

KLASIFIKASI DATA SERANGAN JANTUNG MENGGUNAKAN METODE *SUPER VECTOR MACHINE* DAN *ARTIFICIAL NEURAL NETWORK*

Devanska Uzieltama Mardanus

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail: devanska.535220089@stu.untar.ac.id

ABSTRAK

Penelitian ini bertujuan untuk membandingkan kinerja dua algoritma machine learning, yaitu *Super Vector Machine* (SVM) dan *Artificial Neural Network* (ANN), dalam memprediksi risiko serangan jantung. Data yang digunakan diperoleh dari Kaggle pada tahun 2021 dan terdiri dari 10 fitur, termasuk parameter seperti usia, jenis kelamin, dan faktor-faktor lain yang relevan untuk memprediksi risiko serangan jantung. Setelah proses pembersihan dan pembagian data, kedua algoritma dilatih dan diuji menggunakan metrik akurasi, presisi, recall, dan skor F-1. Hasil penelitian menunjukkan bahwa SVM mencapai akurasi sebesar 61%, sementara ANN mencapai akurasi 84%. Berdasarkan hasil ini, dapat disimpulkan bahwa ANN lebih efektif dalam memprediksi risiko serangan jantung dari dataset yang digunakan. Penelitian ini memberikan wawasan mendalam tentang aplikasi dua algoritma machine learning, khususnya SVM dan ANN, dalam konteks prediksi risiko serangan jantung. Hasilnya dapat bermanfaat bagi praktisi kesehatan dan peneliti yang tertarik dalam pengembangan model prediktif untuk pencegahan serangan jantung.

Kata kunci—*Machine Learning, Serangan Jantung, Super Vector Machine, Artificial Neural Network.*

ABSTRACT

This study aims to compare the performance of two machine learning algorithms, namely Support Vector Machine (SVM) and Artificial Neural Network (ANN), in predicting the risk of heart attacks. The data used was obtained from Kaggle in 2021 and consists of 10 features, including parameters such as age, gender, and other relevant factors for predicting the risk of heart attacks. After the data cleaning and splitting process, both algorithms were trained and tested using accuracy, precision, recall, and F-1 score metrics. The research results show that SVM achieved an accuracy of 61%, while ANN achieved 84% accuracy. Based on these results, it can be concluded that ANN is more effective in predicting the risk of heart attacks from the dataset used. This study provides in-depth insights into the application of two machine learning algorithms, specifically SVM and ANN, in the context of predicting the risk of heart attacks. The findings may be beneficial for healthcare practitioners and researchers interested in developing predictive models for heart attack prevention.

Keywords— *Machine Learning, Heart Attack, Super Vector Machine, Artificial Neural Network.*

1. PENDAHULUAN

Salah satu organ terpenting dalam tubuh manusia ialah jantung. Jantung sendiri bertugas untuk memompa darah dari jantung ke paru-paru lalu dari jantung ke seluruh tubuh manusia. Jantung sendiri memiliki 4 ruang yaitu Ventrikel kanan dan kiri serta atrium kanan dan kiri, dimana posisi ventrikel berada di bawah atrium. Jantung juga memiliki katup, katup pada jantung sendiri bertujuan agar tidak terjadi percampuran darah[1]. Kerja jantung sangatlah penting dan kompleks bagi tubuh manusia, maka dari itu kemungkinan jantung terserang penyakit atau mengalami malfungsi sangatlah besar.

Salah satu penyakit yang sangat mematikan ialah serangan jantung. Serangan jantung menjadi penyebab kematian tertinggi di dunia yang dijuluki sebagai sebutan pembunuh secara diam-diam[2]. Pengertian serangan jantung sendiri ialah gangguan jantung Ketika otot jantung tidak mendapat aliran darah yang cukup. Kondisi ini akan mengganggu fungsi jantung untuk mengalirkan darah ke seluruh tubuh[3]. Tentu hal demikian bisa terjadi karena banyak faktor yang terjadi dalam tubuh kita, seperti kurangnya pola hidup sehat, sering mengonsumsi zat-zat yang membahayakan bagi tubuh, serta kurang menjaga waktu tidur dan kurang berolahraga.

