Received: 30 Mar 2025 | Revised: 24 Jun 2025 | Accepted: 18 May 2025 | Available online: 05 Dec 2025
https://doi.org/10.13057/ijas.v8i1.100904

E-ISSN: 2621-086X
https://jurnal.uns.ac.id/ijas

Indonesian Journal of Applied Statistics Vol. 8, No.1, pp. 24-37, 2025

Penerapan Teknik *Soft Voting Ensemble* pada Klasifikasi *Rating* Film

Alina Selia Rizka*, Virgania Sari
Program Studi Matematika, Universitas Negeri Semarang, Semarang, Indonesia
* Corresponding Author. E-mail: alinaaselrizka@students.unnes.ac.id

Abstrak

Pertumbuhan jumlah penonton film yang begitu pesat mendorong industri perfilman untuk terus berinovasi, sehingga menghasilkan beragam judul baru dengan genre dan karakteristik yang semakin bervariasi. Kondisi ini menyebabkan kompleksitas data yang tinggi, sehingga dibutuhkan metode klasifikasi yang efektif dan akurat untuk mengelompokkan *rating* film berdasarkan karakteristiknya. Penelitian ini bertujuan untuk meningkatkan kinerja klasifikasi *rating* film dengan menggunakan metode Ensemble Soft Voting, yang menggabungkan tiga algoritma klasifikasi, yaitu k-*nearest neighbor* (KNN), *decision tree* (DT), dan *support vector machine* (SVM). Evaluasi dilakukan dengan membandingkan kinerja metode *soft voting* terhadap masing-masing metode individu berdasarkan metrik akurasi, presisi, sensitivitas, dan F1-*score*. Hasil penelitian menunjukkan bahwa metode *soft voting* memberikan kinerja klasifikasi yang lebih baik dibandingkan metode KNN, *decision tree*, dan SVM secara terpisah, dengan capaian akurasi sebesar 89,64%, presisi 85,63%, sensitivitas 89,64%, dan nilai F1-*score* sebesar 86,52%.

The rapid growth in the number of movie viewers has driven the film industry to continuously innovate, resulting in a diverse range of new titles with increasingly varied genres and characteristics. This has led to significant data complexity, necessitating an effective and accurate classification method to categorize movie ratings based on their characteristics. This study aims to evaluate the performance of the Soft Voting ensemble method in classifying movie ratings. The classification results from soft voting are compared to those of individual models, namely k-nearest neighbor (KNN), decision tree (DT), and support vector machine (SVM). The evaluation was conducted by comparing the performance of the soft voting method against each individual method based on accuracy, precision, sensitivity, and F1-score metrics. The results showed that the soft voting method provided better classification performance than the KNN, decision tree, and SVM methods individually, with an accuracy of 89.64%, a precision of 85.63%, a sensitivity of 89.64%, and an F1-score of 86.52%.

Kata kunci: klasifikasi, ensemble learning, KNN, decision tree, SVM

Keywords: *classification, ensemble learning, KNN, decision tree, SVM*

How to Cite:
A. S. Rizka, and V. Sari, "Penerapan teknik soft voting ensemble pada klasifikasi rating film," *Indonesian Journal of Applied Statistics,* vol. 8, no. 1, pp. 24-37, 2025, doi: 10.13057/ijas.v8i1.100904.

1. PENDAHULUAN

Perkembangan industri film mengalami pertumbuhan yang sangat cepat dan bervariasi. Film dalam berbagai genre dan bentuknya, telah menjadi bagian penting dari budaya populer di seluruh dunia dan menjadi salah satu media hiburan yang sangat disukai. Peningkatan jumlah penonton setiap tahunnya mendorong industri perfilman untuk terus berinovasi, sehingga menghasilkan banyak judul baru dengan genre dan karakteristik yang semakin beragam.

Platform digital seperti Netflix, Disney+, dan berbagai platform digital lainnya menghadapi tantangan dalam menyajikan rekomendasi film yang relevan bagi pengguna. Salah satu pendekatan

penting dalam sistem rekomendasi adalah dengan mengklasifikasikan film berdasarkan *rating*. Klasifikasi *rating* film menjadi aspek penting dalam sistem rekomendasi karena dapat membantu menyaring dan menyusun daftar tontonan yang sesuai preferensi pengguna, memfilter konten berdasarkan kualitas, serta mempercepat proses pencarian film yang layak ditonton. Peran klasifikasi *rating* tidak hanya terbatas pada rekomendasi, tetapi juga mencakup pengawasan konten dan strategi produksi film yang disesuaikan dengan tren preferensi pasar. Semakin banyaknya judul film yang tersedia, pengguna kerap dihadapkan pada pilihan berlebih (*information overload*), sehingga sistem rekomendasi yang berbasis pada klasifikasi *rating* menjadi sangat dibutuhkan. Kompleksitas data yang sangat besar dan beragam ini menimbulkan tantangan tersendiri dalam menghasilkan klasifikasi yang efektif dan akurat. Mengatasi tantangan tersebut, dibutuhkan metode klasifikasi yang mampu mengelola data berukuran besar dan juga memahami pola preferensi yang bervariasi.

Klasifikasi merupakan salah satu teknik penting dalam *machine learning* yang bertujuan untuk mengelompokkan data ke dalam kategori tertentu berdasarkan pola yang dipelajari dari data sebelumnya. *Machine learning* telah menjadi salah satu cabang kecerdasan buatan yang berkembang pesat dan banyak diterapkan dalam berbagai bidang, termasuk pengolahan data film. Penelitian terdahulu telah melakukan studi membangun sistem rekomendasi film menggunakan *support vector machine* (SVM) dan *content-based filtering* [1]. Penelitian lain yang membahas tentang mengembangkan sistem rekomendasi produk berbasis film dengan pendekatan *hybrid* yang menggabungkan IMDb *weighted rating* dan *term frequency – inverse document frequency* (TF-IDF) untuk menilai kesesuaian antara pengguna dan produk juga telah dilakukan [2]. Pendekatan ini menekankan pentingnya kombinasi informasi konten dengan nilai *rating* sebagai dasar rekomendasi yang lebih akurat. Penelitian lain juga telah melakukan perbandingan performa antara algoritma k-*nearest neighbor* (KNN) dan Naïve Bayes dalam memprediksi *rating* film Indonesia dan menunjukkan bahwa penggunaan teknik *feature selection* mampu meningkatkan akurasi model secara signifikan [3]. Ketiga penelitian tersebut memberikan kontribusi dalam pemrosesan dan analisis data film, namun belum mengeksplorasi penggunaan teknik *ensemble* seperti *soft voting* dalam klasifikasi *rating* film. Ketiga penelitian ini juga menunjukkan bahwa pendekatan klasifikasi berbasis *machine learning* telah mulai banyak diterapkan dalam domain hiburan, khususnya film, namun penggunaan metode *ensemble* seperti *soft voting* masih jarang ditemukan. Hal ini menunjukkan adanya peluang untuk mengembangkan pendekatan klasifikasi yang lebih kompleks dan akurat dibidang ini, sebagaimana dilakukan dalam penelitian ini.

