

Optimization of Decision Tree Algorithm for Chronic Kidney Disease Classification Based on Particle Swarm Optimization (PSO)

Laili Aulia Fitri^{1*}, Anna Baita^{2*}

* Informatika, Universitas Amikom Yogyakarta

lailiauliafitri@students.amikom.ac.id¹, anna@amikom.ac.id²

Article Info

Article history:

Received 2024-11-21

Revised 2024-12-02

Accepted 2024-12-04

Keyword:

Classification,
Data Mining,
Decision Tree,
Kidney Disease,
Particle Swarm Optimization.

ABSTRACT

The body's most important vital organ is the kidney. The kidneys are responsible for maintaining acid and alkaline balance, regulating blood pressure, and filtering blood to prevent the accumulation of metabolic waste in the body. However, chronic kidney disease does not always show symptoms and signs but can progress to kidney failure. Algorithm-based predictive methods in data processing show great potential in the health field to predict various diseases, one of which is kidney disease. One of the techniques in data mining is classification. One of the classification algorithms in data mining that is often used to detect diseases is *Decision Tree*. In this study, it is expected that by combining these two methods, it will make a new contribution to the *Decision Tree* algorithm that is optimized with *Particle Swarm Optimization (PSO)* for the selection of relevant features, and improve the weaknesses in the model to improve more accurate predictions. By performing feature selection with the *Particle Swarm Optimization (PSO)* algorithm, it is shown that the use of *Particle Swarm Optimization (PSO)* can improve the accuracy and performance of the *Decision Tree* algorithm in the chronic kidney disease classification process. The accuracy of the *Decision Tree* algorithm with feature selection using *Particle Swarm Optimization (PSO)* is higher, reaching 0.967%, compared to the accuracy of *Decision Tree* without *Particle Swarm Optimization (PSO)* feature selection which is only 0.95%. This shows that *Particle Swarm Optimization (PSO)* is effective in selecting relevant features so that it can significantly improve model performance.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Kesehatan merupakan hal yang sangat penting bagi manusia dimana kondisi tubuh seseorang baik secara pikiran, jiwa dan raga, yang berfungsi secara optimal. Namun, pola hidup yang tidak teratur seperti stress berkepanjangan, bertambahnya usia, maupun tekanan pekerjaan dapat memicu banyaknya berbagai macam penyakit. Salah satu organ vital tubuh yang berperan paling penting adalah ginjal [1]. Ginjal berperan untuk menjaga keseimbangan asam dan basa, mengatur tekanan darah, menyaring darah dengan menjaga kandungan darah untuk mencegah penumpukan sisa metabolisme pada tubuh [2]. Menurut *National Library of Medicine* umumnya penyakit ginjal kronis (PGK) merupakan masalah kesehatan global yang diakui sebagai penyebab

utama kematian dari abad ke- 21 [3]. Kelainan pada ginjal dapat ditandai dengan menurunnya kemampuan ginjal dalam menyaring darah [4]. Salah satu faktor yang mempengaruhi keterlambatan kesadaran penderita adalah kurangnya pengetahuan dalam mengenai bahayanya penyakit ginjal [5]. Namun, Penyakit ginjal kronis tidak selalu menunjukkan akan gejala dan tanda tetapi dapat berkembang menjadi gagal ginjal [6]. Metode prediktif berbasis algoritma dalam pengolahan data menunjukkan potensi besar dalam bidang kesehatan untuk memprediksi berbagai penyakit, salah satunya seperti penyakit ginjal [7].

Data mining adalah metodologi statistik untuk menemukan pola dalam menyelesaikan masalah dengan proses menggali informasi dengan menganalisis data besar. Dalam ahli bidang kedokteran, data mining dapat digunakan untuk membantu

pengambilan keputusan pada klinis dalam memprediksi penyakit [8]. Salah satu teknik dalam data mining adalah klasifikasi [9]. Klasifikasi merupakan proses pencarian kelompok dengan karakteristik yang sama ke dalam beberapa kelas menggunakan dataset [10]. Dalam proses klasifikasi, terdapat dua kumpulan data yang digunakan, yaitu data latihan (*training*), kumpulan data yang diberi label digunakan untuk membangun model dan data uji (*testing*), kumpulan data yang tidak diberi label digunakan untuk memprediksi label dan menentukan tingkat akurasi pada model. Secara umum, dataset latihan lebih besar daripada dataset uji, rasio dataset latihan yang sering digunakan berkisar antara 70% hingga 90% [11]. Metode klasifikasi pada data mining dapat digunakan dalam menganalisis penyakit, salah satunya seperti pada ginjal kronis [8]. Salah satu algoritma klasifikasi pada data mining yang sering digunakan untuk mendeteksi penyakit adalah *Decision Tree* [12].

Algoritma *Decision Tree* (DT) adalah salah satu metode pengklasifikasian yang populer menggunakan contoh pohon yang menghasilkan model yang mudah diinterpretasikan [13]. Namun, kelemahan utama yang dimiliki algoritma ini adalah rentan terhadap overfitting, terutama ketika diterapkan pada dataset yang memiliki atribut atau ukuran dataset yang tidak seimbang [14]. *Particle Swarm Optimization* (PSO) adalah salah satu cara untuk mengatasi kelemahan ini, dengan melakukan optimasi pemilihan fitur yang relevan untuk membuat model *Decision Tree* (DT) [15].

