
ANALISIS PENERAPAN OPTIMASI PERBANDINGAN KINERJA ALGORITMA C4.5 DAN NAIVE BAYES BERBASIS PARTICLE SWARM OPTIMIZATION (PSO) UNTUK MENDETEKSI GAGAL GINJAL KRONIK

Delvi Fitri Assary¹, Aida Nirmalasari², Taufik Satria Nugraha³

NIM : ¹H1D020010, ²H1D020018, ³H1D020028

^{1,2,3}Informatika, Fakultas Teknik, Universitas Jenderal Soedirman, Indonesia

Email: delvi.assary@mhs.unsoed.ac.id¹, aida.nirmalasari@mhs.unsoed.ac.id²,
taufik.nugraha@mhs.unsoed.ac.id³

(Artikel dikirimkan tanggal : 04 Juni 2022)

Abstrak

Gagal Ginjal Kronik terjadi perlahan-lahan, bisa dalam hitungan bulan bahkan tahun, dan sifatnya tidak dapat disembuhkan. Perlu dilakukan upaya pencegahan untuk meningkatkan kesadaran masyarakat dalam mengenali gejala dan risiko penyakit gagal ginjal kronik. Sejalan dengan hal itu data mining merupakan salah satu pemanfaatan teknologi informasi dalam bidang kesehatan yang banyak digunakan sebagai sistem pendukung keputusan klinis dalam memprediksi dan mendiagnosa berbagai penyakit dengan *accuracy* data yang sangat baik. Pengujian ini bertujuan mengevaluasi perbandingan penerapan optimasi kinerja terbaik metode klasifikasi data mining algoritma C4.5 dan Naïve Bayes berbasis *Particle Swarm Optimization* (PSO) untuk mendeteksi penyakit gagal ginjal menggunakan pengukuran confusion matrix, AUC dan T-Test. Dataset gagal ginjal yang digunakan berjumlah 400 record dengan 25 parameter indikator yang diolah menggunakan software RapidMiner Versi 9. Hasil penelitian ini didapatkan bahwa optimasi *Particle Swarm Optimization* (PSO) dapat meningkatkan kinerja *accuracy* C4.5 dari 93,75% menjadi 97,50% dan Naïve Bayes 96,75% menjadi 98,25%. Hasil kinerja terbaik yang diuji menggunakan T-Test adalah algoritma C4.5 (PSO) memiliki nilai tertinggi sebesar 0,975, sedangkan algoritma Naïve Bayes *Particle Swarm Optimization* (PSO) memiliki nilai tertinggi sebesar 0,983. Dengan demikian algoritma Naïve Bayes berbasis *Particle Swarm Optimization* (PSO) dapat memberikan solusi terbaik terhadap *accuracy* pendeteksian penyakit gagal ginjal kronik.

Kata kunci: algoritma C4.5, data mining, gagal ginjal, klasifikasi, Naive Bayes, Particle Swarm Optimization

ANALYSIS OF PERFORMANCE APPLICATION OF COMPARISONAL OPTIMIZATION OF C4.5 AND NAIVE BAYES ALGORITHM BASED ON PARTICLE SWARM OPTIMIZATION (PSO) FOR DETECTING CHRONIC KIDNEY DISEASE

Abstract

Chronic Kidney Failure occurs slowly, can take months or even years, and is not curable. Prevention efforts need to be made to increase public awareness in recognizing the symptoms and risks of chronic kidney failure. In line with this, data mining is one of the uses of information technology in the health sector which is widely used as a clinical decision support system in predicting and diagnosing various diseases with very good data accuracy. This test aims to evaluate the comparison of the application of the best performance optimization data mining classification method C4.5 algorithm and Naïve Bayes-based Particle Swarm Optimization (PSO) to detect kidney failure using confusion matrix, AUC, and T-Test measurements. The kidney failure dataset used amounted to 400 records with 25 indicator parameters processed using RapidMiner Version 9. The results of this study showed that Particle Swarm Optimization (PSO) can improve the performance of C4.5 accuracy from 93.75% to 97.50% and Nave Bayes 96.75% to 98.25%. The best performance results tested using the T-Test are the C4.5 algorithm (PSO) which has the highest value of 0.975, while the Naïve Bayes algorithm based on Particle Swarm Optimization (PSO) has the highest value of 0.983. Thus, the Naïve Bayes algorithm based on Particle Swarm Optimization (PSO) can provide the best solution for the accuracy of detecting chronic kidney disease.

Keywords: C4.5 algorithm, data mining, kidney failure, classification, Naive Bayes, Particle Swarm Optimization

1. PENDAHULUAN

Di negara maju ataupun negara berkembang penyakit kronik tidak menular seperti penyakit kardiovaskuler, hipertensi, diabetes melitus, dan penyakit ginjal kronik, sudah menggantikan penyakit menular sebagai masalah kesehatan masyarakat utama [1]. Di Indonesia, menurut WHO penyakit hipertensi dan gagal ginjal selalu mengalami peningkatan tiap tahunnya. Gagal Ginjal Kronik (GGK) adalah suatu sindrom klinis disebabkan penurunan fungsi ginjal yang bersifat menahun, berlangsung progresif dan cukup lanjut, serta bersifat persisten dan ireversibel [2].

Berdasarkan hal tersebut, perlu dilakukan upaya pencegahan untuk meningkatkan kesadaran masyarakat dalam mengenali gejala dan risiko penyakit gagal ginjal kronik sehingga dapat menentukan langkah-langkah pencegahan dan deteksi dini yang tepat [3]. Sejalan dengan hal itu data mining merupakan salah satu pemanfaatan teknologi informasi dalam bidang kesehatan yang banyak digunakan sebagai sistem pendukung keputusan klinis dalam memprediksi dan mendiagnosa berbagai penyakit dengan *accuracy* data yang sangat baik.

Dengan adanya data history penderita gagal ginjal dapat dibuat rekomendasi prediksi penyakit gagal ginjal. Klasifikasi merupakan salah satu teknik data mining untuk membantu prediksi, dilakukan dengan Decision Tree yaitu dengan algoritma C4.5 dan Naive Bayes. Pengujian bertujuan mengevaluasi perbandingan penerapan optimasi kinerja terbaik metode klasifikasi data mining algoritma C4.5 dan Naive Bayes berbasis Particle Swarm Optimization (PSO) untuk mendeteksi penyakit gagal ginjal kronik menggunakan pengukuran confusion matrix, AUC dan T-Test [4]. Dataset gagal ginjal yang digunakan berjumlah 400 record dengan 25 parameter indikator yang terdiri dari *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*, serta *class* dengan *class* sebagai label dalam pengujian.

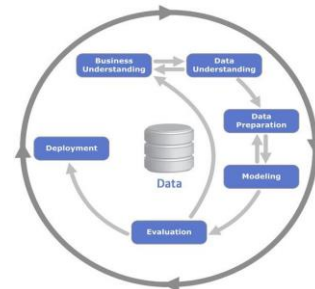
Perbandingan klasifikasi data mining menggunakan algoritma (Naive Bayes dan C4.5) dalam mengelola data transaksi penjualan POS (Pont Of Sales) [5]. Hasil klasifikasi dari masing-masing algoritma dalam penelitian ini nantinya akan dibandingkan untuk mendapatkan evaluasi kinerja terbaik dalam pendeteksian penyakit gagal ginjal.

