Jurnal Elektronik Ilmu Komputer Udayana
Volume 10, No 4. May 2022

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Diagnosis Penyakit Ginjal Kronis dengan Algoritma C4.5, K-Means dan BPSO

I Gede Aditya Mahardika Pratama[a1], Luh Gede Astuti[a2], I Made Widiartha[a3], I Gusti Ngurah Anom Cahyadi Putra[a4], Cokorda Rai Adi Pramartha[a5], I Dewa Made Bayu Atmaja Darmawan[a6]

[a]Program Studi Informatika, Universitas Udayana
Kuta Selatan, Badung, Bali, Indonesia
[1]adityamahardika@student.unud.ac.id
[2]lg.astuti@unud.ac.id
[3]madewidiartha@unud.ac.id
[4]anom.cp@unud.ac.id
[5]cokorda@unud.ac.id
[6]dewabayu@unud.ac.id

Abstrak

Penyakit ginjal kronis atau *Chronic Kidney Disease* (CKD) adalah gangguan pada ginjal yang mengakibatkan ginjal tidak dapat melakukan fungsinya dengan baik karena turunnya kinerja organ ginjal. Klasifikasi adalah teknik *data mining* yang dapat digunakan dalam mendiagnosis penyakit ginjal kronis. Pada penelitian ini, klasifikasi dilakukan dengan menggunakan algoritma C4.5. K-Means *Clustering* digunakan untuk mendiskritisasi data bertipe numerik. *Binary Particle Swarm Optimization* (BPSO) berfungsi untuk menseleksi subset fitur yang berlebihan dan kurang informatif pada dataset atau yang disebut dengan seleksi fitur. Pengujian dilakukan dengan menggunakan metode *10-fold cross validation* pada dataset *Chronic Kidney Disease* (CKD) yang didapat dari *UCI Machine Learning Repository.* Hasil pengujian pada penelitian ini didapatkan bahwa penerapan seleksi fitur dengan BPSO mampu meningkat kinerja klasifikasi C4.5 dengan nilai *accuracy, precision, recall* dan *f-measure* berturut-turut yaitu 96%, 96,869%, 96,8% dan 96,781% serta waktu komputasi yang didapatkan yaitu 62,56 ms. Sedangkan pada pengujian parameter BPSO, didapatkan nilai parameter terbaik dengan jumlah partikel adalah 15, jumlah iterasi adalah 40, nilai c1 adalah 1 dan c2 adalah 1,2 serta nilai bobot inersia (w) adalah 0,9.

Keywords: *Penyakit Ginjal Kronis, Klasifikasi, Algoritma C4.5, K-Means Clustering, Binary Particle Swarm Optimization (BPSO)*

1. Pendahuluan

Ginjal merupakan organ yang membantu menjaga kestabilan dalam tubuh dengan cara menyeimbangkan hasil metabolisme, cairan tubuh dan keseimbangan elektrolit [1]. Selain itu ginjal juga berfungsi memproduksi *hormone enzim* dalam membantu mengendalikan tekanan darah, menjaga susunan tulang menjadi lebih kuat serta memproduksi sel darah merah [2]. Penyakit ginjal kronis atau *Chronic Kidney Disease* (CKD) adalah bentuk gangguan pada ginjal yang mengakibatkan ginjal tidak dapat melakukan fungsinya dengan baik karena turunnya kinerja organ ginjal [3].

Seiring pesatnya laju pertumbuhan penduduk maka semakin bertambah jumlah penyakit CKD. Menurut *Global Burden of Disease*, dikatakan bahwa penyakit CKD menempati rangking ke-27 pada tahun 1990 dan rangking ke-18 pada tahun 2010 [2]. Menurut Kementerian Kesehatan RI, 2 dari setiap 1.000 orang Indonesia atau 499.800 orang menderita penyakit ginjal kronis pada tahun 2013. Prevalensi penyakit ginjal kronis meningkat seiring bertambahnya usia [4].