Terdapat penelitian sebelumnya yang melakukan klasifikasi terhadap Penyakit Ginjal yang menggunakan dataset Chronic Kidney Disease (CKD) University of California Irvine (UCI) [16]. Menurut pratama dkk [10], menggunakan algoritma C4.5, K-means dan BPSO yang diterapkan untuk pemilihan fitur, menghasilkan peningkatan kinerja mencapai akurasi 96,875%. Sebagai perbandingan model tanpa BPSO hanya mencapai akurasi 88,125%. Kemudian, menurut Naufal Rifqi dkk [17], menggunakan algoritma Random Forest dengan Particle Swarm Optimization (PSO) mencapai akurasi 98,%. Sebagai perbandingan model tanpa Particle Swarm Optimization (PSO) hanya mencapai akurasi 0,867% dalam mengklasifikasikan pasien Ckd dan non-CKD. Selanjutnya pada penelitian Madani dkk [18], membandingkan dua algoritma, yaitu algoritma Support Vector Machine (SVM) dan algoritma Decision Tree (DT), untuk membantu tenaga medis dalam upaya pencegahan CKD. Support Vector Machine (SVM) terbukti memiliki performa yang lebih baik dengan akurasi mencapai nilai 97,5%, dibandingkan Decision Tree (DT) dengan akurasi dengan nilai 92,5%. Sedangkan Widiati dkk [19], menggunakan pendekatan hibrida yang menggabungkan Decision Tree (DT) dengan Particle Swarm Optimization (PSO) untuk deteksi dini Penyakit Ginjal Kronis (CKD). meningkatkan akurasi dari 98,25% menjadi 98,50% dan AUC dari 0,931 menjadi 0,984, menunjukkan kemampuan yang lebih baik dalam mendeteksi kasus CKD. Presisi juga mengalami peningkatan dari 98,04% menjadi 98,71%, meskipun nilai recall tetap pada 97,33%.

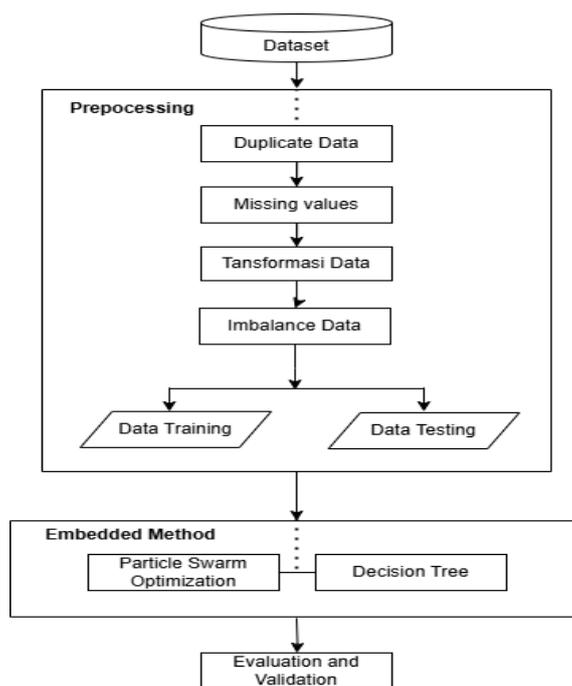
Secara keseluruhan, pendekatan ini lebih unggul dibandingkan hanya menggunakan DT untuk deteksi dini CKD.

Berdasarkan uraian penelitian sebelumnya menunjukkan berbagai penelitian pada algoritma *Decision Tree* yang berbasis *Particle Swarm Optimization* (PSO), kombinasi untuk kedua metode ini dalam satu pendekatan pada prediksi penyakit ginjal masih jarang ditemukan pada penelitian lainnya dan pasti dapat menjadi potensi untuk dikembangkan pada penelitian ini. Oleh karena itu, Pada penelitian ini diharapkan dengan mengkombinasikan kedua metode ini memberikan kontribusi baru pada algoritma *Decision Tree* yang dioptimalkan dengan *Particle Swarm Optimization* (PSO) untuk pemilihan fitur yang relevan, serta memperbaiki kelemahan pada model untuk meningkatkan prediksi yang lebih akurat pada dataset penyakit ginjal dalam penelitian ini.

II. METODE PENELITIAN

A. Tahap Penelitian

Dalam penelitian ini dilakukan beberapa tahapan penelitian, adapun tahapan penelitian yang dilakukan pada gambar 1, menampilkan diagram alur penelitian.



Gambar 1. Alur Penelitian

B. Dataset

Pada penelitian ini dataset yang digunakan didapatkan dari website *University of California Irvine* (UCI) [16], yang memiliki 25 atribut dengan total data keseluruhan 400 data. Dataset ini masih tergolong ke dalam jumlah dataset yang kecil.

TABEL I
KETERANGAN DATASET

Attribut	Deskripsi	Nilai
Age(numerical)	Umur	Umur dalam tahun
Blood Pressure(numerical) bp	Ukuran tekanan yang dihasilkan oleh darah	mm/Hg
Specific Gravity(nominal) sg	Semakin besar nilainya semakin bahaya	Dari 1.005 hingga 1.025
Albumin(nominal)al	Protein plasma darah	Dari 0 hingga 5
Sugar(nominal) su	Kadar gula darah	Dari 0 hingga 5
Red Blood Cells (nominal) rbc	Pengukuran sel darah merah	Normal,abnormal
Pus Cell (pc)	Pengukuran sel nanah dalam urin	Normal,abnormal
Pus Cell clumps (pcc)	Pengukuran gumpalan sel nanah dalam urin	Present,notpresent
Bacteria (ba)	Pengukuran bakteri yang terdapat dalam tubuh	Present,notpresent
Blood Glucose Random (bgr)	Pengukuran kadar glukosa	mgs/dl
Blood Urea (bu)	Pengukuran kadar urea dalam darah	mgs/dl
Serum Creatinine (sc)	Pengukuran kreatinin pada darah	mgs/dl
Potassium (pot)	Pengukuran natrium pada darah	mEq/L
Hemoglobin (hemo)	Pengukuran kalium pada darah	gms
Packed Cell Volume (pcv)	Menghitung jumlah hemoglobin	numeric
White Blood Cell Count (wc)	Parameter pengukuran sel darah	Millions/cmm
Red Blood Cell Count (rc)	Menghitung jumlah total sel darah merah	Millions/cmm
Hypertension (htn)	Memiliki masalah hipertensi	Yes,no
Diabetes Mellitus (dm)	Memiliki masalah diabetes	Yes,no
Coronary Artery Disease (cad)	Coronary Artery Disease	Yes,no
Appetite (appet)	Memiliki masalah nafsu makan	Good,poor
Pedal Edema (pe)	Pengukuran cairan pada betis	Yes,no
Anemia (ane)	Pengukuran darah merah	Yes,no
Class (class)	Target Class	Ckd, notckd

C. Preprocessing Data

Preprocessing data merupakan salah satu metode yang sangat penting dalam *machine learning* yang melibatkan sebuah langkah utama dalam mengimplementasikan data preprocessing yang dapat mempengaruhi analisis, tahap pertama yang dilakukan adalah *data cleaning* dengan mengatasi *missing value*, mengecek duplikat pada data, transformasi data untuk mengubah data yang bersifat kategori menjadi format data numerik, *splitting data* dengan membagi data menjadi dua bagian, yaitu *data testing* dan *data training*. Data training digunakan untuk melatih model sehingga model dapat memahami pola pada data, dan data pengujian digunakan untuk melakukan validasi latihan pada model [20].

