Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

Perbandingan Kinerja Klasifikasi Penyakit Ginjal Menggunakan Algoritma Support Vector Machine (SVM) dan Decision Tree (DT)

Puja Milenia Sriwildan Madani, Tatang Rohana*, Kiki Ahmad Baihaqi, Ahmad Fauzi

Fakultas Teknik Informatika, Program Studi Teknik Informatika, Universitas Buana Perjuangan, Karawang, Indonesia
Email: [1]if20.pujamileniasriwildanmadani@mhs.ubpkarawang.ac.id, [2,*]tatang.rohana@ubpkarawang.ac.id,
[3]kikiahmad@ubpkarawang.ac.id, [4]afauzi@ubpkarawang.ac.id
Email Penulis Korespondensi: tatang.rohana@ubpkarawang.ac.id
Submitted: **21/05/2024**; Accepted: **23/06/2024**; Published**: 23/06/2024**

Abstrak−Penyakit Ginjal Kronis merupakan salah satu penyakit yang mematikan. Pada stadium awal penyakit ini bisa saja tidak terdeteksi keberadaannya, sehingga pasien cenderung menganggap sepele, namun, penyakit ini bisa berkembang sedikit demi sedikit dan menjadi serius tanpa terdeteksi. Hal tersebut dapat menimbulkan komplikasi penyakit lain serta dapat mengakibatkan organ ginjal rusak permanen. Oleh karena itu, penelitian ini bertujuan untuk melakukan klasifikasi terhadap individu yang berisiko memiliki Penyakit Ginjal Kronis yang dapat membantu tenaga medis dalam upaya mengurangi jumlah penderita penyakit tersebut. Penelitian ini menggunakan data Penyakit Ginjal Kronis yang diperoleh dari web UCI Repository. Data tersebut memiliki 25 atribut dengan 400 baris. Penelitian ini membandingkan algoritma *Support Vector Machine* dan *Decision Tree* serta menggunakan metode evaluasi *Confusion Matrix*. Hasil penelitian menunjukkan bahwa algoritma *Support Vector Machine* memiliki hasil akurasi, presisi, *recall*, dan f1-*score* yang lebih unggul dibanding algoritma *Decision Tree*. Hasil akurasi algoritma *Support Vector Machine* sebesar 97.5, presisi sebesar 0.98, *recall* sebesar 0.96, dan f1-*score* sebesar 0.97. Sedangkan pada algoritma *Decision Tree* akurasi yang diperoleh sebesar 92.5, presisi sebesar 0.92, *recall* sebesar 0.90, dan f1-*score* sebesar 0.91. dengan hasil tersebut, penelitian ini dapat dilanjutkan menjadi sebuah aplikasi yang dapat mengklasifikasikan individu yang berisiko terkena Penyakit Ginjal Kronis.

Kata Kunci: Klasifikasi; Penyakit Ginyal Kronik; Support Vector Machine; Decision Tree; Confusion Matrix

Abstract−Chronic Kidney Disease is one of the deadliest diseases. In the early stages, the disease may go undetected, so patients tend to take it lightly, however, the disease can progress little by little and become serious without being detected. This can lead to complications of other diseases and can cause permanent damage to the kidney organs. Therefore, this study aims to classify individuals who are at risk of having Chronic Kidney Disease which can help medical personnel in an effort to reduce the number of people with the disease. This study uses Chronic Kidney Disease data obtained from the UCI Repository web. The data has 25 attributes with 400 rows. This research compares the Support Vector Machine and Decision Tree algorithms and uses the Confusion Matrix evaluation method. The results showed that the Support Vector Machine algorithm has superior accuracy, precision, recall, and f1-score results compared to the Decision Tree algorithm. The accuracy of the Support Vector Machine algorithm is 97.5, precision is 0.98, recall is 0.96, and f1-score is 0.97. While the Decision Tree algorithm obtained accuracy of 92.5, precision of 0.92, recall of 0.90, and f1-score of 0.91. with these results, this research can be continued into an application that can classify individuals at risk of Chronic Kidney Disease.

Keywords: Classification; Chronic Kidney Disease; Support Vector Machine; Decision Tree; Confusion Matrix

1. PENDAHULUAN

Pada saat ini isu kesehatan yang menjadi permasalahan global dan semakin mendesak adalah Penyakit Ginjal Kronis (PGK). Penyakit ini tidak memiliki gejala pada stadium awal, namun penyakit ini bisa berkembang sedikit demi sedikit dan menjadi serius tanpa terdeteksi. Selain itu, apabila penyakit ini tidak terdeteksi dengan baik dapat menimbulkan komplikasi penyakit lain serta dapat mengakibatkan organ ginjal rusak permanen [1].

Berdasarkan laporan dari *World Health Organization* (WHO) pada tahun 2018, jumlah kasus Penyakit Ginjal Kronis (PGK) di seluruh dunia mencapai 188 juta [2]. Berdasarkan hasil Riset Kesehatan Dasar (RISKESDAS) tahun 2018 yang dilakukan oleh Badan Penelitian dan Pengembangan Kesehatan, prevalensi PGK di Indonesia mencapai 0,38% atau sekitar 3,8 orang per 1000 penduduk. Dari jumlah ini, sekitar 60% penderita gagal ginjal memerlukan dialisis [3].

Oleh karena itu dibutuhkan pemahaman yang mendalam terkait faktor apa saja yang menjadi risiko penyakit tersebut, penyebab seseorang mengalami penyakit tersebut, dan metode yang digunakan untuk mendeteksi penyakit tersebut untuk dapat mendiagnosis, mengobati dan melakukan pencegahan terhadap Penyakit Ginjal Kronis. Pendekatan pada penelitian ini dalam klasifikasi adalah menggunakan pengklasifikasi pembelajaran terawasi menggunakan algortima *Support Vector Machine* (SVM) dan *Decision Tree* (DT).