Klasifikasi adalah teknik *data mining* yang dapat digunakan dalam mendiagnosis penyakit ginjal kronis. Dimana *data mining* merupakan suatu metode yang digunakan untuk menemukan pola dari data yang digunakan untuk mencari solusi dari suatu masalah berdasarkan berbagai aturan proses [5]. Definisi dari klasifikasi adalah porses pencarian kelompok berdasarkan pada suatu dataset [2]. Algoritma C4.5 merupakan algoritma *machine learning* yang dapat digunakan dalam melakukan klasifikasi data. Algoritma C4.5 adalah sebuah metode yang digunakan untuk membentuk sebuah pohon keputusan

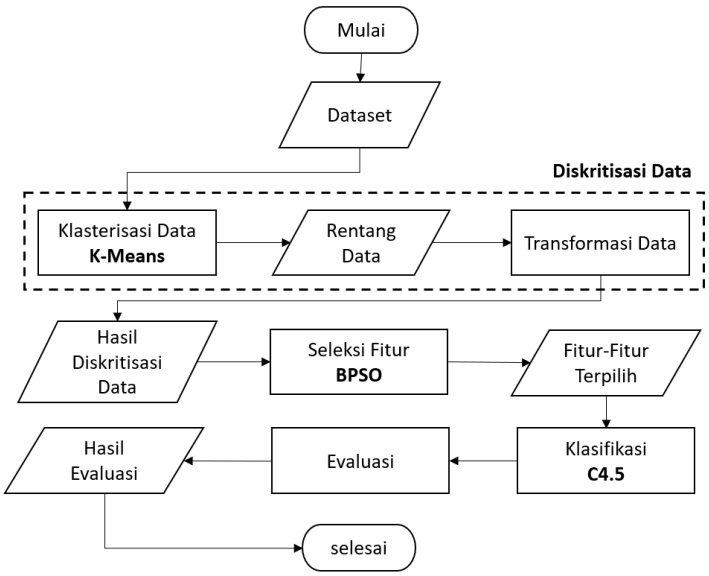
yang merepresentasikan aturan dari fakta yang sangat besar [5]. Penelitian terdahulu tentang klasifikasi telah dilakukan pada tahun 2019 membandingkan algoritma C4.5 dengan Naïve Bayes untuk memprediksi penyakit diabetes. Penelitian ini menunjukan algoritma C4.5 lebih unggul dengan akurasi sebesar 82,74% [6]. Pada penelitian lainnya melakukan klasifikasi keberlangsungan hidup pasien hepatitis dengan membandingkan metode SVM dengan C4.5 dan memperoleh nilai akurasi 80,6452% pada C4.5 dan 80,3279% untuk hasil SVM [7].

Dalam pengklasifikasian dengan C4.5, diperlukan diskritisasi dari suatu kumpulan data numerik. Jika data masih berupa numerik, maka C4.5 akan membentuk sangat banyak cabang pada aturan atau *decision tree* yang dihasilkan. Oleh karena itu, dengan adanya diskritisasi akan mempermudah dalam proses pembentukan *rule* pada C4.5 [8]. Diskritisasi adalah suatu proses konversi pada data numerik menjadi data kategorikal berdasarkan label interval atau label konseptual. Algoritma *clustering* dapat diterapkan untuk mendiskritisasi data numerik dengan mempartisi data ke dalam sebuah *cluster* atau kelompok [9]. Salah satu algoritma *clustering* adalah *K-Means*. Penelitian ini menggunakan algoritma *K-Means* dikarenakan performansi *K-Means* dibandingkan algoritma *clustering* lainnya pada beberapa penelitian yang telah dilakukan sebelumnya. Penelitian tersebut melakukan perbandingan algoritma *K-Means, K-Medoids*, dan DBSCAN pada segmentasi pelanggan berdasarkan RFM. Hasil penelitian menunjukkan bahwa *K-Means* memiliki tingkat validitas yang paling baik dibandingkan dengan *K-Medoids* dan DBSCAN, dimana nilai Davies-Bouldin Index sebesar 0,33009058 dan nilai Silhouette Index sebesar 0,912671056 [10].