D. Handle Imbalance

Handle imbalance merupakan terjadinya ketidakseimbangan pada jumlah sampel pada setiap kelas yang akan diprediksi pada klasifikasi dalam *machine learning* [21]. Pada penelitian ini menggunakan teknik *SMOTE* (*Synthetic Minority Over-sampling Technique*) adalah proses untuk menangani kelas yang tidak seimbang, untuk menghasilkan sebuah metode baru pada sampel kelas yang minoritas, yang bertujuan untuk membantu meningkatkan representasi dan kontribusi kelas pada klasifikasi dalam *machine learning*, sehingga dapat menghasilkan model yang seimbang pada kelas dalam dataset [22]. Cara perhitungan menggunakan rumus *Euclidean distance* menggunakan

persamaan (1).

$$dist = \sqrt{(x_1 + y_1)^2 + (x_2 + y_2)^2 + \dots + (x_n + y_n)^2} \quad (1)$$

Ket:

X : Sebagai minoritas

N : Persentase jumlah instance yang akan dibuat

x : Jumlah instance terdekat dari instance yang dicari

Setelah menghitung jarak geometris instance, data replikasi dibuat dari instance terdekat pada persamaan (2).

$$x_{syn} = x_i + (x_{knn} - x_i)\sigma \quad (2)$$

Ket:

x_{syn} : Sintesis yang akan diciptakan

x_i : Data yang akan direplikasi

x_{knn} : Data yang memiliki jarak terdekat dari data yang akan direplikasi

σ : Nilai random antara 0 dan 1

Pada teknik *SMOTE* (*Synthetic Minority Over-sampling Technique*), Langkah pertama yaitu menghitung jarak antara setiap sampel pada kelas minoritas. Setelah perhitungan pada jarak, satu sampel minoritas dipilih untuk titik awal, dan k sebagai tetangga terdekat dari s.

E. Embedded Method

Dalam penelitian ini model *Decision Tree* yang diterapkan tanpa melakukan proses *hyperparameter tuning*. Model pada *Decision Tree* sepenuhnya menggunakan nilai *default* yang telah ditentukan oleh *scikit learn*, maka tidak ada penyesuaian atau optimasi parameter yang dilakukan. Pada tahap ini, pemilihan metode *machine learning* merupakan langkah yang sangat penting dalam pengembangan pemodelan yang efektif, diantaranya:

1) Decision Tree

Algoritma *decision tree* merupakan salah satu teknik pada algoritma di dalam *machine learning* yang digunakan untuk membentuk sebuah pohon keputusan. Setiap node pada pohon menunjukkan fitur yang akan diklasifikasikan, dan setiap simpul daun menunjukkan label pada kelas [21]. Salah satu algoritma *decision tree* yang umumnya digunakan adalah algoritma *Iterative Dichotomiser 3 (ID3)*.

• Entropy

Entropy adalah ukuran seberapa acak suatu kelompok pada data, pembagian dilakukan dari atas ke bawah secara bertingkat. Sementara itu, untuk mencari perhitungan nilai entropy menggunakan persamaan (3):

$$Entropy(S) = \sum_{i=1}^c -P_i * \log_2 P_i \quad (3)$$

Keterangan:

S	: Himpunan Kasus
A	: Atribut/Fitur
N	: Jumlah Partisi S
Pi	: Proporsi dari Si terhadap S

• Information Gain

Gain adalah informasi yang diperoleh dari perubahan *entropy* pada sekumpulan data besar. *Entropy* memiliki karakteristik yang menjadikannya sebagai porosnya. Algoritma ini secara ilmiah memperbesar atribut cabang pohon sehingga seluruh pohon terbentuk, karena dalam seleksi atribut, sebagian dari atribut yang ada tidak digunakan. Proses ini menggunakan *information gain* yang memberikan nilai bobot pada setiap atribut dalam pohon keputusan, sehingga proses *machine learning* pada pohon keputusan tidak akan memasukkan semua atribut yang tersedia pada dataset [22]. Nilai *gain* tertinggi dari atribut digunakan sebagai penentu atribut yang akan dijadikan akar. Perhitungan nilai gain menggunakan persamaan (4):

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^c \frac{|S_i|}{|S|} * Entropy(S) \quad (4)$$

Keterangan:

S	: Himpunan Kasus
A	: Atribut/Fitur
n	: Jumlah Partisi atribut A
Si	: Jumlah kasus pada partisi ke-i
S	: Jumlah kasus dalam S

2) Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) adalah suatu algoritma *heuristic* yang termasuk dalam kelompok metode optimasi [23]. Pada penelitian ini, digunakan modifikasi dari *Particle Swarm Optimization (PSO)* yang dikenal sebagai *Binary Particle Swarm Optimization (BPSO)* yang dirancang untuk seleksi fitur dengan representasi biner, di mana setiap atribut yang direpresentasikan sebagai *false* atau 0 (tidak dipilih) dan *true* atau 1 (terpilih).