Terdapat beberapa penelitian sebelumnya yang melakukan klasifikasi terhadap Penyakit Ginjal Kronis antara lain, penelitian yang dilakukan oleh A. Ariani, mengklasifikasikan Penyakit Ginjal Kronis menggunakan K-*Nearest Neighbor* dengan data Penyakit Ginjal Kronis. Hasil akurasi yang didapat sebesar 85,3% [4]. Selanjutnya pada penelitian dengan membandingkan algoritma Naïve Bayes *dan Support Vector Machine*, dan *Random Forest* dalam mengklasifikasi penyakit ginjal yang dilakukan oleh I. I. M. Rizky, S. Y. Irianto, dan Sriyanto, memperoleh hasil akurasi paling tinggi sebesar 99,64% dari algoritma *Random Forest*, *Support Vector Machine* sebesar 92,50%, dan *Naïve Bayes* sebesar 97,14% [5].

Selanjutnya, penelitian yang dilakukan M. Rizal, M. Zakhy Syahaf, S. Rully Priyambodo, Y. Ramdhani, and U. Adhirajasa Reswara Sanjaya, membandingkan algoritma C4.5, K-NN, *Logistic Regression* dan *Naïve Bayes* dengan menambahkan metode *Forward Selection*. Hasil akurasi yang diperoleh adalah *Naïve Bayes* memiliki akurasi tertinggi

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

yaitu sebesar 92,92% [6]. Penelitian menggunakan algoritma *Adaboos*t yang dibandingkan dengan algoritma C4.5 dilakukan T. Asra, A. Setiadi, M. Safudin, E. W. Lestari, N. Hardi, and D. P. Alamsyah, dalam mengklasifikasikan penyakit ginjal memperoleh hasil akurasi sebesar 96% pada algoritma *Adaboost*, sedangkan algoritma C4.5 memperoleh akurasi sebesar 95% [7]. Algoritma *K-Means* dengan bantuan metode *Binary Particle Swarm Optimization* dalam mengklasifikasikan penyakit ginjal yang dilakukan I. G. A. M. Pratama, L. G. Astuti, Wiartha I Made, I. G. N. A. C. Putra, C. R. A. Pramartha, and Darmawan I Dewa Made Bayu Atmaja, memperoleh hasil akurasi sebesar 96,875% [8].

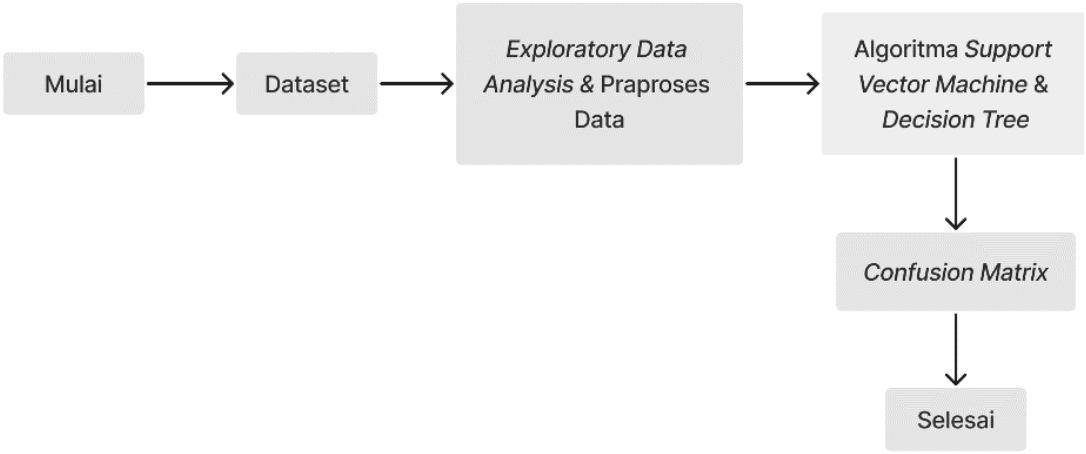
Mengacu pada permasalahan yang telah diuraikan sebelumnya dan didukung oleh penelitian terkait penyakit ginjal kronis, kami memutuskan untuk membandingkan kinerja dua algoritma populer, yaitu *Support Vector Machine* (SVM) dan *Decision Tree*, dalam mengklasifikasikan penyakit ginjal kronis dengan menggunakan pembelajaran terawasi (*Supervised Learning*). Tujuan dari penelitian ini adalah untuk mengidentifikasi perbedaan dalam nilai akurasi antara kedua algoritma yang digunakan dan mengevaluasi kinerja penelitian sebelumnya dengan menggunakan data penyakit ginjal kronis yang sama. Dengan demikian, diharapkan bahwa model yang dikembangkan melalui penelitian ini dapat memberikan kontribusi yang signifikan dalam meningkatkan kemampuan para ahli dalam melakukan penilaian yang lebih konsisten dan efisien terkait penyakit ginjal kronis, serta memberikan dasar yang lebih kuat untuk pengambilan keputusan dalam diagnosis dan pengelolaan penyakit ini.

Penelitian ini diharapkan dapat memberikan wawasan yang lebih mendalam tentang kemampuan kedua algoritma dalam mengklasifikasikan data penyakit ginjal kronis, serta memberikan dasar yang lebih kokoh bagi pengembangan model prediktif yang dapat digunakan dalam praktek klinis. Dengan menggunakan data yang ada dan metode yang tepat, diharapkan hasil penelitian ini dapat memberikan solusi yang lebih efektif dalam menghadapi tantangan yang kompleks terkait diagnosis dan pengobatan penyakit ginjal kronis. Dengan demikian, langkah-langkah preventif dan terapeutik yang diterapkan dapat menjadi lebih efisien dan efektif, sehingga meningkatkan kualitas hidup pasien yang terkena penyakit ini dan mengurangi dampak negatifnya secara keseluruhan.

2. METODOLOGI PENELITIAN

Metode penelitian merupakan langkah-langkah yang dilakukan untuk mencapai tujuan penelitian. Langkah-langkah dalam penelitian ini antara lain Pengumpulan dataset, *Exploratory Data Analysis* (EDA), implementasi perbandingan algoritma, dan evaluasi menggunakan *confusion matrix*.