Dengan demikian, dibutuhkan salah satu teknik data optimasi yang bertujuan untuk meningkatkan kinerja metode klasifikasi data mining konvensional yang sudah dipilih. Salah satu algoritma optimasi yang cukup populer adalah *Particle Swarm Optimization* (PSO). *Particle Swarm Optimization*

(PSO) telah banyak memecahkan masalah optimasi algoritma [6], [7].

2. METODOLOGI PENELITIAN

Pada penelitian ini, menggunakan langkah metodologi yang terdiri dari beberapa tahap berdasarkan tahapan proses CRISP-DM dan dapat ditunjukkan oleh Gambar 1.



Gambar 1 CRISP-DM

Langkah yang dilakukan di metodologi penelitian pada Gambar 1 dijelaskan sebagai berikut:

2.1 Business Understanding (Pemahaman Bisnis)

Dalam penelitian ini berfokus pada pendeteksi penyakit gagal ginjal dengan menggunakan perbandingan dua algoritma klasifikasi data mining yaitu Algoritma C4.5 dan Naive Bayes berbasis *Particle Swarm Optimization* (PSO) [8].

2.2 Data Understanding (Pemahaman Data)

Dataset gagal ginjal yang digunakan pada penelitian ini diambil dari sumber: https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease [9]

2.3 Data Preparation (Persiapan Data)

Untuk mendukung penelitian ini dilakukan pemilihan data terhadap seluruh indikator dalam membentuk dataset penyakit gagal ginjal. Dataset yang dipakai untuk penyakit gagal ginjal ini berjumlah 400 dengan 25 parameter indikator yang terdiri dari *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*, serta *class* sebagai label.

Tabel 1. Hasil Kategorisasi Variabel Dataset

Var	Attributes	Type	Information
X1	Age	Integer	{age in year}
X2	Blood Pressure	Integer	{bp in mm/hg}

X3	Specific Gravity	Real	{sg in (1.0005, 1.010, 1.015, 1.020, 1.025)}
X4	Albumin	Integer	{al in (0,1,2,3,4,5)}
X5	Sugar	Integer	{su in (0,1,2,3,4,5)}
X6	Red Blood Cells	Polynomial	{normal, abnormal}
X7	Pus Cell	Polynomial	{normal, abnormal}
X8	Pus Cell Clumps	Polynomial	{present, not present}
X9	Bacteria	Polynomial	{present, not present}
X10	Blood Glucose Random	Integer	{bgr in mgs/dl}
X11	Blood Urea	Integer	{bu in mgs/dl}
X12	Serum Creatinine	Integer	{sc in mEq/l}
X13	Sodium	Integer	{so in mEq/l}
X14	Potassium	Integer	{po in mEq/l}
X15	Hemoglobin	Integer	{hmo in gms}
X16	Packed Cell Volume	Integer	{numerical}
X17	White Blood Cell Count	Integer	{wc in cells/cumm}
X18	Red Blood Cell Count	Integer	{rc in millions/cmm}
X19	Hypertension	Polynomial	{yes, no}
X20	Diabetes Mellitus	Polynomial	{yes, no}
X21	Coronary Artery Disease	Polynomial	{yes, no}
X22	Appetite	Polynomial	{good, poor}
X23	Pedal Edema	Polynomial	{yes, no}
X24	Anemia	Polynomial	{yes, no}
Y	Class	Label	{ckd, notckd}

Setelah mendapatkan variabel-variabel yang mempengaruhi, dilakukan kategorisasi terhadap setiap data yang ada, hasil kategorisasi dapat dilihat pada Tabel 1.

2.4 Modelling (Pemodelan)

Model klasifikasi ini diawali dengan dataset yang akan dilakukan pemodelan dengan algoritma klasifikasi sehingga dihasilkan model klasifikasi yang memunculkan parameter evaluasi. Pemodelan yang dilakukan yaitu menggunakan Algoritma C4.5 dan Algoritma Naive Bayes berbasis *Particle Swarm*

Organization (PSO) yang telah melalui proses *Preprocessing* Data.

2.4.1 Preprocessing Data

Data *Preprocessing* adalah salah satu tugas data mining yang meliputi persiapan dan transformasi data ke dalam bentuk yang sesuai dengan prosedur mining. *Preprocessing* data bertujuan untuk memperkecil ukuran data, menemukan relasi antar data, menormalkan data, menghilangkan outlier dan mengekstrak fitur untuk data. Ini mencakup beberapa teknik seperti pembersihan data, integrasi, transformasi dan reduksi[10].

Pada tahap awal pengolahan data agar dihasilkan data yang berkualitas, beberapa tahapan yang dilakukan adalah sebagai berikut [11]:

- Data *validation*, untuk mengidentifikasi dan menghapus data yang ganjil, (*outlier/noise*), data yang tidak konsisten, dan data yang tidak lengkap (*missing value*).
- Data *intergration and transformation*, untuk meningkatkan *accuracy* dan efisiensi algoritma. Data yang digunakan dalam penelitian ini bernilai kategorikan.
- Data *size reduction and dicritization*, untuk memperoleh *dataset* dengan jumlah atribut dan *record* yang lebih sedikit tetapi bersifat informatif.

2.4.2 Implementasi C4.5

Langkah yang dilakukan pada tahap ini adalah menerjemahkan desain yang telah terbentuk menjadi sebuah sistem yang menerapkan algoritma C4.5 [12].

Algoritma C4.5 digunakan untuk membentuk pohon keputusan. Ada beberapa tahap dalam membuat sebuah pohon keputusan dengan algoritma C4.5, yaitu [13]:

- Memilih atribut sebagai akar.
- Membuat cabang untuk masing-masing nilai.
- Membagi setiap kasus dalam cabang.
- Mengulangi proses dalam setiap cabang sehingga semua kasus dalam cabang memiliki kelas yang sama.

Kemudian dilakukan perhitungan untuk mencari nilai *entropy* dan *gain*. Berikut ini rumus untuk mencari nilai *entropy* dan *gain*.

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (1)$$

Persamaan (1) adalah persamaan yang digunakan dalam perhitungan *entropy* untuk menentukan *heterogenety* dari sebuah kumpulan data sample. Berikut keterangannya :

S : himpunan kasus

A : atribut

n : jumlah partisi S

Pi : proporsi dari Si terhadap S

Kemudian hitung nilai *gain* dengan metode *information gain* :

$$Gain(S, A) = Entropy(S) - \sum_{i=0}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (2)$$

Persamaan (2) adalah persamaan yang digunakan dalam perhitungan gain setelah melakukan perhitungan *entropy*. Berikut keterangannya :

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|S_i| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

Dengan mengetahui rumus-rumus diatas, data yang telah diperoleh dapat dimasukkan dan diproses dengan algoritma C4.5 untuk proses pembuatan *decision tree*.