Di Indonesia sendiri penyakit jantung menjadi penyebab kematian paling tinggi. Data pada tahun 2021 menunjukkan khusus penyakit jantung sekitar 12,93juta kasus. Bahkan di tahun 2022 kasus kematian akibat serangan jantung meningkat cukup signifikan mejadi 15,5 juta kasus[4]. Maka dari itu perlu ada peninjauan khusus untuk melihat dan mencari tau resiko dan penyebab serangan jantung. Data pada *Kaggle* yang dimuat 3 tahun lalu berjudul *Heart Attack Analysis and Prediction Dataset* bisa menjadi alat untuk melakukan klasifikasi data yang bertujuan untuk memprediksi kemungkinan manusia terkena serangan jantung. Serta mengambil tindakan secepat mungkin untuk mencegah hal yang tidak diinginkan.

2. METODE PENELITIAN

Untuk mendapatkan hasil yang maksimal dalam melakukan metode penelitan. Ada beberapa langkah-langkah yang harus dipenuhi dalam memproses data yang ada. Dimulai dari tahap pengumpulan data, diikuti dengan tahap *preprocessing* data untuk membersihkan dan merapihkan *dataset*. Setelah *dataset* rapih, *dataset* harus di bagi menjadi dua bagian, yaitu data latih dan data uji. Lalu dilakukan pengujian menggunakan metode klasifikasi. Dalam penelitian ini ada dua algoritma yang digunakan, yaitu *Support Vector Machine (SVM)* dan *Artificial Neural Network (ANN)*[5].

2.1 Klasifikasi

Klasifikasi adalah suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target yang terorganiasi ke dalam *supervised learning* atau pembelajaran yang diawasi. Maksud dari *supervised learning* adalah data label atau target ikut berperan sebagai guru yang mengawasi proses pembelajaran dalam mencapai tingkat akurasi atau presisi tertentu[6].

2.2 Preprocessing Data

Data *preprocessing* merupakan tahapan selanjutnya dalam melakukan mining data [7]. Tahap ini cukup krusial, karena data mentah akan diolah terlebih dahulu. Data *preprocessing* biasanya dilakukan dengan *system* eliminasi bila data yang tidak sesuai atau tidak dibutuhkan. Ada beberapa tahapan dalam melakukan data *preprocessing*, yaitu data cleaning, data integration, data transformation, dan data reduction [8]. Semua proses itu dilakukan untuk mendapatkan data yang bersih dan teratur.

2.3 Support Vector Machine (SVM)

Algoritma *Support Vector Machine (SVM)* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi[9]. Algoritma *SVM* dapat mengklasifikasikan data dengan cara menemukan *hyperplane* ini dipilih sedemikian rupa sehingga jarak antara *hyperplane* dan titik- titik data yang terdekat disebut vektor pendukung maksimum[10]. Berikut adalah rumus (1) umum untuk *SVM* dalam konteks klasifikasi sebagai berikut.

$$f(x) = \text{sign}\left(\sum_{i=1}^n a_i y_i K(x, x_i) + b\right) \quad (1)$$

dimana:

- a) $f(x)$ adalah fungsi keputusan.
- b) x adalah data uji.
- c) n adalah jumlah sampel dalam set pelatihan.
- d) y_i adalah label kelas dari sampel ke i .
- e) a_i adalah koefisien Lagrange.
- f) $K(x, x_i)$ adalah fungsi kernel, yang menghitung produk titik antara dua vektor dalam ruang fitur.
- g) b adalah bias.