Metode-metode *machine learning* memungkinkan sistem untuk mengenali pola kompleks dalam data yang mungkin sulit dideteksi oleh manusia. Terdapat beberapa metode klasifikasi, seperti KNN, SVM, *decision tree*, dan lainnya. Metode KNN umumnya dikenal sebagai suatu pendekatan yang berfokus pada kesamaan, yakni dengan mencari kelompok objek di dalam dataset pelatihan yang memiliki kedekatan terbesar dengan objek pada data baru atau data uji [4]. Metode lain yang umum digunakan untuk menangani masalah klasifikasi adalah *decision tree* [5]. *Decision tree* bekerja dengan memecahkan solusi dari masalah melalui penggunaan kriteria sebagai simpul yang saling terikat, membentuk struktur mirip pohon. Metode lain seperti *support vector machine* juga merupakan metode klasifikasi yang mendasarkan prinsip kerjanya pada klasifikasi linier, di mana dapat memisahkan dua kelas secara linier. Meskipun demikian, SVM telah dikembangkan guna menyelesaikan permasalahan non-linier dengan menyatukan aturan kernel ke dalam ruang kerja yang memiliki dimensi tinggi. Selain metode individual, terdapat juga pendekatan *ensemble learning* yang menggabungkan beberapa algoritma untuk meningkatkan akurasi dan stabilitas prediksi. Teknik *ensemble learning* yang cukup populer adalah *voting classifier*, khususnya *soft voting*, yang mengombinasikan probabilitas hasil prediksi dari beberapa model. *Soft voting* memungkinkan sistem untuk lebih adaptif dalam menghasilkan prediksi dan mampu memproses data besar secara efisien. metode ini bekerja dengan menggabungkan beberapa model untuk mencapai solusi prediksi yang lebih baik dan lebih akurat dibandingkan dengan satu model tunggal [6].

Penelitian sebelumnya yang menggunakan teknik *ensemble learning* model *soft voting* dalam memprediksi diabetes lebih unggul dibandingkan dengan metode individu dengan akurasi mencapai

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

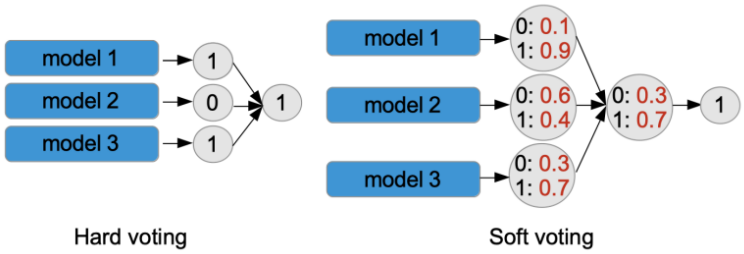
97.74% [7]. Penelitian lain juga menyimpulkan bahwa metode *soft voting classifier* (random forest, KNN, naive bayes) sangat baik degan akurasi sebesar 99,03% dibandingkan dengan metode tunggal [8]. Penelitian lain juga telah dilakukan mengenai prediksi gempa bumi dengan teknik *ensemble soft voting* memperoleh hasil dua algoritma terbaik untuk model *soft voting*, yaitu HGBoost dan AdaBoost dengan akurasi mencapai 94% [9]. Model *soft voting* terbukti memiliki performa terbaik dibandingkan model lainnya. Masih terbuka ruang penelitian untuk mengeksplorasi penerapan metode ini dalam konteks klasifikasi film berdasarkan *rating* pengguna, yang belum banyak dibahas sebelumnya dalam penelitian serupa. Penelitian-penelitian sebelumnya telah membuktikan keunggulan metode *soft voting*, namun sebagian besar belum menerapkannya pada domain film.

Penelitian ini memberikan kontribusi dengan menerapkan metode *soft voting* dalam domain klasifikasi film, yang masih jarang dibahas dalam penelitian sebelumnya. Selain itu, kombinasi tiga algoritma klasifikasi (KNN, *decision tree*, dan SVM) dengan teknik ekstraksi fitur TF-IDF serta penyeimbangan data menggunakan *synthetic minority over-sampling technique* (SMOTE), menjadikan pendekatan ini lebih kuat dalam menangani data tekstual yang tidak seimbang. Kontribusi ini menyoroti potensi penerapan teknik *ensemble* pada domain hiburan yang kompleks dan terus berkembang. dan dapat membantu pengembangan sistem rekomendasi film berbasis kualitas *rating* secara lebih efisien.

2. METODE

2.1. *Voting Classifier*

Voting classifier merupakan metode *ensemble learning* yang menggabungkan beberapa model untuk mencapai solusi prediksi yang lebih baik dan lebih akurat dibandingkan dengan satu model tunggal [6]. Dengan mengintegrasikan beberapa algoritma, metode ini bertujuan menghasilkan hasil prediksi yang lebih andal, stabil, dan optimal. Penggunaan metode *ensemble* memang biasanya membutuhkan lebih banyak komputasi dibandingkan model tunggal, tetapi stabilitas dan akurasi yang lebih tinggi membuatnya bernilai. Teknik *voting classifier* memiliki dua jenis metode, yaitu *hard voting classifier* dan *soft voting classifier*. *Soft voting classifier* adalah salah satu teknik dalam metode *ensemble* yang umum digunakan. Dalam pendekatan ini, setiap model dalam *ensemble* memberikan prediksi dalam bentuk probabilitas atau skor keanggotaan untuk masing-masing kelas, bukan hanya sekadar menetapkan satu label kelas tertentu [8]. Teknik ini bertujuan untuk mengurangi bias dan varians, sehingga menciptakan prediksi yang lebih akurat dan andal. Gambar 1 memperlihatkan perbedaan proses antara *hard voting* dan *soft voting*.