Tahapan dalam seleksi fitur menggunakan *Particle Swarm Optimization (PSO)* adalah sebagai berikut:

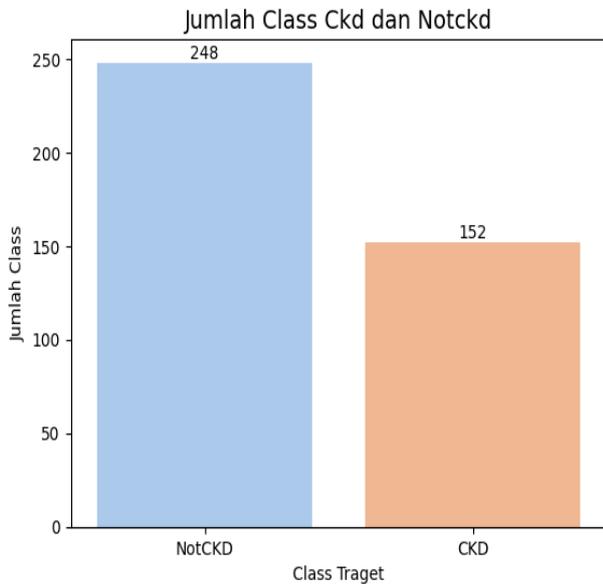
- 1) Menginisialisasi parameter yang dibutuhkan seperti jumlah partikel pada populasi, nilai batas iterasi, cognitive learning (c1), social learning (c2) dan inertia weight (w).
- 2) Menginisialisasi posisi dan kecepatan awal dari seluruh partikel. Setiap partikel yang memiliki jumlah fitur (gen) sebanyak jumlah fitur pada dataset. Pembentukan kecepatan awal partikel dibuat dengan nilai *default* 0, sehingga pada pembentukan posisi awal partikel dibuat secara random dengan nilai 0 atau 1. Inisialisasi partikel digunakan untuk pemilihan pada fitur yang digunakan pada klasifikasi *Decision Tree*.
- 3) Menghitung nilai fitness pada setiap partikel dengan cara melakukan klasifikasi pada model *Decision Tree*. Dimana fitur dengan partikel yang bernilai 1 akan terpilih yang digunakan untuk klasifikasi dan nilai partikel 0 tidak akan dipilih dan digunakan dalam klasifikasi.
- 4) Selanjutnya menentukan Pbest dan Gbest. Pada awal iterasi, Pbest akan disamakan dengan nilai posisi awal partikel. Sedangkan pada iterasi berikutnya, Pbest ditentukan dengan cara melihat nilai fitness yang tertinggi dari posisi partikel di setiap iterasi. Gbest ditentukan dengan memilih partikel pada Pbest dengan nilai fitness tertinggi.
- 5) Berikutnya yaitu pengecekan kondisi selesai berdasar jumlah iterasi yang sudah dilewati. Jika jumlah perulangan belum sampai batas maksimum iterasi, maka akan diteruskan ke tahapan update kecepatan dan penentuan posisi baru. Tetapi, apabila jumlah perulangan telah mencapai maksimum iterasi, maka telah diperoleh hasil seleksi fitur paling baik dari nilai Gbest di iterasi paling akhir.
- 6) Pembaruan kecepatan untuk menetapkan arah kemana partikel akan melakukan perpindahan dan juga untuk membenarkan posisi awal. Persamaan (1) merupakan rumus untuk update kecepatan pada algoritma BPSO.

$$v_{id}^{new} = w * v_{id}^{old} + c_1 r_1 (p_{id}^{old} - x_{id}^{old}) + c_2 r_2 (g_{id}^{old} - x_{id}^{old}) \quad (5)$$

Keterangan:

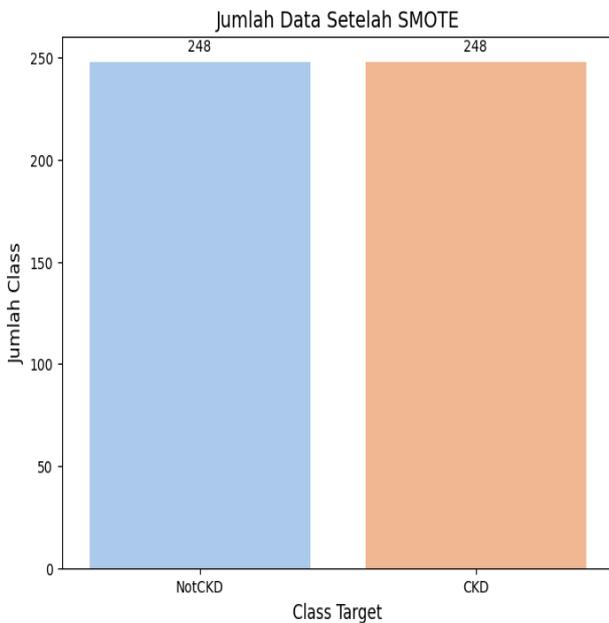
Vi,m	: Kecepatan partikel ke-i pada iterasi ke-i
w	: Faktor bobot inersia
c1, c2	: Konstanta akselerasi (learning rate)
R	: Bilangan random (0-1)
Xi,d	: Posisi terbaik sebelumnya dari partikel ke-I pada iterasi ke-i
Pbesti	: Posisi terbaik sebelumnya dari partikel ke-i

Over-sampling Technique), menambahkan sampel baru pada kelas target dengan jumlah yang sedikit untuk menyeimbangkan sampel dengan kelas target dengan jumlah yang lebih banyak. Dengan menerapkan teknik *SMOTE*, jumlah data pada Class disamakan, sehingga kedua kelas seimbang yaitu dengan jumlah data 248. Data perbandingan pada class sebelum dan setelah dapat dilihat pada gambar 3 dan gambar 4.



Gambar 3. Sebelum SMOTE

Setelah menerapkan teknik *SMOTE* maka jumlah data pada kategori class *Ckd* dan *Notckd* menjadi seimbang.