2.1 Tahapan Penelitian



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, tahapan pertama dari penelitian ini dimulai dengan pengumpulan dataset. Dataset yang digunakan adalah dataset Penyakit Ginjal Kronis yang tersedia di situs web resmi Kaggle, yang dapat diakses melalui tautan https://www.kaggle.com/. Dataset ini terdiri dari 25 atribut dan 400 baris data. Atribut-atribut tersebut mencakup berbagai informasi yang relevan dengan studi ini, seperti *"id"*, *"age"*, *"blood pressure"*, *"specific gravity"*, *"albumin"*, *"sugar"*, *"red blood cells"*, *"pus cell"*, *"pus cell clumps"*, *"bacteria"*, *"blood glucose random"*, *"blood urea"*, *"serum creatinine"*, *"sodium"*, *"potassium"*, *"hemoglobin"*, *"packed cell volume"*, *"white blood cell count"*, *"red blood cell count"*, *"hypertension"*, *"diabetes mellitus"*, *"coronary artery disease"*, *"appetite"*, *"pedal edema"*, *"anemia"*, dan *"classification"*. Tabel 1 menunjukkan sebagian kecil dari dataset yang digunakan dalam penelitian ini, yang mencakup beberapa baris data untuk memberikan gambaran tentang struktur dan atribut-atribut yang ada. *Dataset* ini akan digunakan untuk menganalisis dan mengembangkan model klasifikasi untuk diagnosis penyakit ginjal kronis, dengan harapan hasilnya dapat memberikan kontribusi positif dalam bidang medis.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

Tabel 1. *Dataset*

Id	*age*	*blood pressure*	*specific gravity*	*albumin*	*sugar*	....	*appetite*	*pedal edema*	*anemia*	*classification*
0	48.0	80.0	1.02	1.0	0.0	….	0.0	1.0	1.0	0
1	70.0	50.0	1.02	4.0	0.0	….	0.0	1.0	1.0	0
2	62.0	80.0	1.01	2.0	3.0	….	1.0	1.0	0.0	0
3	48.0	70.0	1.005	4.0	0.0	….	1.0	0.0	0.0	0
…	…..	…..	…..	…..	…..	….	…..	…..	…..	…..
396	42.0	70.0	1.025	0.0	0.0	….	0.0	1.0	1.0	1
397	12.0	80.0	1.020	0.0	0.0	….	0.0	1.0	1.0	1
398	17.0	60.0	1.025	0.0	0.0	….	0.0	1.0	1.0	1
399	58.0	80.0	1.025	0.0	0.0	….	0.0	1.0	1.0	1

Selanjutnya data yang telah didapatkan akan mengalami proses *Exploratory Data Analysis* (EDA) dan praproses untuk membersihkan data dari nilai yang hilang, duplikat, atau outlier yang dapat mengganggu analisis. Langkah ini penting untuk memastikan integritas data sebelum dilanjutkan ke tahap analisis lebih lanjut. Setelah membersihkan data, langkah selanjutnya adalah melakukan visualisasi data untuk menggambarkan pola, tren, dan hubungan antar variabel sebelum data diolah lebih lanjut. Visualisasi ini dapat berupa histogram, diagram pencar, atau plot lainnya yang membantu dalam memahami struktur data secara visual sebelum dilakukan analisis lebih lanjut.

Pengimplementasian algoritma *Support Vector Machine* (SVM) dan *Decision Tree* (DT) dilakukan setelah proses praproses data selesai. Kedua algoritma ini dipilih untuk memodelkan dan mengklasifikasikan data penyakit ginjal kronis. Setelah proses pelatihan model selesai, evaluasi dilakukan menggunakan *Confusion Matrix* untuk mengevaluasi kinerja dari kedua algoritma tersebut. *Confusion Matrix* memberikan informasi tentang seberapa baik model mampu mengklasifikasikan data dengan benar, termasuk di antaranya akurasi, presisi, *recall*, dan F1-*score*. Melalui evaluasi ini, dapat dievaluasi seberapa akurat dan efektif kedua algoritma dalam memprediksi kelas dari data penyakit ginjal kronis yang diberikan.

2.2 *Exploratory Data Analysis* (EDA) & Praproses Data

Exploratory Data Analysis (EDA) adalah teknik yang digunakan untuk mengolah data dalam rangka mencari struktur tersembunyi, membersihkan data dari *missing value*, duplikat data, *outlier* yang terdapat pada data, meningkatkan wawasan ke dalam kumpulan data, dan mencari anomali [9]. Selain itu, EDA juga digunakan untuk memvisualisasikan data menggunakan aritmatika dan grafis. Proses EDA penting dilakukan karena sebelum data diterapkan pada model, data harus dipahami masalahnya terlebih dahulu, memahami hubungan antara fitur pada data tersebut [10]. Di bawah ini merupakan langkah-langkah dari proses EDA:

a. Membersihkan data dari *missing value*, duplikat data, dan *outlier*:

Langkah pertama dalam EDA adalah membersihkan data dari gangguan yang dapat mempengaruhi analisis. Ini mencakup mengidentifikasi dan menghapus nilai yang hilang (*missing value*), menghapus duplikat data untuk memastikan setiap entri adalah unik, dan mendeteksi serta menangani *outlier* yang dapat mengganggu distribusi data dan analisis statistik.

b. Melihat *mean*, *median*, dan modus pada data:

Setelah membersihkan data, langkah selanjutnya adalah menganalisis statistik deskriptif dasar dari setiap atribut atau variabel dalam dataset. Ini meliputi melihat rata-rata (*mean*), median (*median*), dan modus (*mode*) dari setiap atribut. Informasi ini memberikan pemahaman awal tentang distribusi nilai dalam *dataset* dan memungkinkan untuk mengidentifikasi pola atau tren yang mungkin ada.

c. Membuat visualisasi pada data sebelum data diolah lebih lanjut:

Visualisasi data adalah langkah penting dalam EDA untuk membantu memahami struktur dan karakteristik data dengan lebih baik. Ini dapat dilakukan dengan menggunakan berbagai jenis plot, seperti histogram, diagram pencar (*scatter plot*), box plot, dan lainnya. Visualisasi memungkinkan untuk menemukan pola, anomali, dan hubungan antar variabel dengan cara yang lebih intuitif daripada hanya mengandalkan analisis statistik.

d. Melihat korelasi antar variabel:

Terakhir, dalam proses EDA, penting untuk mengevaluasi korelasi antar variabel dalam *dataset*. Ini dapat dilakukan dengan menggunakan matriks korelasi atau *Correlation Heatmap*. Melihat korelasi membantu dalam memahami hubungan antar variabel dan dapat memberikan wawasan tentang fitur mana yang saling berkaitan atau saling memengaruhi. Informasi ini penting dalam pemilihan fitur untuk analisis lanjutan atau pembuatan model prediktif.