Gambar 1. Perbedaan proses *hard* dan *soft voting* [19]

Berdasarkan proses *soft voting* yang diilustrasikan dalam penelitian [19], dapat dirumuskan suatu pendekatan matematis yang digunakan untuk menghitung hasil akhir klasifikasi berdasarkan rata-rata probabilitas dari masing-masing model. Proses ini tidak hanya mempertimbangkan label kelas yang diprediksi oleh tiap model, tetapi juga mempertimbangkan tingkat kepercayaan dari masing-masing prediksi. Keputusan akhir dibuat berdasarkan agregasi nilai probabilitas yang diberikan oleh seluruh model yang digunakan. Rumus perhitungan *soft voting* adalah sebagai berikut:

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

$$\hat{y} = \arg max_{c \epsilon C} \left(\sum_{i=1}^{n} w_i \, . P_i(c) \right)$$

dengan $\hat{y}$ adalah kelas hasil prediksi akhir, C adalah himpunan semua kelas, w_i adalah bobot model ke-i, dan $P_i(c)$ adalah probabilitas prediksi model ke-i untuk kelas ke-c.

2.2. K-*Nearest Neighbor*

K-*nearest neighbor* merupakan salah satu metode klasifikasi yang melibatkan pengukuran jarak antara data baru dengan data pelatihan yang ada. KNN bekerja dengan mencari sekumpulan k objek terdekat dari kumpulan data pelatihan untuk memprediksi kelas atau nilai dari data yang sedang diuji [10]. Pada penerapan k-*nearest neighbor*, dataset dibagi menjadi dua, yaitu data latih dan data uji. Data latih berfungsi untuk membentuk dasar prediksi, sementara data uji berisi nilai-nilai yang akan diprediksi [11]. Proses mengelompokkan data baru dilakukan dengan mengukur jarak antara data baru tersebut dengan sejumlah data tetangga terdekat. Pada k-*nearest neighbor*, Euclidean *distance function* dipakai untuk mengukur jarak antara titik-titik data. Rumus Euclidean *distance* adalah sebagai berikut:

$$euc = \sqrt{\left(\sum_{i=1}^{n} (x_i - y_i)^2\right)}$$

dengan x_i adalah data latih, y_i adalah data uji, i adalah variabel data, dan n adalah dimensi data.

2.3. *Decision Tree*

Decision tree merupakan salah satu teknik klasifikasi dalam data mining yang menggunakan dataset berlabel untuk membangun dan menghasilkan pohon keputusan sebagai *output* [12]. Metode *decision tree* terdiri dari berbagai simpul yang terhubung melalui cabang-cabang. Cabang-cabang tersebut dimulai dari *root node* dan berakhir pada *leaf node*. *Leaf node*, yang sudah tidak dapat dibagi lagi, menampilkan jawaban yang diharapkan untuk masalah tertentu (data uji). *Decision Tree* atau pohon keputusan adalah suatu metode yang mengubah kumpulan data yang besar menjadi sebuah struktur pohon yang mempresentasikan serangkaian aturan untuk pengambilan keputusan. Proses pembangunan model klasifikasi *decision tree* melibatkan langkah-langkah berikut [13]:

1. Menentukan nilai *entropy*

Misal $S \in R$ menyatakan variabel acak diskrit yang berisi nilai $s_1, s_2, \dots, s_k$ dengan probabilitas $p_1, p_2, \dots, p_k$ masing-masing *entropy* $H(S)$ dari S, didefinisikan dengan:

$$Entropy(S) = -\sum_{i=1}^{k} \frac{n_i}{|S|} \times \log_2 \left(\frac{n_i}{|S|} \right)$$

dengan S adalah variabel tujuan, n_i adalah proporsi sampel pada partisi ke-i dalam S, k adalah banyaknya partisi S.

2. Setelah memperoleh nilai *entropy*, diberi nilai *information gain*

$$Gain(S, A) = H(S) - \sum_{v \in Values(A)} \frac{S_{A(v)}}{S} \times H(S_{A(v)})$$

dengan

A	: atribut (variabel yang diuji)
S	: variabel target
$H(S)$	: *entropy* untuk dataset sebelum terjadi perubahan apa pun
$S_{A(v)}$	: himpunan bagian dari S dengan nilai v untuk atribut A
$H(S_{A(v)})$	: *entropy* kelompok sampel dengan nilai v untuk atribut A
v	: setiap nilai yang mungkin untuk atribut A

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

3. Mengidentifikasi atribut dengan nilai *information gain* tertinggi dan menetapkannya sebagai simpul utama yang berisi atribut dengan nilai *information gain* tertinggi $G(x_i)$
4. Proses perhitungan *information gain* berlanjut hingga seluruh data terbagi ke dalam kelompok yang seragam. Atribut yang sudah dipilih tidak akan digunakan lagi dalam perhitungan nilai *information gain* berikutnya.

2.4. *Support Vector Machine*

Support vector machine adalah metode pembelajaran yang mengklasifikasikan data dengan memanfaatkan ruang hipotesis berupa fungsi linear dalam ruang fitur dimensi tinggi. SVM membangun *hyperplane* dalam ruang berdimensi banyak untuk memisahkan dua kelas yang berbeda. SVM sudah mengalami pengembangan guna mengatasi permasalahan *non-linear* dengan memperkenalkan bentuk penggunaan *kernel*, yang memungkinkan pemetaan data ke ruang fitur berdimensi lebih tinggi. Dalam ruang berdimensi tinggi ini, SVM mencari *hyperplane* yang dapat mengoptimalkan jarak (*margin*) antar berbagai kelas data [14]. Rumus umum pada SVM linear adalah sebagai berikut:

$$f(x) = sign(w.x + b)$$

dengan
$f(x)$: fungsi prediksi
w : vektor normal *hyperplane*
x : vektor fitur *input*
b : bias atau *intercept*
Terdapat 4 fungsi *kernel* yang sering digunakan yaitu:

1. *Kernel* linear

Kernel linear menghitung hasil dari perkalian titik (*dot product*) antara dua vektor *input* langsung dalam ruang asli tanpa memerlukan transformasi eksplisit ke ruang fitur berdimensi lebih tinggi.

$$K(x, y) = (x.y)$$

2. *Kernel* polinomial

Kernel polinomial menghitung hubungan *polynomial* antara dua vektor *input* pada ruang asli.