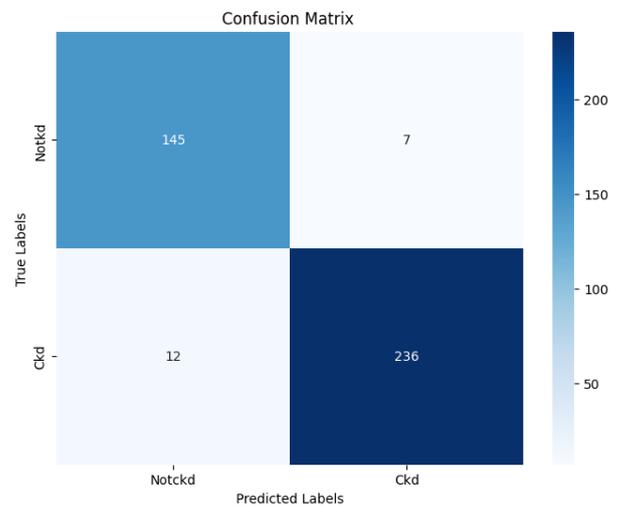
Gambar 4. Setelah SMOTE

C. *Embedded Method*

Setelah dilakukan tahap *cleaning data*, selanjutnya pada penelitian ini menggunakan algoritma *Decision Tree* yang dioptimalkan dengan *Particle Swarm Optimization (PSO)*, yang bertujuan untuk meningkatkan performa model melalui seleksi fitur sehingga hanya fitur-fitur yang relevan digunakan dalam proses pelatihan yang diharapkan dapat meningkatkan performa pada algoritma *decision tree*. Setelah data class seimbang, langkah berikutnya yaitu *splitting data*, yang dibagi menjadi dua *subset*, yaitu dengan data latih (*training*) dan data uji (*testing*). Rasio perbandingan data pada penelitian ini yaitu 80:20%, dimana 80% dari total keseluruhan data digunakan untuk melatih model dan 20% digunakan untuk menguji performa pada model yang telah dilatih. Sehingga total dataset yang terdiri dari 400 data, data yang dipilih secara acak untuk proses pelatihan dengan rasio 80% yaitu sebesar 320 data, sementara sisanya 20% sebesar 80 data, yang digunakan untuk menguji model yang telah dilatih. Tahapan ini merupakan syarat penting sebelum melanjutkan ke dalam proses analisis.

1) Hasil Pemodelan Algoritma *Decision Tree*

Pada tahap pertama penerapan klasifikasi pada algoritma *decision tree*, dengan pembagian data 20% untuk data testing. Pengujian model dilakukan dengan menggunakan *confusion matrix* yang digunakan untuk mengevaluasi kinerja pada model yang akan menghasilkan beberapa matrik evaluasi utama, seperti nilai akurasi, *precision*, *recall*, dan *f1-score*, yang memberikan gambaran mengenai seberapa baik model dalam memprediksi setiap kelas. Gambar 5 menampilkan hasil *confusion matrix* pada algoritma *decision tree*.



Gambar 5. Confusion Matrix pada Algoritma Decision Tree

Gambar 5, menampilkan *confusion matrix* hasil klasifikasi model yang terdiri dari dua kelas, yaitu Penderita penyakit ginjal Ckd dan yang bukan Penderita penyakit ginjal Notckd. Model ini memiliki akurasi sebesar 95,25%. Model berhasil mengklasifikasikan 236 sampel dengan label *Ckd* dan 145 sampel dengan label *Notckd* secara benar. Namun, terdapat 7

kesalahan pada kelas *ckd* dan 12 kesalahan pada kelas *Notckd*. Berikut pada tabel 2, ditampilkan performa model dalam mengklasifikasikan data.

TABEL 2
HASIL AKURASI PADA DECISION TREE

Pengujian pada model	Nilai			
	Precision	Recall	F1-score	Accuracy
Decision Tree	95.26%	95.25%	95.38%	95.25%

Tabel 2, dapat disimpulkan hasil evaluasi model *Decision Tree* dalam mengklasifikasikan penderita penyakit ginjal (*Ckd*) dan yang bukan penderita penyakit ginjal (*Notckd*). Akurasi yang diperoleh pada model *decision tree* dengan nilai precision sebesar 95,26%, recall 95,25%, F1-score 95,38%, dan accuracy 95,25%.

2) Pemilihan fitur dengan *Particle Swarm Optimization (PSO)* pada *Decision Tree*

Pemilihan fitur menggunakan metode *Particle Swarm Optimization (PSO)* digunakan untuk pemilihan fitur 184imana nilai biner 1 atau *true* yang menunjukkan fitur terpilih atau 0 atau *false* menunjukkan fitur tidak terpilih, yang dilakukan menggunakan parameter yang bertujuan untuk mencapai keseimbangan optimal antara eksplorasi ruang solusi yang terbaik. Tabel 3, merupakan parameter pada *psa*.

TABEL 3
PARAMETER PADA PSO

Parameter PSO	Nilai parameter
Koefisien kognitif (<i>c1</i>)	0,2
Koefisien sosial (<i>c2</i>).	1,0
Solusi global (<i>w</i>)	0,9
Jumlah iterasi (<i>k</i>)	25
jumlah partikel (<i>p</i>)	35

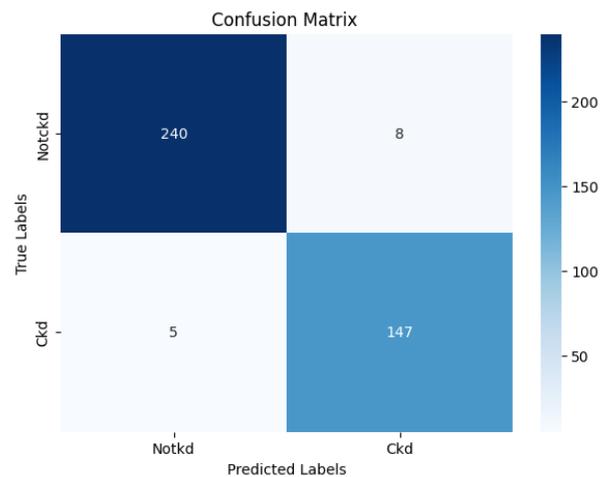
Nilai konstanta *c1* (koefisien pembelajaran untuk kecepatan individu), dengan nilai 0.2 yang berfungsi untuk mendorong partikel mengeksplorasi solusi berdasarkan pengalaman individu, dan *c2* dengan nilai 1.0, yang berfungsi mengarahkan partikel mengikuti solusi terbaik dari kelompok, yang memastikan partikel tidak hanya fokus pada satu titik tetapi juga mempertimbangkan pada solusi global. Serta, nilai bobot inersia konstanta *w* dengan nilai 0.9, berfungsi untuk menjaga keseimbangan kecepatan partikel, yang membantu partikel bergerak untuk menjelajahi ruang solusi, tetapi perlahan melambat saat mendekati solusi optimal. Bobot nilai yang terlalu kecil (<0,3) membuat partikel terlalu cepat konvergen, sementara bobot yang terlalu besar (>1) menyebabkan partikel sulit fokus ke solusi terbaik. Dan jumlah iterasi sebanyak 25 (*k=25*), iterasi yang lebih dari 35 akan meningkatkan beban pada komputasi. Serta ukuran populasi yang cukup besar, yaitu 35 partikel (*p=35*), Partikel dengan nilai yang sedikit (<25) mengurangi kemampuan eksplorasi, sedangkan partikel yang terlalu banyak dengan