Dalam meningkatkan kinerja dari klasifikasi C4.5, dapat dilakukan dengan menseleksi subset fitur yang berlebihan dan kurang informatif pada dataset atau yang disebut dengan seleksi fitur. Pencarian untuk mendapatkan fitur optimal, salah satunya dapat menggunakan *Particle Swarm Optimization* (PSO). *Binary Particle Swarm Optimization* (BPSO) adalah hasil penyesuaian dari PSO yang digunakan sebagai *feature selection* [11]. Penelitian terdahulu menggunakan metode BPSO sebagai *feature selection* dan C4.5 sebagai metode klasifikasi. Penelitian tersebut melakukan deteksi pada penyakit kanker dengan membandingkan kinerja seleksi fitur dari BPSO dengan *Information Gain* (IG). Akurasi yang diperoleh berdasarkan skema BPSO-C4.5 dan IG-C4.5 berturut-turut adalah 99% dan 54% [11].

Berdasarkan latar belakang diatas, maka penelitian ini akan membahas untuk mengukur hasil kinerja C4.5 dengan menggunakan *K-Means Clustering* sebagai diskritisasi data dan *Binary Particle Swarm Optimization* (BPSO) sebagai seleksi fitur, mencari parameter optimal pada BPSO dan mencari jumlah klaster optimal pada K-Means.

2. Metode Penelitian



Gambar 1. Gambaran Umum Sistem

Alur sistem secara umum pada penelitian yang dilakukan oleh penulis, yaitu dimulai dengan menginputkan dataset *Chronic Kidney Disease* yang telah didapatkan pada tahapan pengumpulan data. Dilanjutkan dengan tahapan diskritisasi data yang dimulai dengan proses klasterisasi data dengan

K-Means *Clustering.* Dari proses tersebut akan menghasilkan rentang data dari tiap atribut dan digunakan dalam tahapan transformasi data. Setelah data melewati tahapan diskritisasi, maka seluruh data menjadi data bertipe kategorikal. Lalu dilanjutkan dengan proses seleksi fitur dengan *Binary Particle Swarm Optimization* (BPSO) dari data yang telah terdiskritisasi. Fitur-fitur yang terpilih digunakan dalam tahapan klasifikasi dengan menggunakan algoritma C4.5. Setelah data terklasifikasi akan dilanjutkan dengan tahapan evaluasi untuk mengetahui hasil evaluasi dari model yang dibangun.

2.1. Data dan Metode Pengumpulan Data

Dataset yang digunakan yaitu *Chronic Kidney Disease* (CKD) merupakan data sekunder didapatkan dari *UCI Machine Learning Repository*. Pada dataset *Chronic Kidney Disease* terdapat 400 data dengan pembagian jumlah data untuk setiap kelas yaitu CKD sebanyak 250 data dan NOTCKD sebanyak 150 data. Dataset ini memiliki 25 atribut dimana terdapat 11 atribut numerikal dan 14 atribut kategorikal. Dari 400 jumlah data yang digunakan, dataset akan dibagai menjadi dua bagian yaitu data *training* sebanyak 360 data dan data *testing* sebanyak 40 data.

2.2. K-Means

K-Means adalah algoritma klasterisasi sederhana yang digunakan untuk mempartisi data ke suatu klaster. Algoritma ini relatif cepat, mudah diimplementasikan dan dijalankan, banyak digunakan dan mudah disesuaikan [12]. Pada penelitian ini K-Means dimanfaatkan untuk mendiskritisasi data pada atribut-atribut yang memiliki jenis data numerikal menjadi kategorikal. Proses ini dilakukan sebanyak jumlah atribut numerik yaitu 11, yang artinya proses diskritisasi tiap atribut akan dilakukan secara terpisah. Berikut merupakan tahapan dari proses diskritisasi dengan K-Means:

a) Sebelum dilakukan proses klasterisasi, perlu dilakukan tahapan normalisasi terlebih dahulu agar rentang nilai atau domain tiap atribut menjadi sama dengan rentang nilai [0,1]. Metode *Min-Max Normalization* digunakan untuk proses normalisasi.

b) Tentukan jumlah k-*cluster* yang akan digunakan.