$$K(x, y) = (x.y + c)^d$$

3. *Kernel radial basis function* (RBF)

Kernel RBF memakai fungsi *Gaussian*, yang dikenal sebagai fungsi basis radial, guna mengukur kesamaan antara dua vektor *input* dalam ruang fitur.

$$K(x, x') = \exp(-\gamma(x, y)^2)$$

4. *Kernel sigmoid function*

Fungsi tangen hiperbolik sering kali dikenal sebagai fungsi *sigmoid*.

$$K(x, y) = \tan(ax^T y + c) \quad , a, c > 0$$

2.5. SMOTE

SMOTE (*synthetic minority over-sampling technique*) merupakan metode dalam *machine learning* yang bertujuan untuk mengatasi ketidakseimbangan kelas dalam dataset dengan menghasilkan data sintetis pada kelas minoritas [15]. Penerapan SMOTE dapat meningkatkan efektivitas dalam klasifikasi. Teknik ini tidak hanya menambah jumlah sampel pada kelas minoritas, tetapi juga berperan dalam meningkatkan akurasi model.

2.6. *Term Frequency-Inverse Document* (TF-IDF)

TF-IDF adalah teknik pembobotan statistik numerik yang berguna untuk menentukan seberapa penting sebuah kata pada suatu dokumen. TF-IDF merupakan gabungan dari dua ukuran utama yang

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

berbeda yaitu, *term frequency* (TF) dan *inverse document frequency* (IDF). TF berfungsi untuk mengukur frekuensi kemunculan sebuah kata dalam sebuah dokumen. Rumus TF dapat dilihat di bawah ini:

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}}$$

dengan
TF_{ij} : TF kata i dalam dokumen j
f_{ij} : frekuensi kemunculan i dalam dokumen j
$max_k f_{kj}$: jumlah kata dalam dokumen j

Inverse document frequency (IDF) adalah teknik pengukuran yang berfungsi untuk mengukur pentingnya sebuah kata dalam suatu dokumen [16]. Nilai IDF akan meningkat atau tinggi jika kata tersebut jarang ditemukan di dokumen, menandakan bahwa kata tersebut lebih signifikan dan membawa informasi unik dan sebaliknya. Rumus IDF dapat dilihat di bawah ini:

$$IDF_i = log \frac{N}{n_i}$$

dengan
IDF_i : frekuensi kata i dalam *corpus*
N : jumlah total keseluruhan dokumen dalam *corpus*
n_i : jumlah dokumen yang mengandung fitur (kata) i

2.7. K-*Fold Cross Validation*

Cross-validation atau estimasi rotasi adalah salah satu teknik statistik yang digunakan untuk menguji efektivitas algoritma machine learning. Terdapat berbagai metode *cross-validation*, tetapi k-fold *cross-validation* dipilih karena populer dan mudah dipahami [17]. Model dilatih dengan k-1 subset sebagai training data dan diuji menggunakan 1 subset yang tersisa sebagai testing data. Langkah-langkah k-*fold cross-validation* yaitu [18]:

1. Membagi kumpulan data menjadi bagian yang sama atau biasa disebut lipatan (*fold*). Total data dipecah menjadi k bagian.
2. *Fold* ke-1 adalah saat bagian ke-1 sebagai data uji dan sisanya sebagai data latih.
3. *Fold* ke-2 adalah saat bagian ke-2 sebagai data uji dan sisanya sebagai data latih.
4. Seterusnya sampai memperoleh *fold* ke-k. Selanjutnya, dihitung rata-rata akurasi dari k buah akurasi di atas. Rata-rata akurasi ini merupakan akurasi final.

2.8. *Confusion Matrix*

Confusion matrix merupakan tabel hasil kinerja klasifikasi. *Confusion matrix* digunakan untuk menentukan kinerja klasifikasi pada machine learning dengan ditunjukkan dalam bentuk matriks yang menghasilkan perbandingan antara nilai aktual dan prediksi. Contoh *confusion matrix* yang terdiri dari dua kelas, yaitu kelas positif dan kelas negatif, yang tunjukkan pada Tabel 1. Matriks konfusi memberikan gambaran jelas tentang kinerja model klasifikasi serta jenis kesalahan yang dihasilkannya. Berdasarkan Tabel 1 dapat diketahui ringkasan prediksi yang benar yaitu (TP dan TN) dan yang salah (FP dan FN) yang dirinci oleh setiap kategori.

Tabel 1. *Confusion matrix*

Aktual	Prediksi	
	Positif	Negatif
Positif	*true positive* (TP)	*false negative* (FN)
Negatif	*false positive* (FP)	*true negative* (TN)

Dari penjelasan *confusion matrix* di atas dapat diidentifikasi berbagai metrik evaluasi kinerja, sebagai berikut.

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

1. Akurasi

Menghitung presentase hasil klasifikasi (memprediksi peristiwa positif dan negatif yang benar) sebagai presentase dari jumlah peristiwa yang terjadi.

$$\text{Akurasi} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

2. Presisi

Menghitung presentase kejadian positif yang diprediksi dengan benar terhadap total kejadian positif yang diprediksi. Presisi dapat dihitung dengan persamaan berikut:

$$\text{Presisi} = \frac{TP}{(TP + FP)}$$

3. Sensitivitas

Recall dapat dijelaskan sebagai presentase hasil positif yang diprediksi dengan benar dari semua hasil positif yang sebenarnya. *Recall* juga disebut sensitivitas. *Recall* dapat dihitung dengan persamaan berikut:

$$\text{Sensitivitas} = \frac{TP}{(TP + FN)}$$

4. F1-score

F1-*score* adalah perbandingan bobot rata-rata presisi dan sensitivitas. F1-*score* dapat dihitung dengan persamaan berikut:

$$F1\ score = \frac{2\ x\ precision\ x\ recall}{precision + recall}$$

2.9. Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari Kaggle, sebuah platform yang menyediakan berbagai dataset untuk penelitian ataupun analisis. Dataset yang akan digunakan merupakan data movie. *Dataset* ini bersumber dari *the movie database* (TMDB) dan mencakup informasi film yang dapat dilihat pada movie_dataset. Jumlah total observasi yang digunakan dalam penelitian ini sebanyak 5.801 data film. Dari 20 atribut pada movie_dataset, dipilih 10 atribut yang relevan berdasarkan kajian pustaka dan kontribusi informasi terhadap klasifikasi *rating*. Atribut tersebut dipilih karena mewakili aspek penting dalam produksi film seperti popularitas, durasi, kata kunci, genre, dan lainnya, yang dinilai mampu memengaruhi persepsi *rating* pengguna. Variabel target dalam penelitian ini adalah *vote average* yang menunjukkan rata-rata *rating* film dari pengguna. Rincian 10 atribut yang digunakan dapat dilihat pada Tabel 2.