nilai (>35) hanya meningkatkan waktu komputasi. Hasil dari pemilihan fitur ini dapat dilihat pada Tabel 4, yang menunjukkan fitur-fitur yang terpilih setelah proses optimasi dengan *Particle Swarm Optimization (PSO)*.

TABEL 4
HASIL PEMILIHAN FITUR PADA PSO

Fitur yang terpilih	Fitur yang tidak terpilih
Umur	gula
Tekanan darah	Albumin
Berat badan	Sel nanah
sel darah merah	Gumpalan sel nanah
bakteri	Glukosa darah acak
urea darah	Serum kreatinin
kalium	potassium
Volume sel yang dikemas	kalium
Jumlah sel darah merah	hemoglobin
hipertensi	Diabetes mellitus
nafsu makan	Penyakit arteri koroner
pedal edema	Anemia

Dari hasil yang diperoleh pada tabel 4, terdapat sebanyak 12 fitur yang terpilih dari 24 fitur pada dataset. Setelah dilakukan optimasi menggunakan metode *Particle Swarm Optimization (PSO)*. Proses optimasi dilakukan dengan menggunakan nilai *c1*, *c2*, *w*, dengan 35 iterasi dan ukuran populasi partikel 25. Setelah 50 iterasi optimasi, hasil terbaik diperoleh dengan biaya optimal 0.0, yang menunjukkan pemilihan fitur yang sangat relevan. Selanjutnya dengan pembagian data 20% untuk data testing. Fitur-fitur yang terpilih oleh *PSO* pada model deteksi penyakit ginjal kronik mencakup variabel-variabel yang memiliki memiliki dasar medis yang kuat, yang meningkatkan kemampuan model untuk mengenali tanda-tanda awal kerusakan ginjal dengan faktor risiko dan gejala penyakit ginjal. Pengujian model dilakukan dengan menggunakan confusion matrix yang akan menghasilkan nilai akurasi, precision, recall, dan f1-score. gambar 6 menampilkan hasil confusion matrix pada algoritma *decision tree*.



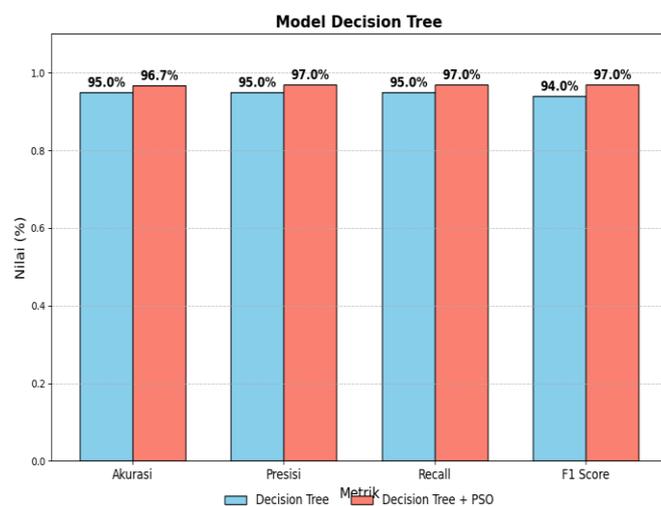
Gambar 6. Confusion Matrix pada Particle Swarm Optimization (PSO) + Decision Tree

Dari gambar 6 diatas, menampilkan confusion matrix hasil klasifikasi model yang terdiri dari dua kelas, yaitu Penderita penyakit ginjal Ckd dan yang bukan Penderita penyakit ginjal Notckd. Model ini memiliki kinerja yang baik dengan akurasi sebesar 0.967%. Model berhasil mengklasifikasikan 147 sampel dengan label Ckd dan 240 sampel dengan label Notckd secara benar. Namun, terdapat 8 kesalahan False Positives pada kelas ckd dan 5 kesalahan False Negatives pada kelas Notckd. Berikut pada tabel 5, ditampilkan performa model dalam mengklasifikasikan.

TABEL 5
HASIL AKURASI PSO + DECISION TREE

Pengujian pada model	Nilai			
	Precision	Recall	F1-score	Accuracy
Decision Tree + Pso	0.97%	0.97%	0.97%	0.967%

Tabel 5 diatas dapat disimpulkan hasil evaluasi model dengan *Particle Swarm Optimization (PSO)* pada algoritma *decision tree*, optimasi fitur memberikan peningkatan signifikan pada kinerja model. Akurasi yang diperoleh pada model *decision tree* dengan nilai precision sebesar 0.97%, recall 0.97%, F1-score 0.97%, dan accuracy 0.967%. Nilai precision menunjukkan bahwa model mampu memprediksi kasus positif (penderita penyakit ginjal) dengan tingkat kesalahan yang sangat kecil, sementara nilai recall yang tinggi memastikan sebagian besar kasus positif terdeteksi. F1-score yang hampir sempurna mencerminkan keseimbangan optimal antara precision dan recall. Dari hasil akurasi yang diperoleh pada gambar 7 menunjukkan hasil pemodelan pada algoritma *decision tree* setelah pemilihan fitur menggunakan *Particle Swarm Optimization (PSO)*.



Gambar 7. Pengujian pengaruh pada seleksi fitur menggunakan PSO

Berdasarkan pada Gambar 7, pada model *Decision Tree* dengan metode *PSO* memberikan hasil evaluasi yang lebih baik dibandingkan dengan model *Decision Tree* tanpa *PSO*.