2.4.3 Implementasi Naive Bayes

Klasifikasi Bayesian adalah klasifikasi statistik yang bisa memprediksi probabilitas sebuah class. Dihitung berdasarkan Teorema Bayes. Teorema Bayes adalah perhitungan statistik dengan menghitung probabilitas kemiripan kasus lama yang ada dibasis kasus dengan kasus baru [14]. Persamaan dari Teorema Bayes adalah sebagai berikut:

$$P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)} \quad (3)$$

Keterangan :

X : Kriteria suatu kasus berdasarkan masukan

C_i : Kelas solusi pola ke-i, dimana i adalah jumlah label kelas

P(C_i|X) : Probabilitas kemunculan label kelas C_i dengan kriteria masukan X

P(X|C_i) : Probabilitas kriteria masukan X dengan label kelas C_i

P(C_i) : Probabilitas label kelas C_i

2.4.4 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) memiliki karakteristik yaitu konsepnya sederhana, mudah implementasinya, efisien dalam komputasi [15]. Modifikasi kecepatan dan posisi tiap partikel dapat dihitung menggunakan kecepatan saat ini dan jarak pbest_{i,d} ke gbest_d seperti ditunjukkan persamaan berikut:

$$v_{i,m} = w.v_{i,m} + c_1 * R * (pbest_{i,m} - x_{i,m}) + c_2 * R * (gbest_m - x_{i,m}) \quad (4)$$

Menghitung kecepatan baru untuk tiap partikel berdasarkan pada kecepatan sebelumnya (V_{i,m}), lokasi partikel dimana nilai fitness terbaik telah dicapai (pbest), dan lokasi populasi global (gbest untuk versi global, lbest untuk versi local) atau local neighborhood pada algoritma versi local dimana nilai fitness terbaik telah dicapai.

$$x_{id} = x_{i,m} + v_{i,m} \quad (5)$$

Memperbaharui posisi tiap partikel pada ruang solusi. Dua bilangan acak c₁ dan c₂ dibangkitkan sendiri. Penggunaan berat inersia w telah memberikan performa yang meningkat pada sejumlah aplikasi [16]. Hasil dari perhitungan partikel yaitu kecepatan partikel diantara interval [0,1].

Keterangan:

n : jumlah partikel dalam kelompok

d : dimensi

v_{i,m} : kecepatan partikel ke-i pada iterasi ke-i

w : faktor bobot inersia

c₁, c₂ : konstanta akselerasi (learning rate)

R : bilangan random (0-1)

x_{i,d} : posisi saat ini dari partikel ke-i pada iterasi ke-i

pbest : posisi terbaik sebelumnya dari partikel ke-i

gbest : partikel terbaik diantara semua partikel dalam satu kelompok atau populasi.

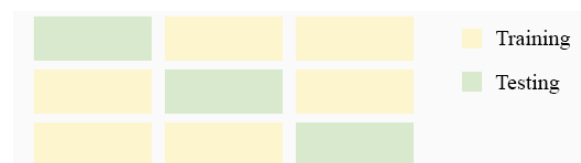
2.5 Evaluation (Pengujian)

Pada tahap Pengujian akan dilakukan perbandingan kuantitatif dengan mempertimbangkan nilai komparasi *confusion matrix* dengan hasil pengukuran berupa *Accuracy*, *Precision*, dan *Recall* dengan menggunakan metode *K-Fold Cross Validation* untuk klasifikasi. Mengevaluasi *accuracy* dari model klasifikasi sangat penting. Untuk mengevaluasi model digunakan metode *confusion matrix*, dan kurva ROC (*Receiver Operating Characteristic*).

Validasi adalah proses mengevaluasi *accuracy* prediksi dari sebuah model, validasi mengacu untuk mendapatkan prediksi dengan menggunakan model yang ada kemudian membandingkan hasil yang diperoleh dengan hasil yang diketahui. Pegukuran pada evaluasi kinerja klasifikasi bertujuan untuk mengetahui seberapa akurat model klasifikasi dalam prediksi kelas dari suatu baris data [17].

2.5.1 K-Fold Cross Validation

Cross Validation atau dapat disebut estimasi rotasi adalah sebuah teknik validasi model untuk menilai bagaimana hasil statistik analisis akan menggeneralisasi kumpulan data independen. Teknik ini utamanya digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya. Salah satu teknik dari validasi silang adalah *K-Fold Cross Validation*, yang mana memecah data menjadi K bagian set data dengan ukuran yang sama. Penggunaan *K-Fold Cross Validation* untuk menghilangkan bias pada data. Pelatihan dan pengujian dilakukan sebanyak K kali, dapat dilihat pada Gambar 2 [18].



Gambar 2. 3-Fold Cross Validation Skema

Model validasi yang digunakan pada penelitian ini adalah *3-Fold Cross Validation*. *3-Fold Cross Validation* digunakan untuk mengukur kinerja model prediksi. Setiap dataset secara acak dibagi menjadi 3 bagian dengan ukuran yang sama. Selama 3 kali, 2

bagian untuk melatih model (data training) dan 1 bagian digunakan untuk menguji (data testing) yang lainnya setiap kali dilakukan pengujian, dapat dilihat pada Tabel 2 contoh pembagian data training dan data testing menggunakan metode *K-Fold Cross Validation*. [19].

Tabel 2. Contoh Pembagian Data Training dan Data Testing

Pengujian	Data Training	Data Testing
Pengujian 3-Fold Cross Validation	267	133
Pengujian 4-Fold Cross Validation	300	100
Pengujian 5-Fold Cross Validation	320	80
Pengujian 8-Fold Cross Validation	350	50
Pengujian 10-Fold Cross Validation	360	40

2.5.2 Confusion Matrix

Confusion matrix memberikan rincian klasifikasi, kelas yang diprediksi akan ditampilkan di bagian atas matrix dan kelas yang diobservasi ditampilkan di bagian kiri. Evaluasi model confusion matrix menggunakan tabel seperti matrix di bawah:

Tabel 3. Confusion Matrix

Class		Actual	
		True	False
Predic	True	True Positif (TP)	False Negative (FN)
	False	False Positive (FP)	True negative (TN)

True positive (tp) merupakan jumlah record positif dalam data set yang diklasifikasikan positive. True negative (tn) merupakan jumlah record negative dalam data set yang diklasifikasikan negative. False positive (fp) merupakan jumlah record negatif dalam data set yang diklasifikasikan positif. False negative (fn) merupakan jumlah record positive dalam data set yang diklasifikasikan negative. Metode *confusion matrix* merepresentasikan hasil evaluasi model dengan menggunakan tabel matriks, jika dataset terdiri dari dua kelas, kelas pertama dianggap positif, dan kelas kedua dianggap negatif. Evaluasi menggunakan *confusion matrix* menghasilkan nilai *accuracy*, *presicion*, *recall*. *Accuracy* dalam klasifikasi merupakan persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi.

Precision atau *confidence* merupakan proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* merupakan proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar.