2.3 *Support Vector Machine* (SVM)

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

Support Vector Machine (SVM) adalah algoritma yang sering digunakan untuk menyelesaikan kasus klasifikasi. Tujuan dari algoritma ini adalah untuk menemukan hyperplane yang mempartisi dataset secara linear menjadi dua kelas. SVM berusaha menemukan *hyperplane* yang paling optimal [11]. SVM memiliki titik atau vektor yang membantu dalam membuat *hyperplane*. SVM bekerja dengan membuat garis atau batas keputusan optimal yang dapat membagi ruang berdimensi n ke dalam kelas-kelas sehingga titik data baru dapat dengan mudah diklasifikasikan ke dalam kategori yang benar [12]. Fungsi pemisahan terbaik adalah dengan mengoptimalkan nilai tepi. Nilai tepi merupakan *hyperplane* pemisah setiap kelas dan posisi ini dapat dicapai apabila garis pemisah terletak tepat di tengah-tengah dan memisahkan kelas negatif dari kelas positif [13]. *Support Vector Machine* merupakan metode pemilihan yang membandingkan parameter standar dari sekumpulan nilai diskrit atau himpunan kandidat dan memilih salah satu dengan menghasilkan akurasi klasifikasi terbaik [14]. Di bawah ini merupakan persamaan dari SVM [15].

$$Large\ f(x) = sign(sum\ _{i}^{n} = 1 y_i\ alpha_i K(x_i, x)\ +\ b) \quad (1)$$

Rumus persamaan (1) merupakan fungsi prediksi yang digunakan dalam algoritma *Support Vector Machine* (SVM) untuk klasifikasi data. Dalam rumus ini, *f(x)* adalah fungsi prediksi yang mengembalikan tanda dari hasil perhitungan, yaitu apakah data termasuk dalam kelas +1 atau -1. Variabel *x* adalah vektor fitur input, yang mewakili data yang akan diklasifikasikan. Setiap elemen *x_i* dalam rumus tersebut adalah satu titik data dalam ruang fitur, dan *y_i* adalah label kelas dari titik data tersebut, yang dapat bernilai +1 atau -1 tergantung pada kelasnya. Parameter *alpha_i* mewakili vektor bobot yang ditentukan selama proses pelatihan model SVM. Bobot ini mencerminkan pentingnya setiap titik data dalam menentukan *hiperplane* optimal yang memisahkan dua kelas.

Fungsi kernel *K(x_i, x)* digunakan untuk menghitung jarak atau kemiripan antara dua vektor fitur, yaitu antara titik data *x_i* dan input *x*. Fungsi kernel ini memungkinkan SVM untuk bekerja dalam ruang fitur berdimensi tinggi tanpa harus secara eksplisit menghitung koordinat dalam ruang tersebut, sehingga memungkinkan pemisahan data yang tidak linear. Parameter *b* adalah bias yang digunakan untuk menyesuaikan posisi hiperplane dalam ruang fitur. Bias ini membantu dalam memastikan bahwa hiperplane memisahkan kelas-kelas dengan margin yang maksimal. Fungsi *sign* pada akhirnya menentukan tanda dari hasil penjumlahan bobot yang dikalikan dengan kernel dan bias, memberikan hasil +1 atau -1, yang menunjukkan kelas prediksi dari input *x*. Dengan menggunakan kombinasi dari vektor bobot, fungsi kernel, dan bias, SVM dapat secara efektif memisahkan data ke dalam dua kelas yang berbeda dengan margin maksimal, yang merupakan inti dari kekuatan metode ini dalam klasifikasi data yang kompleks.

2.4 *Decision Tree* (DT)

Decision Tree adalah algoritma untuk melakukan klasifikasi. Algoritma ini berguna untuk mengklasifikasikan data yang belum memiliki kelas ke dalam kelas yang sudah tersedia. *Decision Tree* berbentuk pohon yang memiliki *node* terdiri dari data *training*, dan setiap node memiliki cabang yang dinamakan *leaf node* berupa *output* dari hasil *training* tersebut [16]. Langkah-langkah pada *Decision Tree* yang pertama harus dilakukan adalah menghitung Entropy [17]. Di bawah ini merupakan persamaan *Entropy*.

$$Entropy(S) = \sum_{i=1}^{n} - pi\ .log2\ pi \quad (2)$$

Rumus persamaan (2) di mana S adalah himpunan kasus yang sedang dianalisis. *Entropy* mengukur tingkat ketidakpastian atau keacakan dalam himpunan kasus tersebut. Parameter n menunjukkan jumlah partisi dari S, yang berarti berapa banyak subset atau kategori yang ada dalam himpunan kasus S. Pi menandakan proporsi Si terhadap S, yaitu rasio jumlah kasus dalam subset atau kategori Si dibandingkan dengan total jumlah kasus dalam S. Setiap pi dikalikan dengan logaritma basis 2 dari pi sendiri, dan hasilnya dijumlahkan untuk semua partisi i dari 1 hingga n. Tanda negatif digunakan karena nilai logaritma dari bilangan antara 0 dan 1 adalah negatif, dan kita ingin entropi memiliki nilai positif. Entropi bernilai tinggi ketika proporsi kasus dalam setiap kategori hampir sama, yang berarti tingkat ketidakpastian tinggi. Sebaliknya, entropi bernilai rendah jika salah satu kategori mendominasi, menunjukkan tingkat ketidakpastian rendah. Dalam konteks *Decision Tree*, *entropy* digunakan untuk menentukan atribut mana yang paling efektif dalam memisahkan data ke dalam kelas yang berbeda, dengan memilih atribut yang memberikan pengurangan terbesar dalam *entropy* setelah pemisahan.