Langkah – Langkah untuk menerapkan SVM [11] :

1. Memproses Data yang ada.
2. Memilih fitur yang dibutuhkan.
3. Pembagian Data.
4. Membuat model *Support Vector Machine* (SVM).
5. Validasi Model.
6. Penyetelan Hyperparameter.
7. Pengujian dan Evaluasi Tambahan.
8. Implemtasi Model.

2.4 *Artificial Neural Network* (ANN)

Jaringan Syaraf Tiruan (JST) adalah salah satu pendekatan paling ampuh dalam pembelajaran mesin yang memiliki kemampuan untuk memodelkan dan menangani masalah yang sangat kompleks. Terinspirasi oleh cara kerja jaringan saraf biologis manusia, ANN terdiri dari sejumlah besar neuron buatan yang terhubung satu sama lain [12]. Setiap neuron, seperti dalam sistem biologis, menerima masukan, memrosesnya, dan mengirimkan keluaran ke neuron berikutnya. Organisasi-organisasi ini menciptakan jaringan pemrosesan informasi yang sangat adaptif dan fleksibel, yang mampu mengenali pola kompleks dalam data. Lapisan masukan berfungsi sebagai pintu masuk data, lapisan tersembunyi yang terletak di antara lapisan masukan dan keluaran memproses informasi secara bertahap, sedangkan lapisan keluaran menghasilkan hasil akhir dari proses komputasi[13].

Kekuatan utama ANN terletak pada kemampuannya belajar dari data. Dengan memberikan contoh pelatihan dalam jumlah besar yang terdiri dari input dan output yang diinginkan, ANN mampu menyesuaikan bobot koneksi antar neuronnya secara otomatis melalui proses yang disebut pembelajaran. Hal ini memungkinkan ANN untuk menangani berbagai macam tugas, mulai dari pengenalan gambar hingga prediksi harga saham, tanpa perlu diatur ulang secara manual untuk setiap tugas baru. Kemajuan dalam teknologi komputasi dan algoritma pembelajaran mesin telah memperluas cakupan dan kemampuan ANN, menjadikannya salah satu alat paling populer dan kuat dalam analisis data modern [14].

2.4.1 *Neuron*

Setiap neuron menerima masukan, memrosesnya menggunakan fungsi aktivasi, dan mengirimkan keluaran ke neuron berikutnya. Matematika di balik pengoperasian neuron dalam suatu jaringan adalah sebagai berikut[15]:

- a) Input (x): Nilai input yang diterima oleh neuron.
- b) Bobot (w): Nilai bobot yang diberikan pada setiap masukan, menunjukkan pentingnya setiap masukan dalam pengambilan keputusan neuron.
- c) Fungsi Aktivasi (f): Fungsi yang diterapkan pada jumlah tertentu dari produk input dan bobot. Hal ini memungkinkan neuron menghasilkan keluaran non-linier.

- d) Bias (b): Nilai yang ditambahkan ke sejumlah produk masukan dan bobot tertentu sebelum menerapkan fungsi aktivasi. Hal ini memungkinkan neuron untuk belajar dari masukan yang tidak sempurna atau tidak lengkap.

Rumus umum (2) pengoperasian neuron dalam suatu jaringan adalah:

$$output = f\left(\sum_{i=1}^n (w_i x_i) + b\right) \quad (2)$$

Langkah-Langkah Implementasi ANN [16]:

1. Pemilihan Arsitektur: Menentukan jumlah lapisan dan jumlah neuron di setiap lapisan (termasuk lapisan masukan, tersembunyi, dan keluaran).
2. Inisialisasi Bobot dan Bias: Bobot dan bias awal biasanya diinisialisasi secara acak dengan nilai kecil.
3. Propagasi Maju: Melakukan perhitungan dari input ke output melalui jaringan, menggunakan rumus neuron yang telah dijelaskan sebelumnya.
4. Perhitungan Kesalahan: Menghitung kesalahan antara keluaran yang dihasilkan dan keluaran yang diharapkan menggunakan metode seperti *Mean Squared Error* (MSE) atau *Cross Entropy*.
5. Propagasi mundur: Lanjutkan mundur melalui jaringan, sesuaikan bobot dan bias untuk memperhitungkan kesalahan gradien terhadap bobot dan bias.
6. Pengoptimalan: Terapkan pengoptimalan algoritme seperti *Stochastic Gradient Descent* (SGD) atau Adam untuk mengoptimalkan bobot dan bias.
7. Validasi dan Evaluasi: Mengevaluasi kinerja jaringan pada data validasi atau pengujian untuk memastikan bahwa model dapat menggeneralisasi dengan baik pada data yang sebelumnya tidak terlihat.