Precision merupakan perhitungan terhadap perkiraan proporsi kasus positif yang benar dan dirumuskan dalam:

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Recall merupakan perhitungan terhadap perkiraan proporsi kasus positif yang diidentifikasi benar dan dirumuskan dalam:

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

Accuracy merupakan perhitungan terhadap proporsi dari jumlah total prediksi yang benar dan dirumuskan dalam persamaan 7:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

Keterangan :

TP : True Positive

TN : True Negative

FP : False Positive

FN : False Negative

2.5.3 Kurva ROC

Kurva *ROC* adalah tool dua dimensi yang digunakan untuk menilai kinerja klasifikasi yang menggunakan dua class keputusan, masing-masing objek dipetakan ke salah satu elemen dari himpunan pasangan, positif atau negatif. Pada kurva ROC, TP rate diplot pada sumbu Y dan FP rate diplot pada sumbu X. The Area Under Curve (AUC) dihitung untuk mengukur perbedaan performansi metode yang digunakan. AUC dihitung menggunakan rumus

$$\theta^r = \frac{1}{mn} \sum_j^n = 1 \sum_i^m = 1 \psi(x_i^r, x_j^r) \quad (9)$$

Dimana :

$$\psi(X, Y) = \begin{cases} 1 & Y < X \\ \frac{1}{2} & Y = X \\ 0 & Y > X \end{cases} \quad (10)$$

X= Output Positif

Y= Output Negatif

Tingkat *accuracy* dapat didiagnosa sebagai berikut.

Accuracy 0.90 –1.00 = Excellent classification

Accuracy 0.80 –0.90 = Good classification

Accuracy 0.70 –0.80 = Fair classification

Accuracy 0.60 –0.70 = Poor classification

Accuracy 0.50 –0.60 = Failure classification

2.6 Deployment (Penyebaran)

Tahapan terakhir ini yang menjadi penentuan model klasifikasi yang memiliki nilai terbaik dari hasil uji T-Test hasil komparasi model C4.5 berbasis *Particle Swarm Optimization* (PSO) dan Naive Bayes berbasis *Particle Swarm Optimization* (PSO.) Kemudian dari hasil perbandingan akan terpilih model algoritma mana yang lebih baik digunakan dengan hasil uji yang sesuai untuk mendeteksi penyakit gagal ginjal.

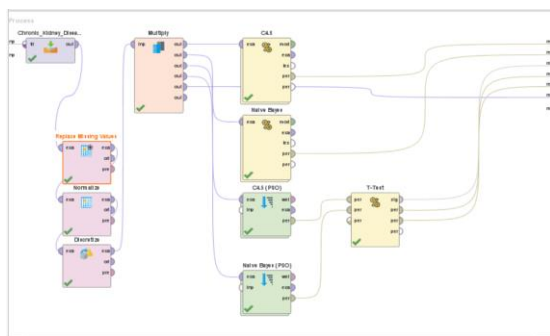
3. HASIL DAN PEMBAHASAN

Dalam melakukan penelitian ini digunakan sistem komputer dengan spesifikasi konfigurasi

hardware yaitu Processor Intel Core i-5 2020, RAM 8 GB, Hardisk 512 GB, dan *software* yang digunakan Sistem Operasi Windows 10 dengan *tools* data mining *RapidMiner*.

3.1. Hasil

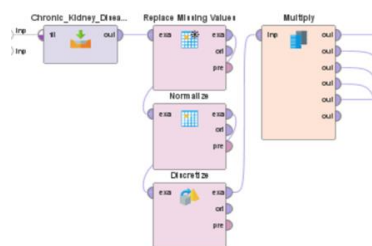
Hasil konfigurasi model pengujian pada RapidMiner dengan perbandingan metode klasifikasi data mining yaitu Algoritma C4.5 dan Naïve Bayes dengan modifikasi *Particle Swarm Optimization* (PSO) menggunakan Preprocessing Data, K-Fold Cross validation, dan uji beda T-Test untuk mencari kinerja terbaik dapat dilihat Main Model Pengujiannya pada Gambar 3.



Gambar 3 Main Model Pengujian

3.1.1 Preprocessing Data

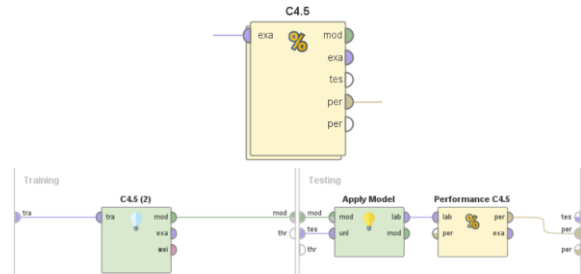
Proses yang pertama adalah read excel untuk menambahkan dataset chronic kidney disease, lalu masuk ke proses preprocessing data menggunakan operator *replace missing value* yang digunakan untuk mengganti nilai yang hilang dimana kami menggunakan semua atribut yang ada pada dataset dengan pengganti yang telah ditentukan. Selanjutnya pada operator *normalize* digunakan untuk menormalkan nilai atribut yang telah dipilih, atribut dengan tipe numeriklah yang nantinya akan dinormalisasi. Dilanjut dengan operator *discretize* yang digunakan untuk mendiskritisasi atribut numerik yang dipilih ke dalam jumlah bin. Terakhir terdapat operator *multiply*, operator ini yang akan mengambil objek pada RapidMiner dari port input dan mengirimkan salinannya ke port output. Data yang tadinya hanya satu jadi memiliki salinan dan dapat digunakan pada proses perhitungan yang lainnya. Dapat dilihat pemodelan *preprocessing* data menggunakan RapidMiner pada Gambar 4 di bawah.



Gambar 4 Preprocessing Data

3.1.2 Algoritma C4.5

Hasil model konfigurasi Algoritma C4.5 pada RapidMiner dengan menggunakan metode *3-Fold Cross Validation* dapat digambarkan seperti pada Gambar 5.



Gambar 5 Konfigurasi Model Algoritma C4.5

Pada tahap awal data masuk ke operator K-Fold Cross Validation, pada tahap ini menggunakan metode *3-Fold Cross Validation* yang digunakan untuk mengukur kinerja model prediksi. Setiap dataset secara acak dibagi menjadi 3 bagian dengan ukuran yang sama, 1 bagian untuk data testing dan 2 bagian untuk data training.

Tahap Selanjutnya terdapat operator algoritma C4.5 yang dapat digunakan untuk proses klasifikasi, dataset akan diolah pada operator algoritma ini. Data yang diolah pada operator algoritma C4.5 ini adalah data training. Lalu untuk data testing yaitu data yang telah melalui tahap uji oleh algoritma C4.5, setelah itu data memasuki operator Apply Model. Operator Apply Model adalah operator yang menerapkan data pada example set, sehingga data ini dapat diterapkan pada example set lain. Pada tahapan selanjutnya yaitu data memasuki operator Performance untuk mendapatkan hasil evaluasi dari kinerja algoritma C4.5, pada operator performance ini data training dan data testing menghasilkan evaluasi berupa *accuracy precision*, *recall*, dan juga AUC.

accuracy: 93.75% +/- 1.15% (micro average: 93.75%)

	true cid	true notcid	class prediction
pred cid	228	3	98.70%
pred notcid	22	147	96.98%
class recall	91.20%	98.00%	

Gambar 6 Hasil Confusion Matrix Algoritma C4.5

Hasil pengujian dan validasi melalui confusion matrix Algoritma C4.5 pada RapidMiner tervisualisasi pada hasil *performance* dengan pengukuran *accuracy*, *precision*, dan *recall* digambarkan pada Gambar 4 dan Gambar 5 sedangkan pada Gambar 7 adalah hasil kurva ROC.