Setelah mengetahui nilai *Entropy*, kemudian dilakukan penghitungan nilai *Gain*. Di bawah ini merupakan persamaan *Gain*.

$$Gain(S, A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} *\ Entropy\ (Si) \quad (3)$$

Pada rumus persamaan (3) dimana S adalah himpunan kasus yang sedang dianalisis, dan A adalah fitur atau atribut yang digunakan untuk memisahkan himpunan kasus tersebut. Parameter n mewakili jumlah partisi dari atribut A, menunjukkan berapa banyak subset yang terbentuk dari S berdasarkan A. Proporsi |Si| terhadap |S| adalah ukuran subset Si yang terbentuk dari himpunan kasus S setelah dipartisi oleh atribut A, dihitung dengan membagi jumlah kasus dalam subset Si dengan total jumlah kasus dalam S. |S| menunjukkan jumlah kasus dalam himpunan kasus S secara keseluruhan. Rumus ini digunakan dalam algoritma *Decision Tree* untuk memilih fitur yang paling baik memisahkan data ke dalam kelas yang berbeda, dengan fitur yang memiliki nilai *Gain* tertinggi dipilih sebagai node karena memberikan pengurangan ketidakpastian terbesar dan memisahkan data secara lebih efektif.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

2.5 *Confusion Matrix*

Confusion Matrix adalah metode evaluasi yang digunakan untuk mengukur kinerja algoritma dalam kasus klasifikasi. *Confusion Matrix* memiliki empat metrik kinerja, yaitu [18]:

a. Akurasi, yang menunjukkan seberapa baik model memprediksi dengan benar untuk kelas positif dan negatif.
b. Presisi, yang menunjukkan seberapa akurat hasil positif dari model klasifikasi.
c. *Recall*, yang menunjukkan kemampuan model mengidentifikasi nilai positif di antara semua nilai yang benar-benar positif.
d. *F1-Score*, yang merupakan kombinasi antara presisi dan recall. Metrik ini digunakan untuk menilai kinerja model ketika terdapat ketidakseimbangan kelas [19].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F1-Score = \frac{2\ x\ Recall\ x\ Precision}{Recall\ x\ Precision} \quad (7)$$

Confusion Matrix memiliki empat kategori utama yang digunakan untuk mengevaluasi kinerja model klasifikasi [20]. Pertama, *True Positive* adalah jumlah data yang benar-benar positif dan terdeteksi sebagai positif oleh sistem. Kedua, *True Negative* adalah jumlah data yang benar-benar negatif dan terdeteksi sebagai negatif oleh sistem. Ketiga, *False Positive* adalah jumlah data yang sebenarnya negatif tetapi terdeteksi sebagai positif oleh sistem. Ini sering disebut sebagai kesalahan tipe I. Keempat, *False Negative* adalah jumlah data yang sebenarnya positif tetapi terdeteksi sebagai negatif oleh sistem, yang juga dikenal sebagai kesalahan tipe II. Keempat kategori ini sangat penting dalam mengevaluasi kinerja model karena memberikan gambaran yang lebih lengkap tentang seberapa baik model dalam mengklasifikasikan data. *True Positive* dan *True Negative* menunjukkan keberhasilan model dalam mendeteksi kelas yang benar, sementara *False Positive* dan *False Negative* menunjukkan kegagalan model dalam klasifikasi. Dengan menganalisis nilai-nilai ini, peneliti dapat menentukan metrik kinerja lainnya seperti akurasi, presisi, *recall*, dan *F1-score*, yang semuanya memberikan wawasan lebih dalam tentang kekuatan dan kelemahan model klasifikasi yang digunakan. Metrik-metrik ini membantu dalam meningkatkan dan mengoptimalkan model untuk penggunaan di masa depan.

3. HASIL DAN PEMBAHASAN

3.1 *Exploratory Data Analysis* (EDA) & Praproses Data

Dataset yang sudah didapatkan kemudian melalui proses *Exploratory Data Analysis* (EDA) dan praproses data untuk menghapus *missing value*, *outlier*, dan duplikat pada data. Proses EDA dilakukan untuk memahami distribusi dan karakteristik data secara lebih mendalam. Selama praproses, *missing value* dihapus untuk memastikan integritas data. Selain itu, *outlier* yang dapat mempengaruhi hasil analisis juga ditemukan dan dihapus, terutama pada atribut "White blood cell count".Mengidentifikasi dan menangani *outlier* adalah langkah penting karena data yang tidak wajar dapat mengganggu analisis dan model prediktif. Setelah itu, penghapusan data duplikat dilakukan untuk memastikan bahwa setiap entri data adalah unik, yang membantu dalam menjaga kualitas data.

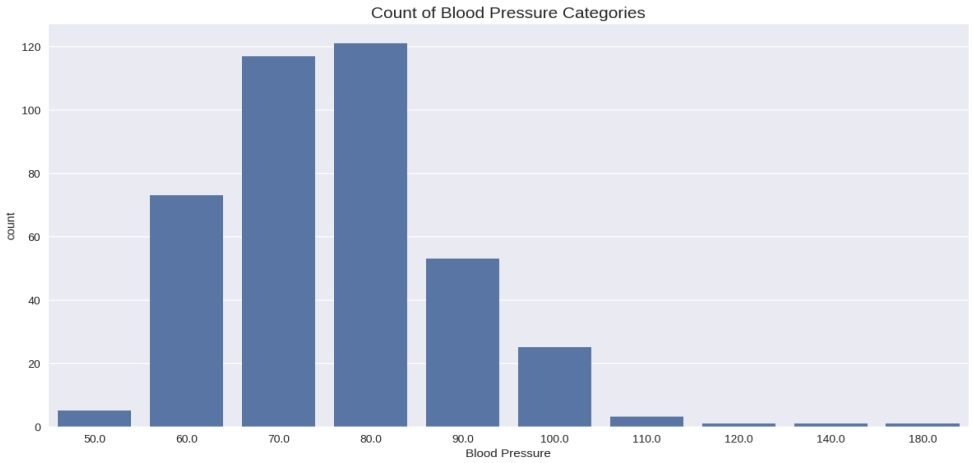
Tabel 2. *Missing value* pada atribut

Atribut	*Missing value*
Age	9
Blood pressure	12
Specific gravity	47
Albumin	446
Sugar	49
Red blood cells	152
Pus cell	65
Pus cell clumps	4
Bacteria	4
Blood glucose random	44
Blood urea	19
Serum creatinine	17
Sodium	87
Potassium	88
Hemoglobin	52
Packed cell volume	71

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

White blood cell count	106
Red blood cell count	131
Hypertension	2
Diabetes mellitus	2
Coronary artery disease	2
Appetite	1
Pedal_edema	1
Anemia	1
Classification	0

Setelah menghapus *missing value* dan *outlier*, langkah berikutnya adalah memvisualisasikan data untuk memperoleh informasi sebelum melanjutkan pengolahan data. Gambar 2 merupakan jumlah pasien pada atribut "*Blood pressure*".