2.5 Confusion Matrix

Confusion Matrix adalah alat evaluasi penting dalam pembelajaran mesin, memberikan gambaran yang jelas tentang seberapa baik model klasifikasi mampu mengidentifikasi kelas dalam data. Dengan menyajikan informasi tentang *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN), matriks konfusi memungkinkan analisis mendalam tentang kekuatan dan kelemahan model [17]. TP dan TN mewakili keberhasilan dalam mengklasifikasikan data positif dan negatif dengan benar, sedangkan FP dan FN menyoroti kesalahan dalam klasifikasi. Dari matriks konfusi, metrik evaluasi seperti akurasi, presisi, perolehan, dan skor F1 dapat dihitung, sehingga memberikan pemahaman komprehensif tentang performa model secara keseluruhan [18].

Penggunaan matriks konfusi tidak hanya memberikan gambaran kinerja model yang lebih jelas, namun juga memungkinkan identifikasi area di mana model dapat ditingkatkan. Dengan menganalisis setiap sel dalam matriks, pola klasifikasi dapat diidentifikasi yang mungkin memerlukan perhatian khusus. Selain itu, metrik evaluasi yang dihasilkan dari matriks konfusi memungkinkan perbandingan objektif antara model klasifikasi yang berbeda. Dengan demikian, matriks konfusi tidak hanya sebagai alat evaluasi, tetapi juga merupakan instrumen penting dalam pengembangan dan peningkatan model pembelajaran mesin [19].

Tabel 1. Confusion Matrix.

	Predicted Negative (0)	Predicted Positive (1)
Actual Negative (0)	True Negative (TN)	False Positive (FP)
Actual Positive (1)	False Negative (FN)	True Positive (TP)

Penjelasan untuk setiap sel dalam confusion matrix [20]:

1. *True Positive* (TP): Merupakan jumlah data yang diklasifikasikan dengan benar sebagai positif oleh model. Misalnya, dalam konteks deteksi spam email, TP adalah jumlah email yang benar-benar spam dan berhasil diidentifikasi sebagai spam oleh model.
2. *True Negative* (TN): Merupakan jumlah data yang diklasifikasikan dengan benar sebagai negatif oleh model. Misalnya, dalam konteks deteksi penyakit, TN adalah jumlah pasien yang sehat dan berhasil diidentifikasi sebagai sehat oleh model.
3. *False Positive* (FP): Merupakan jumlah data yang salah diklasifikasikan sebagai positif oleh model. Ini berarti model salah mengklasifikasikan data negatif sebagai positif. Dalam kasus deteksi spam email, FP adalah jumlah email yang tidak spam namun salah diklasifikasikan sebagai spam.
4. *False Negative* (FN): Merupakan jumlah data yang salah diklasifikasikan sebagai negatif oleh model. Ini berarti model gagal mengidentifikasi data positif sebagai positif. Misalnya, dalam deteksi penyakit, FN adalah jumlah pasien yang sakit namun salah diidentifikasi sebagai sehat oleh model.

Ada empat parameter yang dapat digunakan untuk mengukur *performance metrics* yaitu sebagai berikut [21].