PerformanceVector

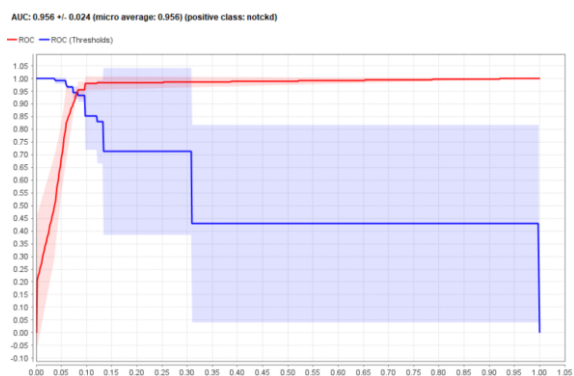
```

PerformanceVector:
accuracy: 93.75% +/- 1.15% (micro average: 93.75%)
ConfusionMatrix:
True: ckd notckd
ckd: 228 3
notckd: 22 147
precision: 87.13% +/- 3.63% (micro average: 86.98%) (positive class: notckd)
ConfusionMatrix:
True: ckd notckd
ckd: 228 3
notckd: 22 147
recall: 98.04% +/- 3.40% (micro average: 98.00%) (positive class: notckd)
ConfusionMatrix:
True: ckd notckd
ckd: 228 3
notckd: 22 147
AUC (optimistic): 0.981 +/- 0.010 (micro average: 0.981) (positive class: notckd)
AUC: 0.956 +/- 0.024 (micro average: 0.956) (positive class: notckd)
AUC (pessimistic): 0.931 +/- 0.039 (micro average: 0.931) (positive class: notckd)

```

Gambar 7 Hasil Performance Algoritma C4.5

Pada Gambar 6 dan 7 terlihat hasil performance algoritma C4.5 dengan accuracy sebesar 93.75%, precision sebesar 87.13%, dan recall sebesar 98.04%.

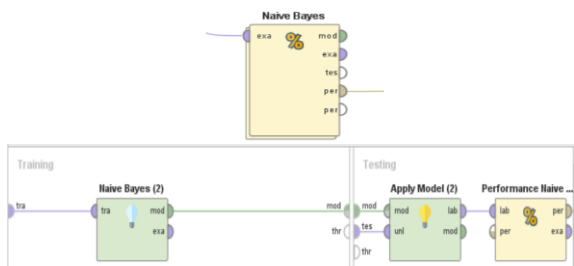


Gambar 8 Kurva ROC Algoritma C4.5

Sedangkan pada Gambar 8 terlihat kurva ROC dari pengujian algoritma C4.5 dengan nilai AUC sebesar 0.956.

3.1.3 Algoritma Naive Bayes

Hasil model konfigurasi Algoritma Naive Bayes pada RapidMiner dengan menggunakan metode 3-Fold Cross Validation dapat digambarkan seperti pada Gambar 9.



Gambar 9 Konfigurasi Model Algoritma Naive Bayes

Tahapannya sama seperti pada konfigurasi algoritma C4.5, hanya mengganti operator algoritma C4.5 dengan Operator Algoritma Naive Bayes.

accuracy: 96.75% +/- 0.88% (micro average: 96.75%)

	true ckd	true notckd	class precision
pred. ckd	240	3	98.77%
pred. notckd	10	147	93.63%
class recall	96.00%	98.00%	

Gambar 10 Hasil Confusion Matrix Naive Bayes

Hasil pengujian dan validasi melalui confusion matrix Algoritma Naive Bayes pada RapidMiner tervisualisasi pada hasil performance dengan pengukuran accuracy, precision, dan recall digambarkan pada Gambar 10 dan 11, sedangkan pada Gambar 12 adalah hasil kurva ROC.

PerformanceVector

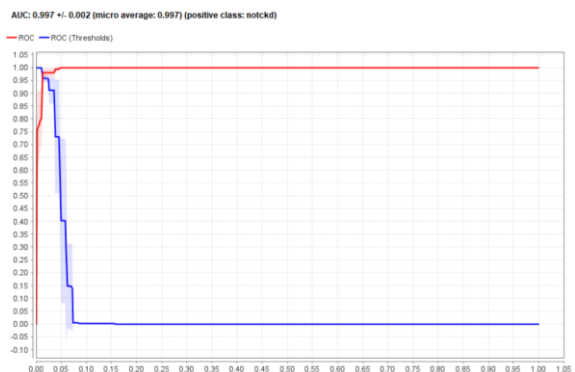
```

PerformanceVector:
accuracy: 96.75% +/- 0.88% (micro average: 96.75%)
ConfusionMatrix:
True: ckd notckd
ckd: 240 3
notckd: 10 147
precision: 93.62% +/- 1.15% (micro average: 93.63%) (positive class: notckd)
ConfusionMatrix:
True: ckd notckd
ckd: 240 3
notckd: 10 147
recall: 97.99% +/- 2.00% (micro average: 98.00%) (positive class: notckd)
ConfusionMatrix:
True: ckd notckd
ckd: 240 3
notckd: 10 147
AUC (optimistic): 0.997 +/- 0.001 (micro average: 0.997) (positive class: notckd)
AUC: 0.997 +/- 0.002 (micro average: 0.997) (positive class: notckd)
AUC (pessimistic): 0.996 +/- 0.002 (micro average: 0.996) (positive class: notckd)

```

Gambar 11 Hasil Performance Algoritma Naive Bayes

Pada Gambar 10 dan 11 terlihat hasil performance algoritma Naive Bayes dengan accuracy sebesar 96.75%, precision sebesar 93.62%, dan recall sebesar 97.99%.

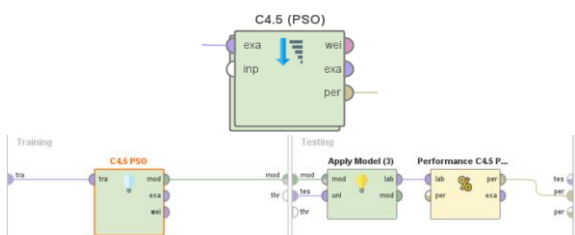


Gambar 12 Kurva ROC Algoritma Naive Bayes

Sedangkan pada Gambar 12 terlihat kurva ROC dari pengujian algoritma C4.5 dengan nilai AUC sebesar 0.997.

3.1.4 Algoritma C4.5 + Particle Swarm Optimization (PSO)

Hasil model konfigurasi Algoritma C4.5 berbasis Particle Swarm Optimization (PSO) pada RapidMiner dengan menggunakan metode 3-Fold Cross Validation dapat digambarkan seperti pada Gambar 13.



Gambar 13 Konfigurasi Model Algoritma C4.5 + PSO

Pada tahap awal data masuk ke operator PSO, PSO ini yaitu metode komputasi yang mengoptimalkan dengan mencoba secara iteratif, untuk meningkatkan solusi kandidat dengan memperhatikan ukuran kualitas yang diberikan.