Tabel 2. Variabel penelitian

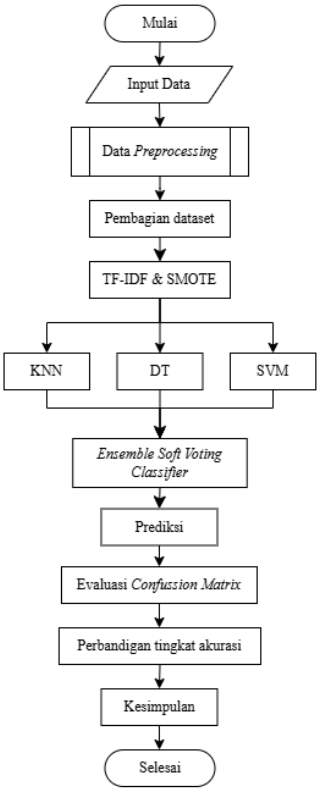
No	Variabel	Tipe data	Keterangan
1	TmdbId	*Integer*	Identifikasi unik dari setiap film
2	*Budget*	*Integer*	Anggaran pembuatan film
3	*Genres*	*Object*	Genre dari masing-masing film
4	*Original Language*	*Object*	Bahasa asli untuk film
5	*Popularity*	*Float*	Skor popularitas film
6	*Release Date*	*Datetime*	Tanggal rilis film
7	*Runtime*	*Integer*	Durasi waktu pada film
8	*Production Countries*	*Object*	Negara asal produksi film
9	*Keywords*	*Object*	Kata kunci dari film
10	*Vote Average*	*Float*	Peringkat rata-rata yang diterima film

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

2.10. Tahapan Analisis Data

Tahapan analisis data pada penelitian ini secara adalah sebagai berikut:

1. Input data sebagai langkah awal untuk memulai proses klasifikasi *rating* film.
2. Tahap *preprocessing* pada data yang akan digunakan untuk pemodelan. Proses ini meliputi penanganan nilai yang hilang (*missing values*), duplikasi data, dan transformasi data. Selain itu, juga dilakukan preprocessing teks pada atribut *keywords*, di mana prosesnya meliputi normalisasi teks, tokenisasi, *stopwords*, dan *lemmatization*. Lalu, dilakukan pelabelan data dengan memberi label sesuai dengan kategori klasifikasi yang telah ditentukan.
3. Pembagian dataset dilakukan dengan membagi dataset menjadi dua bagian utama, yaitu data latih (*data training*) dan data uji (*data test*). Tujuannya agar model dapat dilatih menggunakan data latih dan diuji dengan data uji untuk mengukur performanya.
4. Penerapan SMOTE dan TF-IDF pada data latih untuk menangani ketidakseimbangan kelas dalam dataset. Selain itu, TF-IDF digunakan untuk mengekstrak fitur dari teks yang terdapat dalam dataset.
5. Proses klasifikasi diterapkan pada data latih menggunakan metode KNN, *decision tree*, SVM, dan *soft voting*. Untuk memastikan keandalan model, digunakan teknik k-*fold cross validation* dengan 5 *fold*. Teknik ini membagi data latih menjadi lima bagian, di mana setiap fold digunakan sebagai data uji secara bergantian, sementara *fold* lainnya digunakan untuk pelatihan model. Model yang telah dilatih digunakan untuk memprediki data uji, lalu hasilnya dibandingkan dengan label sebenarnya untuk menilai akurasi model.
6. Model dievaluasi menggunakan akurasi, presisi, sensitivitas, dan F1-*score*. Tujuannya untuk menilai seberapa baik model dalam melakukan klasifikasi *rating* film.
7. Kinerja dari berbagai metode klasifikasi yang digunakan dibandingkan berdasarkan hasil evaluasi. Perbandingan ini bertujuan untuk menentukan metode terbaik untuk klasifikasi *rating* film.



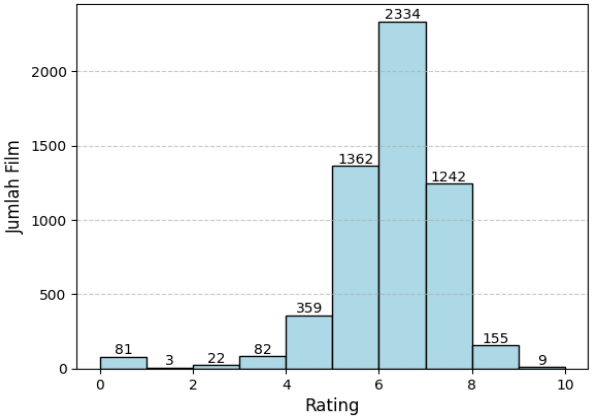
Gambar 2. Diagram alir analisis data

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

3. HASIL DAN PEMBAHASAN

3.1. Analisis Deskriptif

Jumlah data *rating* film yang akan digunakan sebanyak lima ribu delapan ratus satu data dengan sepuluh atribut. Data *rating* film ini kemudian dilakukan analisis distribusi *rating* didapat nilai *rating* film yang berkisar antara nol hingga sepuluh, dengan nilai *rating* dihitung berdasarkan rata-rata dari seluruh pengguna yang sudah memberikan penilaian. Distribusi *rating* akan ditampilkan dalam bentuk histogram yang dapat dilihat pada Gambar 3.

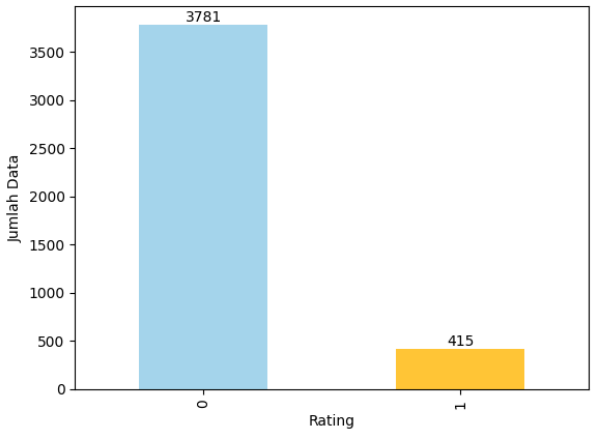


Gambar 3. Grafik distribusi *rating* film

Gambar 3 menunjukkan bahwa mayoritas film mendapatkan *rating* direntang tinggi, yakni antara nilai *rating* enam hingga nilai *rating* delapan yang dapat disimpulkan bahwa kebanyakan film memiliki kualitas baik menurut pengguna. Film yang memiliki *rating* yang rendah, yaitu di bawah lima berjumlah sedikit.