Gambar 2. Visualisasi atribut "*Blood Pressure*"

Gambar 2 menunjukkan bahwa tekanan darah pasien (*blood pressure*) paling sering tercatat adalah 80, dengan jumlah pasien sebanyak 117 orang. Tekanan darah 70 tercatat pada 112 pasien, sementara tekanan darah tertinggi yang tercatat adalah 180, yang hanya dimiliki oleh satu pasien. Setelah itu, dilakukan analisis korelasi data menggunakan *Correlation Heatmap* untuk mengevaluasi hubungan antar atribut dalam *dataset*. Melalui analisis ini, dapat diketahui seberapa kuat hubungan antar atribut yang ada. Atribut *"id"* dihapus karena tidak menunjukkan korelasi yang signifikan dengan atribut lainnya. Ini adalah langkah penting untuk memastikan bahwa hanya atribut yang relevan dan berdampak signifikan yang digunakan dalam analisis lebih lanjut, sehingga dapat meningkatkan akurasi model yang akan dibangun. Setelah atribut yang tidak relevan dihapus, dilakukan proses normalisasi menggunakan *Standard Scaler*. Normalisasi adalah proses penting dalam pengolahan data karena membantu menyamakan skala data sehingga algoritma pembelajaran mesin dapat bekerja lebih efektif dan efisien.

Tabel 3. Data hasil normalisasi *Standard Scaler*

Age	blood pressure	specific gravity	albumin	….	appetite	pedal edema	anemia	classification
-0.278950	0.341304	0.442570	0.000000	….	-0.511693	0.0	0.0	-0.774597
1.090128	-1.785202	0.442570	2.232969	….	-0.511693	0.0	0.0	-0.774597
0.592282	0.341304	-1.301550	0.744323	….	1.954297	0.0	0.0	-0.774597
-0.278950	-0.509298	-2.173610	2.232969	….	1.954297	0.0	0.0	-0.774597
…….	…….	…….	…….	….	…….	…….	…….	…….
-0.652334	-0.509298	1.314631	-0.744323	….	-0.511693	0.0	0.0	1.290994
-2.519258	0.341304	0.442570	-0.744323	….	-0.511693	0.0	0.0	1.290994
-2.208104	-1.359901	1.314631	-0.744323	….	-0.511693	0.0	0.0	1.290994
0.343358	0.341304	1.314631	-0.744323	….	-0.511693	0.0	0.0	1.290994

Tabel 3 menunjukkan hasil normalisasi data menggunakan *Standard Scaler*. Proses normalisasi ini mengubah nilai atribut menjadi nilai rata-rata nol dan deviasi standar satu, yang merupakan praktek umum dalam praproses data. Hal ini bertujuan untuk menghilangkan bias yang disebabkan oleh perbedaan skala antara atribut yang berbeda. Dengan normalisasi, atribut yang awalnya memiliki rentang nilai yang berbeda dapat diselaraskan sehingga tidak ada atribut yang mendominasi atribut lainnya dalam analisis. Ini sangat penting dalam algoritma pembelajaran mesin yang sensitif terhadap skala data, seperti *Support Vector Machine* dan *Neural Networks*. Proses ini meningkatkan kinerja model dengan memastikan bahwa setiap atribut memberikan kontribusi yang seimbang terhadap hasil akhir. Dengan

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

demikian, model yang dihasilkan lebih akurat dan dapat diandalkan dalam membuat prediksi. Normalisasi menggunakan *Standard Scaler* adalah salah satu teknik yang sering digunakan karena kesederhanaannya dan efektivitasnya dalam meningkatkan performa model. Hasil normalisasi data dapat dilihat pada tabel yang telah disiapkan, memberikan gambaran yang lebih jelas tentang distribusi nilai atribut setelah normalisasi dilakukan.

3.2 Implementasi Algoritma *Support Vector Machine* (SVM) dan *Decision Tree* (DT)

Setelah melalui tahap praproses, data kemudian dibagi menjadi data uji dan data latih. Penelitian ini melakukan tiga percobaan yang berbeda dengan tiga rasio perbandingan: 90:10, 80:20, dan 70:30. Atribut "classification" digunakan sebagai kelas target. Tabel 4 menampilkan perbandingan akurasi hasil dari implementasi algoritma *Support Vector Machine* (SVM) dan *Decision Tree*.

Tabel 4. Perbandingan hasil tingkat keakuratan.

Algoritma	Data Rasio	Akurasi
Decision Tree	90:10	92.5
	80:20	91.25
	70:30	90
Support Vector Machine	90:10	97.5
	80:20	95
	70:30	95

Pada percobaan pertama dengan rasio 90:10, algoritma SVM dan *Decision Tree* menunjukkan hasil akurasi yang paling tinggi. Algoritma SVM mencatat akurasi sebesar 97.5%, sedangkan *Decision Tree* menghasilkan akurasi 92.5%. Percobaan kedua dengan rasio 80:20 menunjukkan sedikit penurunan akurasi, dengan SVM mencapai 95% dan *Decision Tree* 90%. Percobaan ketiga dengan rasio 70:30 menunjukkan hasil akurasi terendah di antara ketiga percobaan, dengan SVM mencapai 93% dan *Decision Tree* 87%. Hasil ini menunjukkan bahwa pembagian data yang lebih besar untuk pelatihan (rasio 90:10) memberikan hasil akurasi yang lebih baik dibandingkan dengan rasio yang lebih kecil. Penelitian ini mengindikasikan bahwa SVM memiliki performa yang lebih baik dibandingkan *Decision Tree* dalam mengklasifikasikan data dengan berbagai rasio pembagian

3.3 Evaluasi *Confusion Matrix*

Setelah implementasi model, langkah selanjutnya adalah evaluasi untuk mengukur kinerjanya menggunakan *Confusion Matrix*. Tabel 5 menampilkan hasil evaluasi perbandingan antara algoritma *Decision Tree* dan *Support Vector Machine*. Evaluasi ini krusial dalam menilai efektivitas kedua algoritma dalam mengklasifikasikan data, dengan fokus pada metrik seperti *True Positive, True Negative, False Positive, dan False Negative*. Dari evaluasi ini, kita dapat memahami kelebihan dan kelemahan masing-masing algoritma serta memilih model yang paling sesuai untuk kebutuhan klasifikasi data.