Tahap kedua data masuk ke operator *K-Fold Cross Validation*, pada tahap ini menggunakan metode *3-Fold Cross Validation* yang digunakan untuk mengukur kinerja model prediksi. Setiap dataset secara acak dibagi menjadi 3 bagian dengan ukuran yang sama, 1 bagian untuk data testing dan 2 bagian untuk data training.

Tahap selanjutnya terdapat operator algoritma C4.5 yang dapat digunakan untuk proses klasifikasi, dataset akan diolah pada operator algoritma ini. Data yang diolah pada operator algoritma C4.5 ini adalah data training. Lalu untuk data testing yaitu data yang telah melalui tahap uji oleh algoritma C4.5, setelah itu data memasuki apply model. Operator Apply Model adalah operator yang menerapkan data pada *example set*, sehingga data ini dapat diterapkan pada *example set* lain. Pada tahapan selanjutnya yaitu data memasuki operator *Performance* untuk mendapatkan hasil evaluasi dari kinerja algoritma C4.5, pada operator performance ini data training dan data testing menghasilkan evaluasi berupa *accuracy*, *precision*, *recall*, dan juga AUC.

accuracy: 97.50% +/- 1.87% (micro average: 97.50%)

	true ckd	true notckd	class precision
pred. ckd	242	2	99.18%
pred. notckd	8	148	94.87%
class recall	96.80%	98.67%	

Gambar 14 Hasil Confusion Matrix Algoritma C4.5 + PSO

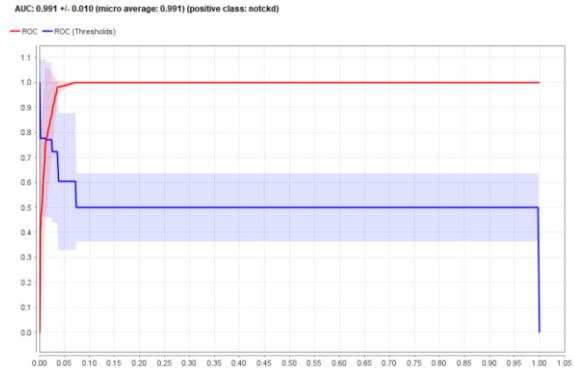
Hasil pengujian dan validasi melalui confusion matrix Algoritma C4.5 berbasis PSO pada RapidMiner tervisualisasi pada hasil *performance* dengan pengukuran *accuracy*, *precision*, dan *recall* digambarkan pada Gambar 12 sedangkan pada Gambar 16 adalah hasil kurva ROC.

PerformanceVector

```
PerformanceVector:
accuracy: 97.50% +/- 1.87% (micro average: 97.50%)
ConfusionMatrix:
True: ckd notckd
ckd: 242 2
notckd: 8 148
precision: 95.18% +/- 5.32% (micro average: 94.87%) (positive class: notckd)
ConfusionMatrix:
True: ckd notckd
ckd: 242 2
notckd: 8 148
recall: 96.65% +/- 1.17% (micro average: 96.67%) (positive class: notckd)
ConfusionMatrix:
True: ckd notckd
ckd: 242 2
notckd: 8 148
AUC (optimistic): 0.999 +/- 0.001 (micro average: 0.999) (positive class: notckd)
AUC: 0.991 +/- 0.010 (micro average: 0.991) (positive class: notckd)
AUC (pessimistic): 0.983 +/- 0.019 (micro average: 0.983) (positive class: notckd)
```

Gambar 15 Hasil Performance Algoritma C4.5 + PSO

Pada Gambar 14 dan 15 terlihat hasil performance algoritma C4.5 berbasis *Particle Swarm Optimization* (PSO) dengan *accuracy* sebesar 97.50%, *precision* sebesar 95.18%, dan *recall* sebesar 98.65%.

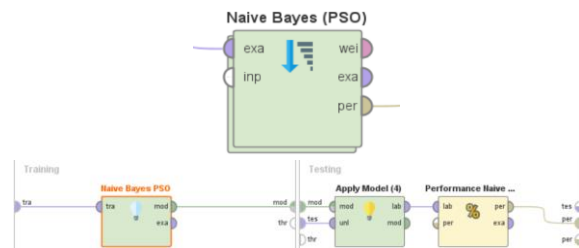


Gambar 16 Kurva ROC Algoritma C4.5 + PSO

Sedangkan pada Gambar 16 terlihat kurva ROC dari pengujian algoritma C4.5 berbasis *Particle Swarm Optimization* (PSO) dengan nilai AUC sebesar 0.991.

3.1.5 Algoritma Naive Bayes + *Particle Swarm Optimization* (PSO)

Hasil model konfigurasi Algoritma Naive Bayes berbasis *Particle Swarm Optimization* (PSO) pada RapidMiner dengan menggunakan metode *3-Fold Cross Validation* dapat digambarkan seperti pada Gambar 17.



Gambar 17 Konfigurasi Model Algoritma Naive Bayes + PSO

Tahapannya sama seperti pada konfigurasi algoritma C4.5 berbasis *Particle Swarm Optimization* (PSO), hanya mengganti operator algoritma C4.5 dengan Operator Algoritma Naive Bayes.

accuracy: 98.25% +/- 1.72% (micro average: 98.25%)

	true ckd	true notckd	class precision
pred. ckd	245	3	98.80%
pred. notckd	4	147	97.35%
class recall	98.40%	98.00%	

Gambar 18 Hasil Confusion Matrix Algoritma Naive Bayes + PSO

Hasil pengujian dan validasi melalui confusion matrix Algoritma Naive Bayes berbasis PSO pada RapidMiner tervisualisasi pada hasil *performance* dengan pengukuran *accuracy*, *precision*, dan *recall* digambarkan pada Gambar 19 dan 20, sedangkan pada Gambar 21 adalah hasil kurva ROC.

PerformanceVector

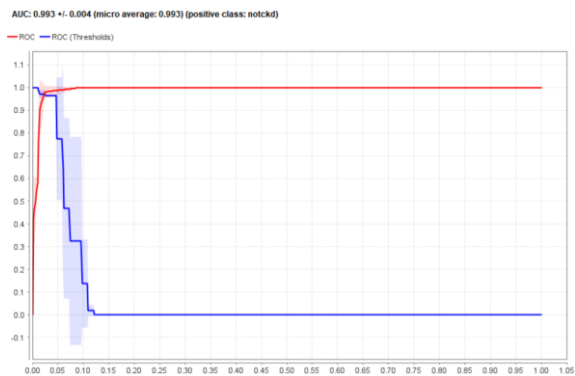
```

PerformanceVector:
accuracy: 98.25% +/- 1.72% (micro average: 98.25%)
ConfusionMatrix:
True:   ckd   notckd
ckd:    246    3
notckd: 4      147
precision: 97.35% +/- 1.17% (micro average: 97.35%) (positive class: notckd)
ConfusionMatrix:
True:   ckd   notckd
ckd:    246    3
notckd: 4      147
recall: 98.04% +/- 3.40% (micro average: 98.00%) (positive class: notckd)
ConfusionMatrix:
True:   ckd   notckd
ckd:    246    3
notckd: 4      147
AUC (optimistic): 0.995 +/- 0.000 (micro average: 0.995) (positive class: notckd)
AUC: 0.993 +/- 0.004 (micro average: 0.993) (positive class: notckd)
AUC (pessimistic): 0.990 +/- 0.008 (micro average: 0.990) (positive class: notckd)

```

Gambar 19 Hasil *Performance* Algoritma Naive Bayes + PSO

Pada Gambar 19 dan 20 terlihat hasil performance algoritma Naive Bayes berbasis *Particle Swarm Optimization* (PSO) dengan accuracy sebesar 98.25%, *precision* sebesar 97.35%, dan *recall* sebesar 98.04.