Model *Decision Tree* tanpa *PSO* mendapatkan nilai akurasi, presisi, recall, dan *F1-score* masing-masing sebesar 95%, sedangkan pada model *Decision Tree* dengan *PSO* mencapai nilai akurasi 96,7%, presisi 97%, recall 97%, dan *F1-score* 97%. Hal ini menunjukkan bahwa penerapan seleksi fitur dengan *PSO* mampu meningkatkan performa pada model *decision tree*. Pada model ini menunjukkan kinerja yang cukup baik dalam mengklasifikasikan pasien pada penyakit ginjal kronik. Namun, ada beberapa hal yang harus diperhatikan sebelum model ini diterapkan secara nyata. Pertama, data yang digunakan pada penelitian ini adalah data sekunder yang diambil dari satu sumber dataset, dan mungkin dataset pada penelitian ini tidak sepenuhnya mewakili populasi pasien dari berbagai daerah atau rumah sakit. Oleh karena itu, model perlu diuji lebih lanjut dengan menggunakan dataset baru dari pasien dengan berbagai latar belakang yang berbeda, untuk memastikan kemampuan generalisasinya. Kedua, model ini dirancang untuk bekerja dengan data yang lengkap, sehingga tantangan seperti missing value atau tidak lengkap perlu ditangani melalui metode imputasi data atau preprocessing tambahan lainnya. Dengan pengujian tambahan, model ini memiliki potensi untuk membantu dokter dalam mendeteksi penyakit ginjal kronik.

IV. KESIMPULAN

Penelitian ini bertujuan untuk penerapan pada algoritma *Decision Tree* yang dioptimalkan dengan metode *Particle Swarm Optimization (PSO)* untuk seleksi fitur berhasil meningkatkan performa dalam mengklasifikasikan penyakit ginjal kronis Ckd dan Notckd. Setelah melalui tahap cleaning data, model *Decision Tree* pertama kali dibangun tanpa menggunakan *Particle Swarm Optimization (PSO)* untuk pemilihan fitur, menghasilkan akurasi 95%. dengan presisi sebesar 95.26% dan recall 95.25%. Meskipun akurasi model ini cukup tinggi, masih terdapat fitur-fitur yang tidak relevan. Pada penelitian ini menggunakan *decision tree* dengan optimasi pada *Particle Swarm Optimization (PSO)* untuk memilih fitur yang relevan, untuk meningkatkan akurasi dan efisiensi pada model *Decision Tree*. Proses pada *Particle Swarm Optimization (PSO)* dengan menggunakan nilai konstanta $c1 = 0,2$ dan $c2 = 1,0$ serta nilai $w = 0,9$ untuk mengatur kecepatan pergerakan partikel. Pembatasan jumlah iterasi (25 iterasi) dan populasi partikel 35. Hasil optimasi menunjukkan peningkatan signifikan dalam performa model *Decision Tree* setelah diterapkan pemilihan fitur dengan *Particle Swarm Optimization (PSO)*, akurasi model meningkat dari 95% menjadi 96,75%, yang menandakan peningkatan kemampuan model dalam mendeteksi kelas Ckd dan Notckd dengan baik. Selain itu, pemilihan fitur dengan *Particle Swarm Optimization (PSO)* membuat model lebih efisien dengan hanya mempertahankan fitur-fitur penting yang relevan.

