774	
8	9.117.696	9.375.744	0,790	
9	8.704.000	7.327.744	0,794	
10	8.110.080	7.888.896	0,731	

Hasil dari 10 kali pengujian yang dilakukan pada algoritma ID3 menunjukkan hasil seperti yang ditunjukkan pada Tabel 6. Dapat disimpulkan bahwa akurasi yang didapatkan dari algoritma tersebut dalam melakukan analisis sentimen pada dataset menunjukkan nilai yang cukup tinggi yaitu sebesar 92%. Sementara itu penggunaan sumber daya terbilang bervariasi karena terkadang menunjukkan nilai yang sangat besar dan terkadang menunjukkan nilai yang kecil. Pada iterasi pertama, performa algoritma menyentuh angka sebesar 11.112.448 bytes untuk RSS dan angka sebesar 10.641.408 untuk VMS. Namun pada iterasi ketiga dan kelima, algoritma ini menunjukkan nilai yang lebih kecil dibandingkan algoritma-algoritma sebelumnya yaitu berada di bawah 7.000.000 bytes. Waktu yang dibutuhkan untuk eksekusi program juga terbilang relatif cepat karena hanya membutuhkan waktu sekitar 2-4,1 detik.

Tabel 6. Hasil Algoritma ID3

Percobaan Ke-	RSS (bytes)	VMS (bytes)	Waktu Eksekusi (detik)	Akurasi (%)
1	11.112.448	10.641.408	2,305	92
2	9.637.888	9.203.712	2,039	
3	6.586.368	6.483.968	2,040	
4	8.470.528	7.249.920	2,371	
5	6.721.536	6.676.480	2,170	
6	8.388.608	7.909.376	2,291	
7	8.056.832	7.405.568	2,683	
8	7.475.200	7.622.656	3,360	
9	8.441.856	8.118.272	4,024	
10	8.294.400	8.347.648	2,908	

Hasil dari 10 kali pengujian yang dilakukan pada algoritma C4.5 menunjukkan hasil seperti yang ditunjukkan pada Tabel 7. Dapat disimpulkan bahwa akurasi yang didapatkan dari algoritma tersebut dalam melakukan analisis sentimen pada dataset menunjukkan nilai yang cukup tinggi yaitu sebesar 92%. Sementara itu penggunaan sumber daya terbilang bervariasi karena terkadang menunjukkan nilai yang sangat besar dan terkadang menunjukkan nilai yang kecil. Pada iterasi pertama, performa algoritma menyentuh angka sebesar 11.018.240 bytes untuk RSS dan angka sebesar 10.567.680 untuk VMS. Namun pada iterasi ketiga, algoritma ini

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

menunjukkan nilai yang kecil yaitu berada di bawah 7.000.000 bytes. Waktu yang dibutuhkan untuk eksekusi program juga terbilang relatif cepat karena hanya membutuhkan waktu sekitar 2,7-3 detik.

Tabel 7. Hasil Algoritma C4.5

Percobaan Ke-	RSS (bytes)	VMS (bytes)	Waktu Eksekusi (detik)	Akurasi (%)
1	11.018.240	10.567.680	2,759	
2	9.854.976	9.469.952	2,764	
3	6.799.360	6.676.480	2,908	
4	8.630.272	8.536.064	2,758	
5	7.970.816	8.085.504	2,741	92
6	8.368.128	7.970.816	2,870	
7	8.785.920	8.237.056	2,764	
8	8.617.984	8.265.728	2,710	
9	8.400.896	8.355.840	2,714	
10	8.536.064	8.605.696	2,750	

3.2 Perbandingan Penggunaan Sumber Daya

Dari hasil yang telah didapatkan sebelumnya, penulis melakukan perhitungan serta membandingkan rata-rata penggunaan memori fisik (RSS), memori virtual (VMS), waktu eksekusi serta akurasi masing-masing algoritma. Tabel 8 merupakan perbandingan rata-rata empat hal tersebut pada masing-masing algoritma.