Evaluasi menggunakan *Confusion Matrix* memberikan wawasan mendalam tentang seberapa baik model dapat mengidentifikasi kelas yang benar dan menghindari kesalahan klasifikasi. Dengan memperhatikan matriks ini, peneliti atau praktisi dapat membuat keputusan yang lebih baik dalam memilih algoritma yang paling cocok untuk tugas klasifikasi yang spesifik. Hasil evaluasi ini tidak hanya memungkinkan untuk membandingkan kinerja dua algoritma yang berbeda, tetapi juga untuk mengevaluasi apakah performa model sesuai dengan harapan dan kebutuhan aplikasi yang bersangkutan.

Tabel 5. Perbandingan hasil presisi, *recall*, dan f1-*score*

Algoritma	Data Rasio	Presisi	*Recall*	F1-*Score*
Decision Tree	90:10	0.92	0.90	0.91
	80:20	0.91	0.90	0.90
	70:30	0.90	0.89	0.89
Support Vector Machine	90:10	0.98	0.96	0.97
	80:20	0.96	0.93	0.94
	70:30	0.96	0.94	0.95

Tabel 5. menggambarkan performa algoritma *Decision Tree* pada tiga percobaan yang berbeda. Pada percobaan pertama dengan pembagian data 90:10, nilai presisi yang dicapai adalah 0.92, *recall* sebesar 0.90, dan *f1-score* sebesar 0.91. Untuk pembagian data 80:20, hasilnya hampir serupa dengan 90:10, dengan presisi 0.91, recall dan f1-score masing-masing sebesar 0.90. Evaluasi pada pembagian data 70:30 juga menunjukkan hasil yang tidak jauh berbeda, dengan presisi 0.90, *recal*l dan *f1-score* sebesar 0.89. Di sisi lain, algoritma *Support Vector Machine* memberikan hasil tertinggi pada pembagian data 90:10, dengan presisi, *recall*, dan *f1-score* berturut-turut adalah 0.98, 0.96, dan 0.97. Namun, pada pembagian data 80:20, nilai evaluasi yang diperoleh justru paling rendah, dengan presisi 0.96, *recall* 0.93, dan *f1-score* 0.94. Hasilnya mirip pada pembagian data 70:30, di mana presisi tetap 0.96, *recall* 0.94, dan *f1-score* 0.95.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

Pada tiga percobaan dengan perbandingan algoritma tersebut, ditemukan hasil akurasi yang tinggi yang sejalan dengan presisi, *recall*, dan *f1-score*. Meskipun percobaan pertama dengan pembagian data 90:10 menunjukkan hasil tertinggi, hasil dari percobaan kedua dan ketiga tidak terlalu berbeda jauh. Keduanya menunjukkan performa yang baik dalam menangani kasus klasifikasi. Hasil penelitian menunjukkan bahwa *Decision Tree* menghasilkan akurasi sebesar 92.5, presisi 0.92, *recall* 0.90, dan *f1-score* 0.91 dengan pembagian data 90:10. Sementara itu, *Support Vector Machine* mencapai akurasi 97.5, presisi 0.98, *recall* 0.96, dan *f1-score* 0.97 dengan pembagian data yang sama. Meskipun demikian, hasil akurasi, presisi, *recall*, dan *f1-score* pada pembagian data 80:20 dan 70:30 tidak jauh berbeda dengan pembagian data 90:10, menunjukkan bahwa kedua algoritma dapat bekerja secara optimal dalam mengklasifikasikan Penyakit Ginjal Kronis. Penelitian ini memberikan peningkatan akurasi pada kedua algoritma serta memperkenalkan metode evaluasi *Confusion Matrix*, yang merupakan penyempurnaan dari penelitian sebelumnya.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma *Decision Tree* dan *Support Vector Machine* mampu mengklasifikasikan Penyakit Ginjal Kronis dengan menghasilkan akurasi, presisi, *recall*, dan *f1-score* yang tinggi, sehingga meningkatkan hasil akurasi dan evaluasi dari penelitian sebelumnya. Pada algoritma *Decision Tree*, pembagian data 90 banding 10 menghasilkan akurasi, presisi, *recall*, dan *f1-score* yang lebih tinggi dibandingkan dengan pembagian data 80 banding 20 dan 70 banding 30, dengan nilai akurasi sebesar 92.5%, presisi 0.92, *recall* 0.90, dan *f1-score* 0.91. Sementara itu, algoritma *Support Vector Machine* dengan pembagian data 90 banding 10 juga menunjukkan hasil yang lebih tinggi dibandingkan dengan pembagian data lainnya, dengan akurasi sebesar 97.5%, presisi 0.98, *recall* 0.96, dan *f1-score* 0.97. Antara kedua algoritma tersebut, *Support Vector Machine* memperoleh hasil yang paling tinggi. Penelitian ini menunjukkan bahwa dengan penggunaan algoritma yang tepat dan pembagian data yang optimal, klasifikasi Penyakit Ginjal Kronis dapat dilakukan dengan sangat akurat. Hal ini sangat penting karena klasifikasi yang akurat akan membantu dalam diagnosis dan perawatan penyakit secara lebih efektif. Diharapkan penelitian ini dapat dilanjutkan dengan mengimplementasikan algoritma lain sebagai pembanding dan diintegrasikan ke dalam aplikasi yang mampu mengklasifikasikan Penyakit Ginjal Kronis. Dengan memperluas jenis algoritma yang digunakan, peneliti dapat menemukan metode yang mungkin lebih efisien atau akurat dalam menangani data medis. Selain itu, integrasi ke dalam aplikasi praktis akan membuat teknologi ini lebih mudah diakses oleh praktisi medis dan pasien.Saran untuk penelitian berikutnya adalah untuk menjelajahi metode evaluasi alternatif guna mencapai hasil yang lebih tepat dan perbaikan dalam pengolahan data. Metode evaluasi yang lebih komprehensif dapat memberikan gambaran yang lebih jelas tentang kinerja algoritma dalam situasi yang berbeda. Selain itu, perbaikan dalam pengolahan data, seperti penanganan data yang hilang atau tidak konsisten, dapat meningkatkan akurasi dan reliabilitas model yang digunakan. Penelitian ini membuka jalan bagi pengembangan teknologi medis yang lebih canggih dan efektif di masa depan.