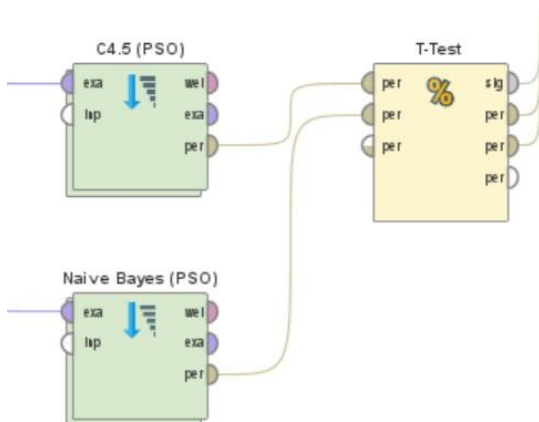


Gambar 21 Kurva ROC Algoritma Naive Bayes + PSO

Gambar 21 terlihat kurva ROC dari pengujian algoritma Naive Bayes berbasis *Particle Swarm Optimization* (PSO) dengan nilai AUC sebesar 0.993.

3.1.6 Uji T-Test

Hasil uji T-Test pada algoritma C4.5 (PSO) dan Naive Bayes (PSO) tertera pada Gambar 22 di bawah ini.



Gambar 22 Uji T-Test

Tahap Terakhir data memasuki operator T-Test. Operator ini digunakan untuk perbandingan kinerja algoritma yang digunakan, dalam kasus ini algoritma

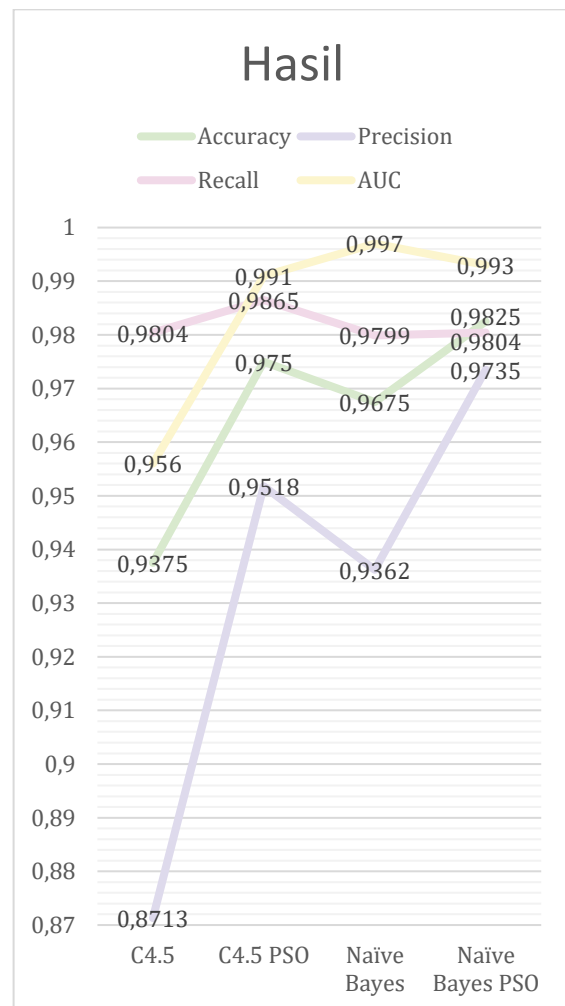
yang digunakan untuk dilakukan uji perbandingan menggunakan operator T-Test adalah kinerja dari Algoritma C4.5 berbasis *Particle Swarm Optimization* (PSO) dan Algoritma Naive Bayes *Particle Swarm Optimization* (PSO) dalam kasus pendeteksian penyakit gagal ginjal.

3.2. Pembahasan

Berdasarkan hasil pengujian dan evaluasi didapatkan perbandingan hasil performa dari masing-masing algoritma dan penerapan optimasi yang dapat dijabarkan sebagai berikut.

Tabel 4. Hasil Komparasi Algoritma

Parameter Kinerja	C4.5	C4.5 PSO	Naive Bayes	Naive Bayes PSO
Accuracy	93.75%	97.50%	96.75%	98.25%
Precision	87.13%	95.18%	93.62%	97.35%
Recall	98.04%	98.65%	97.99%	98.04%
AUC	0.956	0.991	0.997	0.993



Gambar 23 Grafik Hasil Komparasi Algoritma

Berdasarkan hasil penerapan optimasi *Particle Swarm Optimization* (PSO) pada Tabel 2 dan gambar 23 (Grafik Hasil Komparasi Algoritma) didapatkan bahwa terbukti optimasi *Particle Swarm Optimization* (PSO) dapat meningkatkan performa

atau kinerja algoritma C4.5 dan Naive Bayes. Pada indikator *accuracy* algoritma C4.5 mengalami peningkatan dari 93,75% menjadi 97,50% dan *accuracy* Naive Bayes 96,75% menjadi 98,25%. Selain itu berdasarkan evaluasi yang dilakukan secara *confusion matrix* ternyata terbukti bahwa hasil *accuracy* tertinggi pada perbandingan hasil kinerja klasifikasi didapatkan oleh algoritma Naive Bayes berbasis *Particle Swarm Optimization* (PSO) sebesar 98,25% disusul algoritma C4.5 berbasis *Particle Swarm Optimization* (PSO) sebesar 97,50%.

Ditinjau dari indikator pengukuran *precision* dan *recall* juga mengalami peningkatan yang signifikan diantaranya algoritma C4.5 dengan nilai *precision* dan *recall* sebesar 87,13% dan 98,04% meningkat menjadi 95,18% dan 98,65% dan algoritma Naive Bayes dengan nilai *precision* dan *recall* sebesar 93,62% dan 97,99% menjadi 97,35% dan 98,04%.

Analisis yang berbeda ditinjau dari pengukuran AUC bahwa penerapan optimasi berbasis *Particle Swarm Optimization* (PSO) tidak terlalu berpengaruh pada peningkatan kinerja algoritma secara keseluruhan. Terjadi peningkatan pada algoritma C4.5 namun tidak pada algoritma Naive Bayes. Secara keseluruhan algoritma C4.5 dan Naive Bayes berdasarkan nilai AUC masuk pada predikat Excellent Classification.

A	B	C
	0.975 +/- 0.019	0.983 +/- 0.017
0.975 +/- 0.019		0.637
0.983 +/- 0.017		

Gambar 23 Hasil Uji T-Test

Dari hasil T-Test pada Gambar 23 dapat disimpulkan bahwa algoritma Naive Bayes *Particle Swarm Optimization* (PSO) memiliki nilai tertinggi sebesar 0,983 disusul algoritma C4.5 *Particle Swarm Optimization* (PSO) sebesar 0,975. Dengan demikian algoritma Naive Bayes *Particle Swarm Optimization* (PSO) dapat memberikan solusi terbaik terhadap *accuracy* pendeteksian penyakit gagal ginjal.