Tabel 8. Perbandingan Rata-Rata Nilai Algoritma

Algoritma	Nilai Rata-Rata			Akurasi (%)
	RSS (bytes)	VMS (bytes)	Waktu Eksekusi (detik)	
Naïve Bayes	8.708.915,2	8.268.185,6	0,778	97
SVM	9.622.323,2	10.507.468,8	20,144	96
ID3	8.318.566,4	7.965.900,8	2,619	92
C4.5	8.698.265,6	8.477.081,6	2,774	92
K-NN	12.498.534,4	12.442.828,8	2,099	88

Identifikasi mengenai tingkat efisiensi penggunaan sumber daya memori dapat dilakukan dengan memperhatikan sejauh mana memori digunakan selama periode waktu tertentu. Maka dari itu, penentuan tingkat keefektifan algoritma didasarkan pada nilai Resident Set Size per waktu eksekusi dan Virtual Memory Size per waktu eksekusi. Tabel 9 memperlihatkan urutan peringkat kelima algoritma berdasarkan pada nilai Resident Set Size per waktu eksekusi. Dapat dilihat bahwa Naive Bayes menggunakan sumber daya komputasi RSS yang sangat besar dalam 1 detik waktu eksekusi program yaitu sebesar 11.193.978 bytes per waktu eksekusi. Sedangkan SVM hanya menggunakan sumber daya komputasi RSS sebesar 477.677 bytes untuk 1 detik waktu eksekusi program.

Tabel 9. Pengurutan Algoritma Berdasarkan RSS per Waktu Eksekusi

Urutan	Algoritma	RSS (bytes) per Waktu Eksekusi (detik)
1	Naïve Bayes	11.193.978
2	K-NN	5.954.529
3	ID3	3.176.238
4	C4.5	3.135.640
5	SVM	477.677

Tabel 10 memperlihatkan urutan peringkat kelima algoritma berdasarkan nilai Virtual Memory Size per waktu eksekusi. Dapat dilihat bahwa Naive Bayes menggunakan sumber daya komputasi VMS yang sangat besar dalam 1 detik waktu eksekusi program yaitu sebesar 10.627.488 bytes per waktu eksekusi. Sedangkan SVM hanya menggunakan sumber daya komputasi VMS sebesar 521.618 bytes untuk 1 detik waktu eksekusi program.

Tabel 10. Pengurutan Algoritma Berdasarkan VMS per Waktu Eksekusi

Urutan	Algoritma	VMS (bytes) per Waktu Eksekusi (detik)
1	Naïve Bayes	10.627.488
2	K-NN	5.927.979
3	C4.5	3.055.905
4	ID3	3.041.581
5	SVM	521.618

Setelah proses pengujian selesai dilakukan, dilakukan evaluasi dengan mencari hasil klasifikasi terbaik. Hasil klasifikasi terbaik salah satunya adalah dengan melihat algoritma yang memiliki tingkat akurasi tertinggi [35]. Maka dari itu, dapat diambil kesimpulan bahwa algoritma yang paling baik digunakan untuk melakukan

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

analisis sentimen ialah yang memiliki tingkat akurasi mendekati angka 100% yakni algoritma Naive Bayes dan Support Vector Machine.

Selain dilihat dari tingkat akurasinya, klasifikasi terbaik juga dilihat dari seberapa besar penggunaan sumber daya selama satu kali proses dijalankan. Jika dilihat dari sudut pandang tersebut, maka algoritma yang menggunakan memori fisik (RSS) dan memori virtual (VMS) paling besar ialah K-Nearest Neighbor, sedangkan ID3 adalah algoritma dengan penggunaan sumber daya paling kecil. Perhitungan juga dilakukan dengan membandingkan Resident Set Size per waktu eksekusi dan Virtual Memory Size per waktu eksekusi. Dengan demikian dapat diketahui algoritma yang memiliki nilai Resident Set Size per waktu eksekusi dan Virtual Memory Size per waktu eksekusi tertinggi adalah algoritma Naive Bayes, sedangkan algoritma Support Vector Machine merupakan kebalikannya.