DAFTAR PUSTAKA

- [1] F. Yanto, M. I. Hatta, I. Afrianty, and L. Afriyanti, 'Pengaruh Image Enhancement Contrast Stretching dalam Klasifikasi CT-Scan Tumor Ginjal menggunakan Deep Learning', *INOVTEK Polbeng - Seri Inform.*, vol. 9, no. 1, pp. 408–419, 2024, doi: 10.35314/isi.v9i1.4233.
- [2] F. Yanto, N. Jannata, L. Handayani, and E. P. Cynthia, 'Pengaruh Contrast Limited Adaptive Histogram Equalization dalam Klasifikasi CT-Scan Tumor Ginjal menggunakan Deep Learning', *INOVTEK Polbeng - Seri Inform.*, vol. 9, no. 1, pp. 420–433, 2024, doi: 10.35314/isi.v9i1.4235.
- [3] C. P. Kovcsdy, 'Epidemiology of chronic kidney disease: an update 2022', *Kidney Int. Suppl.*, vol. 12, no. 1, pp. 7–11, 2022, doi: 10.1016/j.kisu.2021.11.003.
- [4] A. Tarisa Akbar, N. Yudistira, and A. Ridok, 'Identifikasi Gagal Ginjal Kronis Dengan Mengimplementasikan Metode Support Vector Machine Beserta K-Nearest Neighbour (SVM-KNN)', *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, pp. 301–307, Apr. 2023, doi: 10.25126/jtiik.2023106059.
- [5] A. Fakhira, F. Insani, M. Irsyad, Y. Vitriani, and F. Kurnia, 'Diagnosis Dini Penyakit Gagal Ginjal Dengan Metode Dempster', *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 2, p. 441, 2023, doi: 10.35314/isi.v8i2.3728.
- [6] V. Kynecissia Gliselda, 'Diagnosis dan Manajemen Penyakit Ginjal Kronis (PGK)', *JMH J. Med. Utama*, Jul. 2021, [Online]. Available: <http://jurnalmedikahutama.com>
- [7] Imaniar Ikko Mulya Rizky, Suhendro Yusuf Irianto, and Sriyanto, 'Perbandingan Kinerja Algoritma Naive Bayes, Support Vector Machine dan Random forest untuk Prediksi Penyakit Ginjal Kronis', *Semin. Nas. Has. Penelit. dan Pengabd. Masy.*, pp. 139–151, Aug. 2023.
- [8] A. K. Hermawan and A. Nugroho, 'Analisa Data Mining Untuk Prediksi Penyakit Ginjal Kronik Dengan Algoritma Regresi Linier', *Bull. Inf. Technol.*, vol. 4, no. 1, pp. 37–48, 2023, doi: 10.47065/bit.v3i1.
- [9] L. Ariyanti and A. Alamsyah, 'C4.5 Algorithm Optimization and Support Vector Machine by Applying Particle Swarm Optimization for Chronic Kidney Disease Diagnosis', *Recursive J. Informatics*, vol. 1, no. 1, pp. 18–26, Mar. 2023, doi: 10.15294/rji.v1i1.65196.
- [10] I. G. A. Mahardika Pratama, L. G. Astuti, I. M. Widiartha, I. G. N. A. Cahyadi Putra, C. R. Adi Pramarta, and I. D. M. B. Atmaja Darmawan, 'Diagnosis Penyakit Ginjal Kronis dengan Algoritma C4.5, K-Means dan BPSO', *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 10, no. 4, p. 371, 2022, doi: 10.24843/jlk.2022.v10.i04.p07.
- [11] R. R. Adhitya, Wina Witanti, and Rezki Yuniarti, 'Perbandingan Metode Cart Dan Naive Bayes Untuk Klasifikasi Customer Churn', *INFOTECH J.*, vol. 9, no. 2, pp. 307–318, 2023, doi: 10.31949/infotech.v9i2.5641.
- [12] J. Elektronik, I. K. Udayana, J. H. Abednigo, A. Raharja, and K. Selatan, 'Implementasi Decision Tree berbasis Forward Selection untuk Klasifikasi Penyakit Ginjal Kronis', *J. Elektron. Ilmu Komput. Udayana*, vol. 12, pp. 277–286, Nov. 2023.
- [13] C. N. Syahputri and M. S. Hasibuan, 'Optimasi Klasifikasi Decision Tree Dengan Teknik Pruning Untuk Mengurangi Overfitting', *JSiI | J. Sist. Inf.*, vol. 11, no. 2, pp. 87–96, 2024, doi: 10.30656/jsii.v11i2.9161.
- [14] T. M. Shami, A. A. El-Saleh, M. Alswaiti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, 'Particle Swarm Optimization: A Comprehensive Survey', *IEEE Access*, vol. 10, no. January, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.
- [15] I. K. Ananda, A. Z. Fanani, D. Setiawan, and D. F. Wicaksono, 'Penerapan Random Oversampling dan Algoritma Boosting untuk Memprediksi Kualitas Buah Jeruk', *Edumatic J. Pendidik. Inform.*, vol. 8, no. 1, pp. 282–289, 2024, doi: 10.29408/edumatic.v8i1.25836.
- [16] L. Rubini and P. P. E. Soundarapandian, 'Dataset Penyakit Ginjal Kronis.pdf', UC Irvine Machine Learning Repository. [Online]. Available: <https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease>
- [17] P. M. S. Madani, T. Rohana, K. A. Baihaqi, and A. Fauzi, 'Perbandingan Kinerja Klasifikasi Penyakit Ginjal Menggunakan Algoritma Support Vector Machine (SVM) dan Decision Tree (DT)', *Build. Informatics, Technol. Sci.*, vol. 6, no. 1, pp. 74–82–74–82, 2024, doi: 10.47065/bits.v6i1.5206.
- [18] W. Widiati, N. Iriadi, I. Ariyati, I. Nawawi, and S. Sugiono, 'Pendekatan Hibrida Decision Tree-Particle Swarm Optimization untuk Deteksi Dini Penyakit Ginjal Kronis', *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 6, no. 1, pp. 11–22, 2024, doi: 10.26905/jasiek.v6i1.13006.
- [19] E. Purwaningsih and E. Nurelasari, 'Peningkatan Akurasi Metode Support Vector Machine melalui Particle Swarm Optimization pada Penyakit Ginjal Kronis', *Inf. Manag. Educ. Prof.*, vol. 9, no. 1, pp. 61–70, 2024.
- [20] A. Rosyida and T. B. Sasongko, 'Early Detection of Alzheimer's Disease with the C4.5 Algorithm Based on BPSO (Binary Particle Swarm Optimization)', *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 12, no. 3, pp. 341–349, 2023, doi: 10.32736/sisfokom.v12i3.1716.
- [21] 'Eskiyaturrofikoh' and R. R. 'Suryono, 'Analisis Sentimen Aplikasi X Pada Google Play Store Menggunakan Algoritma Naive Bayes Dan Support Vector Machine (Svm)', *JUPI(Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 3, pp. 1408–1419, 2024, [Online]. Available: <https://www.jurnal.stkipppgritulungagung.ac.id/index.php/jipi/article/view/5392>
- [22] P. B. N. Setio, D. R. S. Saputro, and Bowo Winarno, 'Klasifikasi Dengan Pohon Keputusan Berbasis Algoritme C4.5', *Prism. Pros. Semin. Nas. Mat.*, vol. 3, pp. 64–71, 2020.
- [23] A. Pramudyantoro, E. Utami, and D. Ariatmanto, 'Penggabungan K-Nearest Neighbors dan Lightgbm Untuk prediksi Diabetes Pada Dataset Pima Indians: Menggunakan pendekatan Exploratory Data Analysis', *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 3, pp. 1133–1144, 2024, [Online]. Available: <https://jurnal.stkipppgritulungagung.ac.id/index.php/jipi/article/view/4966/2114>
- [24] R. N. Ramadhon, A. Ogi, A. P. Agung, R. Putra, S. S. Febrihartina, and U. Firdaus, 'Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank', *Karimah Tauhid*, vol. 3, no. 2, pp. 1860–1874, 2024, doi: 10.30997/karimahtauhid.v3i2.11952.
- [25] Z. M. Ahmad Putra, P. Asri, F. Romadloni, and R. R. Arnestanta, 'Penerapan Algoritma Particle Swarm Optimization Untuk Meningkatkan Efisiensi Daya Keluaran Panel Surya', *J. Tek. Elektro dan Komput. TRIAC*, vol. 10, no. 2, pp. 56–64, 2023, doi: 10.21107/triac.v10i2.20717.
- [26] E. P. Saputra, S. Nurajizah, M. Maulidah, N. Hidayati, and T. Rahman, 'Komparasi Machine Learning Berbasis Pso Untuk Prediksi Tingkat Keberhasilan Belajar Berbasis E-Learning', *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 2, pp. 321–328, 2023, doi: 10.25126/jtiik.20231026469.