3.2. Pelabelan dan Pembagian Data

Pada kolom *rating* yang merupakan target atau label dalam peneilian ini juga dilakukan proses kategorisasi untuk membagi nilai *rating* menjadi dua kategori yaitu, direkomendasikan untuk nilai *rating* di atas lima dan tidak direkomendasikan untuk nilai *rating* lima atau kurang dari lima. Kategori ini kemudian dienkode ke dalam format numerik dengan nilai "0" untuk "Direkomendasikan" dan "1" untuk "Tidak Direkomendasikan" menggunakan *label encoding*. Jumlah data yang sudah dikategorikan akan ditunjukkan melalui visualisasi histogram yang dapat dilihat pada Gambar 4.



Gambar 4. Grafik distribusi label *rating*

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

Proses selanjutnya adalah pembagian *dataset*, di mana *dataset* dibagi menjadi dua bagian, yaitu data *train* dan data *test* untuk mendukung proses pelatihan dan evaluasi model. Pada penelitian ini pembagian dilakukan dengan proporsi 80% untuk data *train* dan 20% untuk data *test*. Pembagian data ini menggunakan *library sklearn.model_selection import train_test_split*. Hasil pembagian data *train* dan data *test* dapat dilihat pada Tabel 3.

Tabel 3. Pembagian *dataset*

	Data *training*	Data *testing*
Presentase	80%	20%
Jumlah Data	3356	840

3.3. Pembobotan TF-IDF

Proses perhitungan dilakukan untuk semua dokumen yang ada dalam *dataset*. Dengan menggunakan bantuan Python semua bobot pada kata dapat dilakukan dengan lebih cepat. Hasil pembobotan dengan menggunakan Python didapat 10 kata dengan bobot tertinggi yang ditunjukkan pada Tabel 4.

Tabel 4. Sepuluh bobot tertinggi

No	Kata	Bobot
1	*woman*	0.0294
2	*director*	0.0265
3	*independent*	0.0254
4	*based*	0.0248
5	*relationship*	0.0248
6	*love*	0.0218
7	*murder*	0.0183
8	*war*	0.0176
9	*sport*	0.0160
10	*new*	0.0154

Kata-kata dengan bobot tinggi berperan penting dalam membedakan satu dokumen dengan dokumen lain. Melalui penerapan TF-IDF ini, kata-kata yang sering muncul di banyak dokumen akan memiliki bobot lebih rendah, sedangkan kata-kata yang lebih spesifik dalam dokumen tertentu akan memiliki bobot yang lebih tinggi.

3.4. Klasifikasi

Berdasarkan Gambar 4 menunjukkan ketidakseimbangan jumlah data antar kelas. Oleh karena itu, sebelum melakukan pemodelan digunakan SMOTE untuk menyeimbangkan jumlah data di setiap kelas dalam data latih. Teknik bekerja dengan membuat sampel sintetis untuk kelas minoritas, sehingga model dapat belajar secara lebih seimbang dan tidak bias terhadap kelas mayoritas. Sebelum SMOTE, data menunjukkan ketidakseimbangan kelas yang cukup signifikan, dengan proporsi kelas "Direkomendasikan" sebesar 87% dan "Tidak Direkomendasikan" hanya 13%. Setelah diterapkannya SMOTE, jumlah data pada kelas minoritas meningkat menjadi seimbang dengan kelas mayoritas. Selain itu, agar model dapat dievaluasi dengan lebih baik, digunakan k-*fold cross validation* dengan 5 *fold* pada setiap model. Dengan menggunakan ini, performa setiap model dapat diuji pada berbagai subset data latih, sehingga hasil evaluasi lebih stabil dan tidak bergantung pada satu set data tertentu.

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

3.5. Hasil K-*Nearest Neighbor*

Model KNN diterapkan pada data latih dan diuji menggunakan data uji. Model ini bekerja dengan mencari sejumlah k tetangga terdekat untuk menentukan kelas suatu sampel berdasarkan mayoritas tetangganya. Pemilihan nilai k yang optimal dilakukan melalui eksperimen, dan hasil terbaik diperoleh dengan $k = 1$. Hasil klasifikasi menggunakan KNN dalam satu kali pelatihan dan pengujian ditampilkan dalam *confusion matrix* berikut:

Tabel 5. Hasil *confusion matrix* KNN

Aktual	Prediksi	
	Positif (0)	Negatif (1)
Positif (0)	681 (TP)	76 (FN)
Negatif (1)	60 (FP)	23 (TN)

Dari Tabel 5 dapat diketahui bahwa model KNN mengklasifikasikan 681 data positif yang diprediksi benar, 76 data negatif yang diprediksi salah, 60 data positif yang diprediksi salah, dan 23 data negatif yang diprediksi benar. Hal ini menunjukkan bahwa klasifikasi pada model KNN mampu mempredksi kategori *rating* film dengan benar sebanyak 704 data dari total 840 data. Berdasarkan hasil evaluasi, model KNN memperoleh nilai akurasi sebesar 83,81%, nilai presisi sebesar 85,12%, nilai sensitivitas sebesar 83,81%, dan nilai F1-*score* sebesar 84,43%.