Algoritma K-Nearest Neighbor memerlukan perhitungan jarak antara setiap titik data dalam ruang fitur, yang bisa menjadi sangat mahal secara komputasional, terutama dengan jumlah data yang besar. Jika jumlah tetangga (k) besar, perhitungan jarak dan pemilihan mayoritas membutuhkan sumber daya yang signifikan. K-Nearest Neighbor cenderung kurang efektif dalam dimensi tinggi karena fenomena Curse of Dimensionality. Curse of Dimensionality merujuk pada pertumbuhan eksponensial dalam upaya komputasi yang diperlukan untuk memproses data seiring dengan peningkatan dimensi data [36]. Fenomena ini memiliki implikasi signifikan untuk algoritma seperti algoritma K-Nearest Neighbors, yang bergantung pada perhitungan jarak dalam ruang dimensi tinggi. Kutukan dimensi dapat menyebabkan kompleksitas komputasi yang meningkat, kinerja algoritma yang menurun, dan kesulitan dalam merepresentasikan struktur data yang mendasarinya [37]. Dalam ruang fitur yang tinggi, perbedaan antara titik-titik data bisa menjadi kurang signifikan, dan kinerjanya dapat menurun. Pada data bertekstur tinggi, seperti dataset review dengan representasi teks, K-Nearest Neighbor dapat mengalami kesulitan karena kebanyakan dimensi (kata-kata) yang tidak relevan.

Naive Bayes, khususnya Multinomial Naive Bayes, cenderung memakan waktu eksekusi yang lebih cepat karena didasarkan pada perhitungan probabilitas sederhana dan asumsi independensi fitur. Kemampuannya untuk menangani fitur dengan dimensi tinggi dan kemampuan perhitungan probabilitas membuatnya efisien secara komputasional.

Algoritma pohon keputusan seperti ID3 dan C4.5 cenderung efisien dan memakan sumber daya yang moderat. Performa dan penggunaan sumber daya tergantung pada kompleksitas struktur keputusan dan jumlah fitur. Hal ini dikarenakan proses pembentukan pohon berfokus pada fitur yang paling informatif. Pohon keputusan dapat menangani baik data teks dan numerik. Mereka dapat bekerja dengan baik pada kompleksitas data sentimen dengan memahami pola hubungan antara kata-kata.

Algoritma Support Vector Machine dapat memakan waktu eksekusi yang signifikan, terutama pada data yang memiliki jumlah fitur yang besar. Pada dataset teks, representasi vektor kata-kata dapat menjadi rumit dan memakan banyak waktu eksekusi. Namun Support Vector Machine dapat mengatasi batasan linearitas dan bekerja dengan baik pada data yang kompleks, terutama dalam klasifikasi biner pada dataset teks.