1. Accuracy: Nilai rasio dari jumlah prediksi benar terhadap jumlah data yang di prediksi. Nilai accuracy dapat dihitung dengan persamaan (3).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

2. Precision: Nilai rasio dari jumlah true positive dan jumlah keseluruhan prediksi positif. Nilai precision dapat dihitung dengan persamaan (4).

$$precision = \frac{TP}{TP + FP} \quad (4)$$

3. Recall: Nilai rasio antara jumlah true positive dan jumlah keseluruhan yang sebenarnya positif. Nilai recall dapat dihitung dengan persamaan (5).

$$recall = \frac{TP}{TP + FN} \quad (5)$$

4. F1-Score: Nilai yang menghitung harmonic mean dari recall dan precision. Nilai F1-Score dapat dihitung dengan persamaan (6).

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (6)$$

3. HASIL DAN PEMBAHASAN

Pada bagian pembahasan, disajikan hasil eksperimen yang telah dilakukan di Visual Studio Code. *Dataset* yang digunakan adalah *Heart Attack Analysis & Prediction Dataset*, yang bersumber dari Kaggle <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset> *Dataset* ini terdiri dari 1000 lebih data dengan 10 fitur. Untuk mendapatkan hasil klasifikasi dilakukan dengan algoritma *Support Vector Machine* (SVM) dan *Artificial Neural Network* (ANN). Bahasa pemrograman yang akan digunakan adalah python. Dikarenakan python memiliki modul yang memudahkan Analisa yang ada. Tujuan dari eksperimen ini adalah untuk mengevaluasi kedua algoritma, serta mencari tau algoritma manakan yang lebih sesuai untuk *dataset* ini.

Dataset ini dikumpulkan dari beberapa pasien rumah sakit di beberapa tempat. Pengumpulan data di rumah sakit menggunakan bantuan alat dari *Internet of Things* (IOT). Karena tanpa bantuan dari alat *IOT* akan sulit mengetahui data yang ada. Berikut adalah 10 fitur yang disajikan dalam data yang ada, berikut adalah penjelasannya:

- a. **Age** : Umur.
- b. **Sex** : Jenis kelamin patient.
- c. **Exang**: angina akibat olahraga (1 = ya; 0 = tidak).
- d. **Ca**: jumlah kapal besar (0-3).
- e. **Cp** : Tipe Nyeri Dada Tipe nyeri dada.
 - Nilai 1: angina tipikal
 - Nilai 2: angina atipikal
 - Nilai 3: nyeri nonangina
 - Nilai 4: tanpa gejala
- a. **Trtbps**: tekanan darah istirahat (dalam mm Hg).
- b. **Chol**: kolestoral dalam mg/dl diambil melalui sensor BMI.
- c. **Fbs**: (gula darah puasa > 120 mg/dl) (1 = benar; 0 = salah).
- d. **Rest_ecg**: hasil elektrokardiografi istirahat
 - Nilai 0: normal.
 - Nilai 1: mengalami kelainan gelombang ST-T (inversi gelombang T dan/atau elevasi atau depresi ST > 0,05 mV).
 - Nilai 2: menunjukkan kemungkinan atau pasti hipertrofi ventrikel kiri berdasarkan kriteria Estes.
1. **Thalach**: detak jantung maksimum tercapai.
 - **Target**: 0 = lebih kecil kemungkinan terkena serangan jantung 1 = lebih besar kemungkinan terkena serangan jantung.

Tahap *preprocessing* Perlu dilakukan terlebih dahulu, untuk memastikan kualitas data yang ada. Langkah-langkah *preprocessing* melibatkan penghapusan nilai null dan data duplikat dari *dataset*. Setelah melakukan *preproessing dataset* akan dibagi menjadi 70:30. 70% data latih dan 30 % uji. Lalu algoritma SVM dilakukan dengan 5 kali iterasi , detail hasil akurasi dapat dilihat pada Tabel 2.