3.6. Hasil *Decision Tree*

Model *decision tree* (DT) diterapkan sebagai metode klasifikasi berbasis pohon keputusan. Model ini membagi dataset berdasarkan aturan tertentu yang dipelajari dari data latih, sehingga setiap keputusan yang diambil berdasarkan pemisahan fitur yang paling berpengaruh terhadap target klasifikasi. Hasil klasifikasi menggunakan DT dalam satu kali pelatihan dan pengujian ditampilkan dalam *confusion matrix* berikut:

Tabel 6. Hasil *confusion matrix* DT

Aktual	Prediksi	
	Positif (0)	Negatif (1)
Positif (0)	701 (TP)	56 (FN)
Negatif (1)	62 (FP)	21 (TN)

Dari Tabel 6 dapat diketahui bahwa model *decision tree* mengklasifikasikan 701 data positif yang diprediksi benar, 56 data negatif yang diprediksi salah, 62 data positif yang diprediksi salah, dan 21 data negatif yang diprediksi benar. Hal ini menunjukkan bahwa klasifikasi pada model DT mampu memprediksi kategori *rating* film dengan benar sebanyak 722 data dari total 840 data. Berdasarkan hasil evaluasi, model DT memperoleh nilai akurasi sebesar 85,95%, nilai presisi sebesar 85,49%, nilai sensitivitas sebesar 85,95%, dan nilai *F1-score* sebesar 85,72%.

3.7. Hasil *Support Vector Machine*

Dalam penelitian ini, metode *support vector machine* (SVM) digunakan dengan *radial basis function* (RBF) *kernel*. *Kernel* RBF digunakan karena mampu menangani data yang tidak dapat dipisahkan secara linear dengan memetakan data ke dalam dimensi yang lebih tinggi menggunakan fungsi *non-linear*. Hasil klasifikasi menggunakan SVM dalam satu kali pelatihan dan pengujian ditampilkan dalam *confusion matrix* berikut:

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

Tabel 7. Hasil *confusion matrix* SVM

Aktual	Prediksi	
	Positif (0)	Negatif (1)
Positif (0)	576 (TP)	181 (FN)
Negatif (1)	25 (FP)	58 (TN)

Dari Tabel 7 dapat diketahui bahwa model SVM dengan *kernel* RBF mengklasifikasikan 576 data positif yang diprediksi benar, 181 data negatif yang diprediksi salah, 25 data positif yang diprediksi salah, dan 58 data negatif yang diprediksi benar. Hal ini menunjukkan bahwa klasifikasi pada model SVM dengan *kernel* RBF mampu mempredksi kategori *rating* film dengan benar sebanyak 634 data dari total 840 data. Berdasarkan hasil evaluasi, model SVM memperoleh nilai akurasi sebesar 75,48%, nilai presisi sebesar 88,77%, nilai sensitivitas sebesar 75,48%, dan nilai F1-*score* sebesar 80,01%.

3.8. Hasil *Soft Voting*

Metode *soft voting* digunakan sebagai pendekatan *ensemble learning* untuk meningkatkan kinerja klasifikasi dengan menggabungkan hasil prediksi dari beberapa model, yaitu KNN, *decision tree*, dan SVM. *Soft voting* menggunakan rata-rata probabilitas dari setiap model untuk menentukan prediksi akhir. Hasil klasifikasi menggunakan *soft voting* dalam satu kali pelatihan dan pengujian ditampilkan dalam *confusion matrix* berikut:

Tabel 8. Hasil *confusion matrix soft voting*

Aktual	Prediksi	
	Positif (0)	Negatif (1)
Positif (0)	746 (TP)	11 (FN)
Negatif (1)	76 (FP)	7 (TN)

Dari Tabel 8 dapat diketahui bahwa model *ensemble* dengan teknik *soft voting* mengklasifikasikan 746 data positif yang diprediksi benar, 11 data negatif yang diprediksi salah, 76 data positif yang diprediksi salah, dan 7 data negatif yang diprediksi benar. Hal ini menunjukkan bahwa klasifikasi pada metode *soft voting* mampu mempredksi kategori *rating* film dengan benar sebanyak 753 data dari total 840 data. Berdasarkan hasil evaluasi, model *soft voting* memperoleh nilai sebesar 89,64%, nilai *presisi* sebesar 85,63%, nilai sensitivitas sebesar 89,64%, dan nilai F1-*score* sebesar 86,52%.

3.9. Hasil Perbandingan

Proses klasifikasi menggunakan metode k-*nearest neighbor*, *decision tree*, *support vector machine*, dan *soft voting* sudah dilakukan dalam satu kali pelatihan dan pengujian. Untuk memastikan konsistensi performa dari masing-masing model, dilakukan lagi proses pelatihan dan pengujian sebanyak lima kali dengan pembagian data yang berbeda pada setiap pengulangan. Hasil dari lima kali pengujian ini digunakan untuk mengamati stabilitas kinerja model serta keunggulan metode *soft voting* dibandingkan model individual lainnya. Rincian hasil akurasi dari setiap pengulangan ditampilkan pada Tabel 9.

Tabel 9. Hasil nilai akurasi pelatihan dan pengujian berulang pada setiap metode

Pengulangan	KNN	*Decision tree*	SVM	*Soft voting*
1	71,67%	81,19%	79,64%	81,90%
2	72,02%	80,59%	81,31%	84,88%
3	71,79%	82,74%	81,31%	85%
4	73,09%	82,98%	81,67%	84,05%
5	72,02%	80,12%	79,88%	84,05%

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

Berdasarkan hasil akurasi dari lima kali proses pelatihan dan pengujian dari Tabel 9 dapat dilihat perbandingan keempat metode yang menunjukkan bahwa nilai akurasi metode *soft voting* secara konsisten menunjukkan performa yang lebih unggul dibanding metode individu. Nilai akurasi yang diperoleh dari setiap pengulangan relatif stabil dan tetap lebih tinggi dari metode lainnya. Selain itu, dilakukan perbandingan waktu pelatihan antar metode untuk menilai efesiensi komputasi. Hasilnya menunjukkan bahwa *soft voting* memiliki waktu pelatihan lebih tinggi dibanding dengan metode tunggal, namun memberikan hasil evaluasi yang lebih unggul. Dengan demikian, dapat disimpulkan bahwa hasil klasifikasi dengan menggunakan metode *soft voting* menghasilkan nilai akurasi dan performa yang lebih baik dibandingkan dengan metode individu yaitu KNN, *decision tree*, dan SVM.