4. KESIMPULAN

Berdasarkan percobaan yang dilakukan pada 10.652 buah data dapat diambil kesimpulan bahwa algoritma yang paling baik digunakan dalam konteks analisis sentimen ialah Naive Bayes dan Support Vector Machine dengan tingkat akurasi mendekati 100%. Algoritma Naive Bayes menggunakan memori fisik sebesar 11.193.978 bytes per detik dan memori virtual sebesar 10.627.488 bytes per detik. Karena itu kelebihan dari algoritma ini ialah waktu eksekusi yang cenderung cepat, namun dengan penggunaan sumber daya yang besar pada setiap waktu eksekusinya. Sementara itu algoritma Support Vector Machine menggunakan memori fisik sebesar 477.677 bytes per detik dan memori virtual sebesar 521.618 bytes per detik. Karena itu kelebihan dari algoritma ini ialah penggunaan sumber daya yang kecil pada setiap waktu eksekusinya, namun dengan waktu eksekusi yang cenderung lambat. Sementara itu dari segi penggunaan sumber daya, terlihat bahwa algoritma ID3 menggunakan paling sedikit memori fisik (rata-rata Resident Set Size) dan memori virtual (rata-rata Virtual Memory Size) dibandingkan algoritma lainnya. Namun ia hanya memiliki tingkat akurasi sebesar 92%. Sementara itu dalam penelitian ini, algoritma K-Nearest Neighbor merupakan algoritma yang memakan sumber daya terbesar dengan akurasi terburuk. Karena hal tersebut dapat disimpulkan bahwa dalam konteks analisis sentimen, algoritma K-Nearest Neighbor muncul sebagai pilihan terburuk. Karena itu secara keseluruhan, algoritma ID3, C4.5, Support Vector Machine dan Naive Bayes cenderung lebih cocok untuk analisis sentimen pada data teks karena penggunaan sumber daya yang moderat dan kemampuan menangani data kategorikal. Sementara itu, algoritma K-Nearest Neighbor mungkin kurang cocok untuk dataset teks dengan dimensi tinggi karena Curse of Dimensionality.

REFERENCES

[1] M. Du, X. Li, dan L. Luo, “A Training-Optimization-Based Method for Constructing Domain-Specific Sentiment Lexicon,” Complexity, vol. 2021, hlm. 1–11, Feb 2021, doi: 10.1155/2021/6152494.