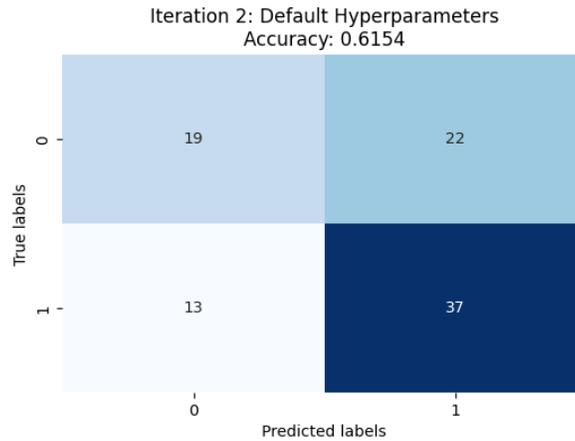
Tabel 2. Akurasi Algoritma SVM.

No.	Iterasi ke - x	Akurasi
1.	Iterasi ke -1	61,45%
2.	Iterasi ke -2	61,54%
3.	Iterasi ke -3	61,54%
4.	Iterasi ke -4	61,54%
5.	Iterasi ke -5	61,54%

Dari tabel di atas, terlihat bahwa akurasi yang diperoleh untuk setiap iterasi, cukup baik. Perbedaan setiap iterasinya juga sangat kecil. Namun yang memiliki keadaan yang paling tinggi adalah iterasi no 2. Berikut adalah SVM evaluationnnya.

Tabel 3. Iterasi ke 2

No.	Metriks	KNN
1.	Akurasi	62%
2.	Precision	61%
3.	Recall	60%
4.	F1-Score	60%



Gambar 1. Confusion Matrix Algoritma SVM dengan nilai iterasi – 2.

Gambar 1 menunjukkan bahwa dari 19 data uji dengan outputnya “0” (tidak ada serangan jantung), model klasifikasi algoritma SVM berhasil memprediksi 19 data uji dengan benar. Dan model SVM melakukan salah prediksi sebanyak 22 data. Sedangkan sisa datanya dinyatakan aman dan tidak memiliki serangan jantung.

Sekarang masuk dalam pengujian menggunakan algoritma ANN, dengan iterasi 1 sampai 5 pengujian, dengan mencapai maximum iter sebanyak 100. Untuk menentukan jumlah iterasi yang diberikan memiliki akurasi yang baik, dan diteailnya dapat di lihat dalam tabel 4.

Tabel 4. Akurasi Algoritma ANN.

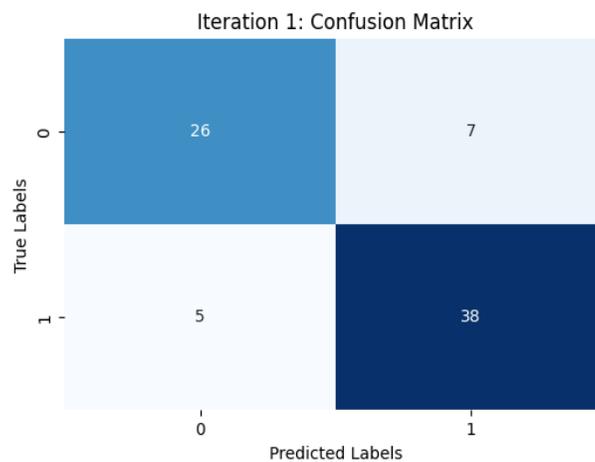
No.	Iterasi ke - x	Akurasi
1.	Iterasi ke -1	84%
2.	Iterasi ke -2	76%
3.	Iterasi ke -3	80%
4.	Iterasi ke -4	83%
5.	Iterasi ke -5	74%

Dari tabel di atas, terlihat bahwa nilai akurasi dari setiap iterasi algoritma Artificial Neural Network (ANN) bervariasi. Iterasi ke-1 mencapai akurasi tertinggi dengan 84%, sementara iterasi ke-5 memiliki akurasi terendah sebesar 74%. Hal ini menunjukkan bahwa kinerja algoritma ANN dapat bervariasi dari satu iterasi ke iterasi lainnya. Penurunan atau peningkatan dalam akurasi antar iterasi dapat dipengaruhi oleh faktor-faktor seperti konvergensi model, variasi data latih, atau parameter algoritma yang diatur. Dengan memantau perubahan akurasi dari iterasi ke iterasi, kita dapat mengidentifikasi tren dan mengoptimalkan algoritma untuk mencapai kinerja yang lebih baik.