4. SIMPULAN

Berdasarkan hasil penelitian dan pembahasan dapat disimpulkan metode *ensemble soft voting* mampu meningkatkan performa klasifikasi dibandingkan metode tunggal seperti KNN, *decision tree*, dan SVM. Teknik penyeimbangan data SMOTE diterapkan untuk menyeimbangkan distribusi kelas pada kumpulan data, sehingga model dapat belajar secara lebih optimal dan menghindari bias terhadap kelas mayoritas. Hasil evaluasi dari lima kali pengujian menunjukkan bahwa metode *soft voting*, yang merupakan gabungan dari KNN, *decision tree*, dan SVM, memiliki performa yang paling konsisten dan unggul dalam mengklasifikasikan *rating* film. Dibandingkan dengan masing-masing model tunggal, metode *ensemble* ini mampu menggabungkan kelebihan dari setiap algoritma dasar sehingga memberikan hasil prediksi yang lebih stabil dan akurat. Dengan demikian, pendekatan *soft voting* dapat menjadi alternatif yang lebih andal dalam sistem rekomendasi dan penilaian film, serta berpotensi membantu platform streaming atau pengguna dalam menentukan kualitas suatu film dengan lebih tepat.

DAFTAR PUSTAKA

[1] J. Leander and A. Wicaksana, "Optimizing a personalized movie recommendation system with support vector machine and content-based filtering," *Journal of System and Management Sciences*, vol. 14, no. 1, pp. 490–501, 2024, https://doi.org/10.33168/JSMS.2024.0128.

[2] M. Johari, and A. Laksito, "The hybrid recommender system of the Indonesian online market products using IMDb weight rating and TF-IDF," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 5, pp. 977-983, 2021.

[3] J. Wiratama, and R. S. Oetama, "KNN and naïve bayes algorithms for improving prediction of Indonesian film ratings using feature selection techniques," *In 2023 4th International Conference on Big Data Analytics and Practices (IBDAP) IEEE*, pp. 1-6, 2023.

[4] L. M. Sinaga, S. Sawaluddin, and S. Suwilo, "Analysis of classification and naïve bayes algorithm k-nearest neighbor in data mining," *IOP Conference Series: Materials Science and Engineering*, vol. 725, no. 1, 2020, https://doi.org/10.1088/1757-899X/725/1/012106.

[5] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decision Analytics Journal*, 3 (November 2021), 100071, 2022, https://doi.org/10.1016/j.dajour.2022.100071.

[6] E. Mardiani, N. Rahmansyah, S. Ningsih, D. A. Lantana, A. Suryaningtyas, P. Wirawan, S. A. Wijaya, and D. N. Putri, "Komparasi metode knn, naive bayes, decision tree, ensemble, linear regression terhadap analisis performa pelajar sma," *Innovative: Journal Of Social Science Research*, vol. 3, no. 2, pp. 13880–13892, 2023, http://j-innovative.org/index.php/Innovative/article/view/1949%0Ahttp://j-innovative.org/index.php/Innovative/article/download/1949/1468.

https://doi.org/10.13057/ijas.v8i1.100904
E-ISSN: 2621-086X

[7] H. Hanif and D. W. Utomo, "Prediksi diabetes menggunakan metode ensemble learning dengan teknik soft voting," *Infotekmesin*, vol. 16, no. 01, pp. 127–134, 2025, https://doi.org/10.35970/infotekmesin.v16i1.2534.

[8] S. Joses, D. Yulvida, and S. Rochimah, "Pendekatan metode ensemble learning untuk prakiraan cuaca menggunakan soft voting classifier," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 72–80, 2024, https://doi.org/10.52158/jacost.v5i1.741.

[9] O. S. Atiyah, "Soft voting classifier of machine learning algorithms to predict earthquake," *Al-Kitab Journal for Pure Sciences*, vol. 9, no. 1, pp. 1–13, 2025, https://doi.org/10.32441/kjps.09.01.p1.

[10] B. Prihambodo, A. W. F. Yahya, E. Prayoga, and A. Jaffar, "Klasifikasi kualitas air sungai berbasis teknik data mining dengan metode k-nearest neighbor (k-nn)," *Emitor: Jurnal Teknik Elektro*, vol. 1, no. 1, pp. 31–36, 2023, https://doi.org/10.23917/emitor.v1i1.20833.

[11] C. S. D. Prasetya, "Sistem rekomendasi pada e-commerce menggunakan k-nearest neighbor," *Jurnal Teknologi Informasi Dan Ilmu Komputer*, vol. 4, no. 3, 194, 2017, https://doi.org/10.25126/jtiik.201743392.

[12] F. Akbar, H. W. Saputra, A. K. Maulaya, M. F. Hidayat, and R. Rahmaddeni, "Implementasi algoritma decision tree c4.5 dan support vector regression untuk prediksi penyakit stroke," *MALCOM: Indonesia Journal of Machine Learning and Computer Science*, vol. 2, no. 2, pp. 61–67, 2022, https://doi.org/10.1088/1742-6596/1641/1/012025.

[13] J. Brownlee, *Probability for Machine Learning: Discover how to harness uncertainty with Python*, 2019.

[14] J. Leander, and A. Wicaksana, "Optimizing a personalized movie recommendation system with support vector machine and content-based filtering," *Journal of System and Management Sciences*, vol. 14, no. 1, pp. 490–501, 2024, https://doi.org/10.33168/JSMS.2024.0128.

[15] Z. Mushtaq, M. F. Ramzan, S. Ali, S. Baseer, A. Samad, and M. Husnain, "Voting classification-based diabetes mellitus prediction using hypertuned machine-learning techniques," *Mobile Information Systems*, 2022, https://doi.org/10.1155/2022/6521532.

[16] K. A. Nugraha and D. Sebastian, "Pembentukan dataset topik kata Bahasa Indonesia pada twitter menggunakan TF-IDF & cosine similarity," *Jurnal Teknik Informatika Dan Sistem Informasi*, vol. 4, pp. 2443–2229, 2018, http://dx.doi.org/10.28932/jutisi.v4i3.862.

[17] P. Sokkhey, and T. Okazaki, "Hybrid machine learning algorithms for predicting academic performance," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 32–41, 2020, https://doi.org/10.14569/ijacsa.2020.0110104.

[18] L. Mardiana, D. Kusnandar, and N. Satyahadewi, "Analisis diskriminan dengan k fold cross validation untuk klasifikasi kualitas air di Kota Pontianak," *Buletin Ilmiah Mat. Stat. Dan Terapannya (Bimaster)*, vol. 11, no. 1, pp. 97–102, 2022.

[19] A. Manconi, G. Armano, M. Gnocchi, and L. Milanesi, "A soft-voting ensemble classifier for detecting patients affected by covid-19," *Applied Sciences (Switzerland)*, vol. 12, no. 15, 7554, pp. 1-23, 2022, https://doi.org/10.3390/app12157554.