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

[2] K. Afifah, I. N. Yulita, dan I. Sarathan, "Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier," dalam 2021 International Conference on Artificial Intelligence and Big Data Analytics, IEEE, Okt 2021, hlm. 22–27. doi: 10.1109/ICAIBDA53487.2021.9689735.
[3] A. Htait, S. Fournier, P. Bellot, L. Azzopardi, dan G. Pasi, "Using Sentiment Analysis for Pseudo-Relevance Feedback in Social Book Search," dalam Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval, New York, NY, USA: ACM, Sep 2020, hlm. 29–32. doi: 10.1145/3409256.3409847.
[4] T. D. Pham dkk., "Natural language processing for analysis of student online sentiment in a postgraduate program," Pacific Journal of Technology Enhanced Learning, vol. 2, no. 2, hlm. 15–30, Sep 2020, doi: 10.24135/pjtel.v2i2.4.
[5] A. Htait, S. Fournier, dan P. Bellot, "Sentiment Analysis and Sentence Classification in Long Book-Search Queries," 2023, hlm. 248–259. doi: 10.1007/978-3-031-24340-0_19.
[6] H. Shirai, N. Inoue, J. Suzuki, dan K. Inui, "Annotating with Pros and Cons of Technologies in Computer Science Papers," dalam Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, hlm. 37–42. doi: 10.18653/v1/W19-2605.
[7] E. Tromp, M. Pechenizkiy, dan M. M. Gaber, "Expressive modeling for trusted big data analytics: techniques and applications in sentiment analysis," Big Data Anal, vol. 2, no. 1, hlm. 1–28, Des 2017, doi: 10.1186/s41044-016-0018-9.
[8] K. Machova, M. Mach, dan M. Vasilko, "Comparison of Machine Learning and Sentiment Analysis in Detection of Suspicious Online Reviewers on Different Type of Data," Sensors, vol. 22, no. 1, hlm. 1–18, Des 2021, doi: 10.3390/s22010155.
[9] V. Umarani, A. Julian, dan J. Deepa, "Sentiment Analysis using various Machine Learning and Deep Learning Techniques," Journal of the Nigerian Society of Physical Sciences, hlm. 385–394, Nov 2021, doi: 10.46481/jnsps.2021.308.
[10] M. R. A. Nasution dan M. Hayaty, "Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter," Jurnal Informatika, vol. 6, no. 2, hlm. 212–218, 2019, doi: 10.31294/ji.v6i2.5129.
[11] F. S. Pattiiha dan H. Hendry, "Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia," JURIKOM (Jurnal Riset Komputer), vol. 9, no. 2, hlm. 506–514, Apr 2022, doi: 10.30865/jurikom.v9i2.4016.
[12] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, dan W. Gata, "ANALISIS SENTIMEN APLIKASI RUANG GURU DI TWITTER MENGGUNAKAN ALGORITMA KLASIFIKASI," Jurnal Teknoinfo, vol. 14, no. 2, hlm. 116–124, Jul 2020, doi: 10.33365/jti.v14i2.679.
[13] U. Y. Ozkan dan T. Demirel, "The influence of window size on remote sensing-based prediction of forest structural variables," Ecol Process, vol. 10, no. 1, hlm. 1–11, Sep 2021, doi: 10.1186/s13717-021-00330-4.
[14] V. Christou dkk., "Evaluating the Window Size's Role in Automatic EEG Epilepsy Detection," Sensors, vol. 22, no. 23, hlm. 1–13, Nov 2022, doi: 10.3390/s22239233.
[15] K. M. Awan, M. Waqar, M. Faseeh, F. Ullah, dan M. Q. Saleem, "Resource management and security issues in mobile phone operating systems: A comparative analysis," PeerJ Prepr, vol. 5, hlm. 1–18, Okt 2017, doi: 10.7287/peerj.preprints.3344v1.
[16] W. Jeon, J. Lee, D. Kang, H. Kal, dan W. W. Ro, "PIMCaffe: Functional Evaluation of a Machine Learning Framework for In-Memory Neural Processing Unit," IEEE Access, vol. 9, hlm. 96629–96640, 2021, doi: 10.1109/ACCESS.2021.3094043.
[17] S. Wang dkk., "Optimization Schemes for In-Memory Linear Regression Circuit With Memristor Arrays," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 12, hlm. 4900–4909, Des 2021, doi: 10.1109/TCSI.2021.3122327.
[18] T. Nguyen dan A. J. McCaskey, "Extending Python for Quantum-classical Computing via Quantum Just-in-time Compilation," ACM Transactions on Quantum Computing, vol. 3, no. 4, hlm. 1–25, Des 2022, doi: 10.1145/3544496.
[19] W. Zhang, X. Chen, Y. Liu, dan Q. Xi, "A Distributed Storage and Computation k-Nearest Neighbor Algorithm Based Cloud-Edge Computing for Cyber-Physical-Social Systems," IEEE Access, vol. 8, hlm. 50118–50130, 2020, doi: 10.1109/ACCESS.2020.2974764.
[20] T. D. Novianto dan I. M. S. Erawan, "Perbandingan Metode Klasifikasi pada Pengolahan Citra Mata Ikan Tuna," Prosiding SNFA (Seminar Nasional Fisika dan Aplikasinya), vol. 5, hlm. 216–223, Des 2020, doi: 10.20961/prosidingsnfa.v5i0.46615.
[21] F. S. Al-Anzi dan D. AbuZeina, "Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 2, hlm. 189–195, Apr 2017, doi: 10.1016/j.jksuci.2016.04.001.
[22] R. Ferdiana, F. Jatmiko, D. D. Purwanti, A. Sekar, T. Ayu, dan W. F. Dicka, "Dataset Indonesia untuk Analisis Sentimen," JNTETI, vol. 8, no. 4, hlm. 334–339, 2019.
[23] V. H. Wong, W. L. Tan, J. L. Kor, dan X. V. Wan, "Autonomous Language Processing and Text Mining by Data Analytics for Business Solutions," dalam Proceedings of the International Conference on Mathematical Sciences and Statistics 2022 (ICMSS 2022), Dordrecht: Atlantis Press International BV, 2023, hlm. 85–93. doi: 10.2991/978-94-6463-014-5_9.
[24] M. B. Hamzah, "Classification of Movie Review Sentiment Analysis Using Chi-Square and Multinomial Naïve Bayes with Adaptive Boosting," Journal of Advances in Information Systems and Technology, vol. 3, no. 1, hlm. 67–74, Apr 2021, doi: 10.15294/jaist.v3i1.49098.
[25] A. Sabrani, I. G. W. Wedashwara W., dan F. Bimantoro, "Multinomial Naïve Bayes untuk Klasifikasi Artikel Online tentang Gempa di Indonesia," Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA), vol. 2, no. 1, hlm. 89–100, Mar 2020, doi: 10.29303/jtika.v2i1.87.
[26] R. Hou, L. Wang, dan Y.-J. Wu, "Predicting ATP-Binding Cassette Transporters Using the Random Forest Method," Front Genet, vol. 11, hlm. 1–11, Mar 2020, doi: 10.3389/fgene.2020.00156.
[27] Q. Gao, "Design and Implementation of 3D Animation Data Processing Development Platform Based on Artificial Intelligence," Comput Intell Neurosci, vol. 2022, hlm. 1–7, Mei 2022, doi: 10.1155/2022/1518331.