Tabel 5. Iterasi ke 1.

No.	Metriks	KNN
1.	Akurasi	84%
2.	Precision	84%
3.	Recall	84%
4.	F1-Score	84%

Dalam tabel yang diberikan, metrik-metrik evaluasi performa untuk algoritma *K-Nearest Neighbors* (KNN) disajikan secara terperinci. Akurasi, precision, recall, dan F1-Score semuanya menunjukkan hasil yang konsisten, yaitu 84%. Hal ini menandakan bahwa algoritma KNN memiliki kinerja yang stabil dan mampu mengklasifikasikan data dengan tingkat keakuratan yang tinggi. Akurasi mencerminkan proporsi total prediksi yang benar, sementara precision mengukur seberapa banyak prediksi positif yang benar dari semua prediksi positif yang dibuat. Recall, di sisi lain, mengukur seberapa banyak dari semua kelas positif yang berhasil diidentifikasi oleh model. F1-Score adalah perpaduan antara precision dan recall, memberikan nilai tunggal yang mencerminkan keseimbangan antara keduanya. Dengan melihat metrik-metrik ini, kita dapat mendapatkan pemahaman yang komprehensif tentang kinerja algoritma KNN dalam melakukan klasifikasi data.



Dari matriks kebingungan yang diberikan:

1. Terdapat 26 True Negatives (TN), yaitu jumlah data negatif yang benar diklasifikasikan sebagai negatif.
2. Terdapat 7 False Positives (FP), yang mewakili jumlah data negatif yang salah diklasifikasikan sebagai positif.
3. Terdapat 5 False Negatives (FN), yaitu banyaknya data positif yang salah diklasifikasikan sebagai negatif.
4. Terdapat 38 True Positives (TP) yaitu banyaknya data positif yang benar diklasifikasikan positif.

Dengan menggunakan matriks konfusi ini, kita dapat menghitung berbagai metrik evaluasi seperti akurasi, presisi, perolehan, dan skor F1 untuk menghasilkan performa klasifikasi model. Sekarang kita akan buat tabel perbandingan antara SVM dan ANN.

Tabel 6. Perbandingan metrik evaluasi antara algoritma KNN dan ANN

No.	Algoritma	Akurasi	Precision	Recall	F1-Score
1	SVM	62%	61%	60%	60%
2	ANN	84%	84%	84%	84%

4 KESIMPULAN

Dari tabel perbandingan metrik evaluasi antara algoritma *K-Nearest Neighbors* (KNN) dan *Artificial Neural Network* (ANN), terlihat bahwa ANN menunjukkan performa yang jauh lebih baik dibandingkan dengan KNN. Dengan nilai akurasi, presisi, recall, dan F1-Score yang konsisten sebesar 84%, ANN menunjukkan kemampuan yang lebih baik dalam mengklasifikasikan data dibandingkan dengan KNN yang hanya mencapai 62% akurasi, serta 61% precision, 60% recall, dan 60% F1-Score. Kesimpulan ini menunjukkan bahwa ANN mungkin menjadi pilihan yang lebih baik dalam konteks pemrosesan data yang kompleks dan memerlukan klasifikasi yang akurat.