4. KESIMPULAN

Terbukti optimasi *Particle Swarm Optimization* (PSO) dapat meningkatkan kinerja *accuracy* algoritma C4.5 dan Naive Bayes. Pada *accuracy* algoritma C4.5 mengalami peningkatan dari 93,75% menjadi 97,50%, dan *accuracy* Naive Bayes dengan nilai 96,75% menjadi 98,25%.

Hasil kinerja terbaik yang diuji menggunakan T-Test pada algoritma C4.5 dan Naive Bayes berbasis *Particle Swarm Optimization* (PSO) dapat dihasilkan bahwa algoritma Naive Bayes *Particle Swarm Optimization* (PSO) memiliki nilai tertinggi sebesar 0,983 dilanjutkan algoritma C4.5 *Particle Swarm Optimization* (PSO) sebesar 0,975. Dengan demikian algoritma Naive Bayes *Particle Swarm Optimization* (PSO) dapat memberi solusi terbaik terhadap *accuracy* pendeteksian penyakit gagal ginjal.

DAFTAR PUSTAKA

- [1] B. herilla Sekti, "Hubungan Pola Pengobatan Gagal Ginjal Kronik Terhadap Kepatuhan Pada Pasien Gagal Ginjal Kronik Di Instalasi Hemodialisa Rumah Sakit 'X' Malang," *J. Kesehat. Hesti Wira Sakti*, vol. 7, no. 2, pp. 54–63, 2020, doi: 10.47794/jkhws.v7i2.272.
- [2] S. Nurchayati, T. Bin Sansuwito, and S. Rahmalia, "Gambaran Deteksi Dini Penyakit Gagal Ginjal Kronik Pada Masyarakat Kecamatan Tambang, Kabupaten Kampar," *J. Ners Indones.*, vol. 9, no. 1, p. 11, 2019, doi: 10.31258/jni.9.1.11-18.
- [3] S. R. Widowati, "Tingkat Kualitas Hidup Pasien Gagal Ginjal Kronik Terapi Hemodialisis," *J. Kesehat. Masy.*, vol. 6, no. 2, pp. 107–112, 2011.
- [4] H. Muhamad, C. A. Prasjojo, N. A. Sugianto, L. Surtiningsih, and I. Cholissodin, "Optimasi Naive Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, p. 180, 2017, doi: 10.25126/jtiik.201743251.
- [5] Mirza Yogy Kurniawan and Muhammad Edya Rosadi, "Optimasi Decision Tree Menggunakan Particle Swarm Optimization Pada Data Siswa Putus Sekolah," *J. Teknol. Inf. Univ. Lambung Mangkurat*, vol. 2, no. 1, pp. 7–14, 2017, doi: 10.20527/jtiulm.v2i1.13.
- [6] E. K. R. Sudarsono, J. F. A. Sasmita, A. B. Handyasto, N. Kuswantiningsih, and S. S. Arissaputra, "Peningkatan Pengetahuan Terkait Hipertensi Guna Perbaikan Tekanan Darah pada Pemuda di Dusun Japanan, Margodadi, Seyegan, Sleman, Yogyakarta," *J. Pengabd. Kpd. Masy. (Indonesian J. Community Engag.)*, vol. 3, no. 1, p. 26, 2017, doi: 10.22146/jpkm.25944.
- [7] S. Keperawatan, S. Kusuma, and H. Surakarta, "Hubungan tingkat pengetahuan tentang hipertensi dengan sikap dalam pencegahan komplikasi hipertensi pada lansia peserta PROLANIS UPT PUSKESMAS JENAWI KARANGANYAR Relationship Of The Level Of Knowledge About Hypertension With Attitude In The Prevention Of," *PLACENTUM J. Ilm. Kesehat. dan Apl.*, vol. 7, no. 2, p. 2019, 2019.
- [8] K. A. Cahyanto, F. P. B. Muhamad, and E. Mulyani, "Penerapan Dizcretize By Frequency Dalam Meningkatkan Akurasi Algoritma C4.5 Dalam Memprediksi Cuaca Pada Jalur Pantura Tegal-Pekalongan-Semarang," *JTT (Jurnal Teknol. Ter.)*, vol. 5, no. 2, p. 78, 2019, doi: 10.31884/jtt.v5i2.195.

- [9] D. . Dr.P.Soundarapandian.M.D., "Chronic_Kidney_Disease Data Set," *UCI Machine Learning Repository*, 2015. https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease
- [10] M. A. Hasanah, S. Soim, and A. S. Handayani, "Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 103–108, 2021, doi: 10.30871/jaic.v5i2.3200.
- [11] M. Azhari, Z. Situmorang, and R. Rosnelly, "Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 640, 2021, doi: 10.30865/mib.v5i2.2937.
- [12] Y. I. Kurniawan, "Perbandingan Algoritma Naive Bayes dan C.45 dalam Klasifikasi Data Mining," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, p. 455, 2018, doi: 10.25126/jtiik.201854803.
- [13] T. A. Y. Siswa, "Analisis Penerapan Optimasi Perbandingan Kinerja Algoritma C4.5 Dan Naïve Bayes Berbasis Particle Swarm Optimization (Pso) Untuk," *J. Bangkit Indones.*, vol. 7, no. 2, p. 1, 2018, doi: 10.52771/bangkitindonesia.v7i2.48.
- [14] F. Sistem Informasi STMIK Nusa Mandiri Jakarta Jl Damai No, W. Jati Barat, and J. Selatan, "Perbandingan Algoritma Klasifikasi Data Mining Model C4.5 Dan Naive Bayes Untuk Prediksi Penyakit Diabetes," *J. Techno Nusa Mandiri*, vol. XIII, no. 1, p. 50, 2016.
- [15] Y. D. Pramudita, S. S. Putro, and N. Makhmud, "Klasifikasi Berita Olahraga Menggunakan Metode Naïve Bayes dengan Enhanced Confix Stripping Stemmer," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, p. 269, 2018, doi: 10.25126/jtiik.201853810.
- [16] W. D. Septiani, "Komparasi Metode Klasifikasi Data Mining Algoritma C4.5 Dan Naive Bayes Untuk Prediksi Penyakit Hepatitis," *J. Pilar Nusa Mandiri*, vol. 13, no. 1, pp. 76–84, 2017, [Online]. Available: <https://ejournal.nusamandiri.ac.id/index.php/pilar/article/view/149/126>
- [17] F. Tempola, M. Muhammad, and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, p. 577, 2018, doi: 10.25126/jtiik.201855983.
- [18] D. J. Hand, *Principles of data mining*, vol. 30, no. 7. 2007. doi: 10.2165/00002018-200730070-00010.
- [19] A. E. Pramadhani and T. Setiadi, "Penerapan Data Mining untuk Klasifikasi Penyakit ISPA dengan Algoritma Desicion Tree," *J. Sarj. Tek. Inform. e-ISSN 2338-5197*, vol. 2, no. 1, pp. 831–839, 2014.