REFERENCES

[1] A. K. Hermawan and A. Nugroho, “Analisa Data Mining Untuk Prediksi Penyakit Ginjal Kronik Dengan Algoritma Regresi Linier,” Bulletin of Information Technology (BIT), vol. 4, no. 1, pp. 37–48, 2023, doi: 10.47065/bit.v3i1.

[2] R. M. P. H. H. F. D. A. T. S. Andi Kartini Eka Yanti, “Karakteristik Pasien Penyakit Ginjal Kronis di Rumah Sakit IbnuSina Makassar Tahun 2019-2021,” Rumah Sakit Pendidikan Ibnu Sina Makassar, vol. 3, no. 2, Dec. 2022.

[3] Riset Kesehatan Dasar (Riskesdas 2018) Laporan Nasional 2018. Jakarta: Badan Penelitian dan Pengembangan Kesehatan Departemen Kesehatan Republik Indonesia, 2018.

[4] A. Ariani, K. Kunci-Penyakit, and G. Kronis, Klasifikasi Penyakit Ginjal Kronis menggunakan K-Nearest Neighbor, vol. 5, no. 1. 2019. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Dise

[5] S. Y. I. S. Imaniar Ikko Mulya Rizky, “Perbandingan Kinerja Algoritma Naive Bayes, Support Vector Machine dan Random forest untuk Prediksi Penyakit Ginjal Kronis,” Institut Informatika dan Bisnis Darmajaya, Aug. 2023.

[6] M. Rizal, M. Zakhy Syahaf, S. Rully Priyambodo, Y. Ramdhani, and U. Adhirajasa Reswara Sanjaya, “OPTIMASI ALGORITMA NAÏVE BAYES MENGGUNAKAN FORWARD SELECTION UNTUK KLASIFIKASI PENYAKIT GINJAL KRONIS,” vol. 05, 2023.

[7] T. Asra, A. Setiadi, M. Safudin, E. W. Lestari, N. Hardi, and D. P. Alamsyah, “Implementation of AdaBoost Algorithm in Prediction of Chronic Kidney Disease,” in 2021 7th International Conference on Engineering, Applied Sciences and Technology, ICEAST 2021 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 264–268. doi: 10.1109/ICEAST52143.2021.9426291.

[8] L. G. A. I. M. W. I. G. N. A. C. P. C. R. A. P. I. D. M. B. A. D. I Gede Aditya Mahardika Pratama, “Diagnosis Penyakit Ginjal Kronis dengan Algoritma C4.5, K-Meansdan BPSO,” Jurnal-Elektronik-Ilmu-Komputer-Udayana, vol. 10, no. 4, May 2022.

[9] R. Indrakumari, T. Poongodi, and S. R. Jena, “Heart Disease Prediction using Exploratory Data Analysis,” in Procedia Computer Science, Elsevier B.V., 2020, pp. 130–139. doi: 10.1016/j.procs.2020.06.017.

[10] M. Z. Siambaton and A. M. Husein, “Menganalisis Data Kesehatan Global : Pendekatan Analisis Data Eksplorasi Visual,” Data Sciences Indonesia (DSI), vol. 1, no. 2, pp. 41–49, Jan. 2022, doi: 10.47709/dsi.v1i2.1315.

[11] C. Chazar and B. E. Widhiaputra, “Machine Learning Diagnosis Kanker Payudara Menggunakan Algoritma Support Vector Machine,” INFORMASI (Jurnal Informatika dan Sistem Informasi), vol. 12, no. 1, 2020.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 74−82
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5206

[12] Sudianto, "Analisis Kinerja Algoritma Machine Learning Untuk Klasifikasi Emosi," Building of Informatics, Technology and Science (BITS), vol. 4, no. 2, Sep. 2022, doi: 10.47065/bits.v4i2.2261.
[13] A. Handayanto, K. Latifa, N. D. Saputro, and R. R. Waliyansyah, "Analisis dan Penerapan Algoritma Support Vector Machine (SVM) dalam Data Mining untuk Menunjang Strategi Promosi," JUITA: Jurnal Informatika, vol. 7, no. 2, pp. 71–79, 2019.
[14] Suhardjono, G. Wijaya, and A. Hamid, "PREDIKSI WAKTU KELULUSAN MAHASISWA MENGGUNAKAN SVM BERBASIS PSO," Bianglala Informatika, vol. 7, no. 2, 2019.
[15] N. Basuni and Amril Mutoi Siregar, "Comparison of the Accuracy of Drug User Classification Models Using Machine Learning Methods," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 7, no. 6, pp. 1348–1353, Dec. 2023, doi: 10.29207/resti.v7i6.5401..
[16] L. Qadrini, A. Seppewali, and Aina Asar, "DECISION TREE DAN ADABOOST PADA KLASIFIKASI PENERIMA PROGRAM BANTUAN SOSIAL," Jurnal Inovasi Penelitian, vol. 2, no. 7, 2021.
[17] M. Maulidah et al., "Algoritma Klasifikasi Decision Tree Untuk Rekomendasi Buku Berdasarkan Kategori Buku," vol. 13, no. 2, pp. 89–96, 2020, [Online]. Available: http://journal.stekom.ac.id/index.php/E-Bisnis-page89
[18] R. Romindo, O. P. Barus, J. J. Pangaribuan, Y. A. Pratama, and E. Wiliem, "Implementasi Algoritma Support Vector Machine Terhadap Klasifikasi Pose Balet," Building of Informatics, Technology and Science (BITS), vol. 4, no. 3, Dec. 2022, doi: 10.47065/bits.v4i3.2647.
[19] Kiki Wahyuddin, Deden Wahiddin, and Dwi Sulistya Kusumaningrum, "Sistem Deteksi Wajah Keamanan Pintu Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Arduino," Scientific Student Journal for Information, Technology and Science, Jan. 2023.
[20] B. P. Pratiwi, A. S. Handayani, and Sarjana, "Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi WSN Menggunakan Confusion Matrix," JURNAL INFORMATIKA UPGRIS, vol. 6, no. 2, 2020.