UCAPAN TERIMA KASIH

Saya Devanska Uzieltama Mardanus berterimakasih kepada Tuhan Yang Maha Esa karena telah diberikan kelancaran untuk melakukan penelitian ini. Serta ibu Teny Handhayani S.Kom., M.Kom., Ph.D., yang telah memberi dukungan terhadap penelitian ini. Jurnal Computatio berterima kasih kepada IJCCS sebagai landasan bentuk format makalah ini.

DAFTAR PUSTAKA

- [1] Thasya Aniamarta, Alifia Salsabilla Huda, Febriana Lizariani Aqsha, 2022, S, *Jurnal Biologica Samudra*, No. 4, Vol. 1, hal. 2, <https://ejurnalunsam.id/index.php/jbs/article/download/3925/3394/> , 2022, *Penyebab dan Pengobatan Serangan Jantung*, DKI Jakarta.
- [2] Astasari, 2022, 5 Faktor Pemicu Gejala Serangan Jantung, <https://ayosehat.kemkes.go.id/5-faktor-pemicu-gejala-serangan-jantung> , diakses tanggal 01 Mei 2024.
- [3] dr. Pittara, 2022, Pengertian Serangan Jantung, <https://www.alodokter.com/serangan-jantung> , diakses tanggal 01 Mei 2024.
- [4] Tri Rini Puji Lestari, 2023, Mengenal Kesehatan Jantung Melalui Deteksi Dini, https://berkas.dpr.go.id/puslit/files/isu_sepekan/Isu%20Sepekan---IV-PUSLIT-September-2023-196.pdf , diakses tanggal 02 Mei 2024.
- [5] Asep Sumpena Nugraha, Ken Kinanti Purnamasari, Penerapan Metode Support Vector Machine Pada Part Of Speech Tag Bahasa Indonesia, Jakarta, 2024.
- [6] Mohamad Frananda Adiezwar Ramadhan, 2023, Portofolio New Classification Machine Learning, <https://bisa.ai/portofolio/detail/MTI3Nw>, diakses tanggal 02 Mei 2024.
- [7] Suropto, Rr Nurul Rahmanita, Ajeng Sekar Kirana, 2022, Teknik Pre-processing dan Classification Dalam Data Science, <https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-data-science/>, diakses tanggal 02 Mei 2024.
- [8] Andre Oliver, 2023, Bikin Data Lebih Mudah Dibaca Yuk Kenalan Dengan Data Preprocessing, <https://glints.com/id/lowongan/data-preprocessing-adalah/> , diakses tanggal 02 Mei 2024.
- [9] Christianini, N., dan Shawe-Taylor, J., Support Vector Machine, *Cambrige University Press* London, 2000.
- [10] Bishop, C. M., Pattern Recognition and Machine Learning, *Springer*, Ohayo, 2006.
- [11] Pedregosa, F, Varoquaux, G, Gramfort, A., Michel, V., Thirion, B., Grisel, O., 2011, "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, 12(Oct), 2825-2830.
- [12] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, *MIT Press*, Cambridge M.A, 2016.
- [13] Simon Haykin, Neural Networks and Learning Machines, *Pearson*, Upper Saddle River, NJ, 2009.
- [14] Yegnanarayana B, Artificial Neural Networks, *PHI Neural Networks*, New Delhi, 2010.
- [15] Raul Rojas, Neural Networks: A Systematic Introduction, *Springer*, Berlin, Heidelberg, 1996.
- [16] Kevin L.Pridy, Paul E. Keller, Artificial Neural Networks: An Introduction, SPIE Press, Bellingham, WA, 2005.
- [17] Jason Brownlee, Machine Learning Mastery With Python: Understand Your Data, Create Accurate Models and Work Projects End-to-End, *Independently Published*, Scotts Valley, CA, 2016.

- [18] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, *Springer*, New York. NY, 2009.
- [19] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning: with Applications in R, *Springer*, New York, NY., 2013.
- [20] Sebastian Raschka, Vahid Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, *Packt Publishing*, Birmingham, UK., 2019.
- [21] Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, *O'Reilly Media*, Sebastopol, CA. 2019.