Submitted: **22/12/2023**; Accepted: **21/01/2024**; Published: **24/01/2024**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 8, Nomor 1, Januari 2024, Page 371-382
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v8i1.7201

[28] W. Han, "Quantitative Modelling of Climate Change Impact on Hydro-climatic Extremes," Swansea University, Swansea, 2021. doi: 10.23889/SUthesis.58888.
[29] R. H. A Alsagheer, A. F. H Alharan, dan A. S. A Al-Haboobi, "Popular Decision Tree Algorithms of Data Mining Techniques: A Review," International Journal of Computer Science and Mobile Computing, vol. 6, no. 6, hlm. 133–142, 2017.
[30] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, dan F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 21, no. 3, hlm. 677–690, Jul 2022, doi: 10.30812/matrik.v21i3.1726.
[31] H. Jeiad, Z. Ameen, dan A. Mahmood, "Employee Performance Assessment Using Modified Decision Tree," Engineering and Technology Journal, vol. 36, no. 7A, hlm. 806–811, Jul 2018, doi: 10.30684/etj.36.7A.14.
[32] R. Alshammari, "Arabic Text Categorization using Machine Learning Approaches," International Journal of Advanced Computer Science and Applications, vol. 9, no. 3, hlm. 227–230, 2018, doi: 10.14569/IJACSA.2018.090332.
[33] A. Suherman, D. KURNAEDI, S. Lusa, dan R. Darmawan, "Junior Class Preparedness Classification Faces A National Exam Using C.45 Algorithm with A Particle Swarm Optimization Approach," bit-Tech, vol. 2, no. 3, hlm. 101–109, Nov 2020, doi: 10.32877/bt.v2i3.133.
[34] R. Pambudi dan F. Madani, "Analysis of public opinion sentiment against COVID-19 in Indonesia on twitter using the k-nearest neighbor algorithm and decision tree," Journal of Soft Computing Exploration, vol. 3, no. 2, hlm. 117–122, Sep 2022, doi: 10.52465/joscex.v3i2.88.
[35] M. F. A. Saputra, T. Widiyaningtyas, dan A. P. Wibawa, "Illiteracy Classification Using K Means-NaÃ¯ve Bayes Algorithm," JOIV : International Journal on Informatics Visualization, vol. 2, no. 3, hlm. 153–158, Mei 2018, doi: 10.30630/joiv.2.3.129.
[36] M. Hutzenthaler, A. Jentzen, dan von W. Wurstemberger, "Overcoming the curse of dimensionality in the approximative pricing of financial derivatives with default risks," Electron J Probab, vol. 25, no. none, hlm. 1–73, Jan 2020, doi: 10.1214/20-EJP423.
[37] S. Parhizkari, "Anomaly Detection in Intrusion Detection Systems," 2023. doi: 10.5772/intechopen.112733.