c) Membentuk titik pusat *cluster* atau k-*centroid* secara acak.

d) Hitung jarak setiap *centroid* terhadap masing-masing data menggunakan rumus *Euclidean Distance* menggunakan persamaan sebagai berikut :

$$D(x_2, x_1) = \sqrt{\sum_{i=1}^{n}(x_2 - x_1)^2} \quad (1)$$

Keterangan:

$D(x_2, x_1)$: dimensi data atau jarak data

x_1 : titik pusat *cluster*

x_2 : titik objek data

e) Bentuk kelompok data berdasarkan jarak data dengan *centroid* yang terdekat.

f) Hitungan rata-rata dari setiap *cluster* untuk menentukan nilai *centroid* yang baru dari *cluster* tersebut dengan rumus berikut :

$$C_k = \frac{1}{n_k}\sum d_i \quad (2)$$

Keterangan:

n_k : banyak data dalam *cluster* k

d_i : nilai jarak tiap data pada masing-masing *cluster*

g) Langkah c sampai e terus dilakukan sampai objek setiap *cluster* tidak ada yang berpindah. Setelah hasil klasterisasi didapatkan, akan dilanjutkan dengan denormalisasi pada data agar rentang data yang didapatkan sesuai dengan nilai pada dataset sebelum dinormalisasi.

h) Menentukan rentang data dari tiap klaster berdasarkan data terbesar dan terkecil dari klaster tersebut. Rentang data inilah yang digunakan sebagai dasar diskritisasi data pada data numerik.

2.3. *Binary Particle Swarm Optimization* (BPSO)

Particle Swarm Optimization (PSO) adalah metode optimasi berbasis populasi yang diusulkankan oleh Eberhart dan Kennedy di tahun 1995. Metode ini berbasis pada perilaku sosial sekumpulan burung [13]. Dengan melihat perilaku sekumpulan burung ketika seekor burung memberikan informasi sumber

makakanan ke seluruh kumpulan burung, dan burung-burung tersebut akan mencari sumber makanan tersebut [11]. Pada penelitian ini PSO yang dimodifikasi menjadi BPSO digunakan sebagai proses seleksi fitur dengan tahapan sebagai berikut:

a) Menginisialisasi nilai parameter yang dibutuhkan seperti jumlah partikel dalam populasi, nilai batas iterasi, *cognitive learning* (c1), *social learning* (c2) dan *inertia weight* (w).

b) Langkah selanjutnya adalah menginisialisasi posisi dan kecepatan awal dari semua partikel. Setiap partikel memiliki jumlah fitur (gen) sebanyak jumlah fitur pada dataset. Pembentukan kecepatan awal partikel dibuat dengan nilai 0 lalu pada pembentukan posisi awal partikel dibuat secara *random* dengan nilai 0 atau 1. Inisialisasi partikel digunakan untuk memilih fitur-fitur apa saja yang digunakan pada klasifikasi C4.5.

c) Selanjutnya menghitung nilai *fitness* tiap partikel dilakukan dengan cara melakukan klasifikasi dengan C4.5. Dimana fitur dengan partikel yang bernilai 1 akan digunakan dalam klasifikasi dan nilai partikel 0 tidak akan digunakan dalam klasifikasi. Sehingga nilai akurasi dari C4.5 akan digunakan sebagai nilai *fitness.*

d) Berikutnya penentuan Pbest dan Gbest. Pada awal iterasi, Pbest akan disamakan dengan nilai posisi awal partikel. Sedangkan pada iterasi selanjutnya, 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.

e) Selanjutnya akan dilakukan pengecekan kodisi berhenti berdasarkan jumlah iterasi yang telah dilakukan. Jika jumlah perulangan belum mencapai maksimum iterasi maka akan dilanjutkan ke tahap f dan g. Jika perulangan telah mencari maksimum iterasi maka dilanjutkan ke tahap h.

f) *Update* kecepatan digunakan untuk menentukan ke arah mana partikel akan berpindah dan dapat memperbaiki posisi sebelumnya. Dimana dalam *update* kecepatan tiap partikel menggunakan persamaan (3) [14].

$$v_{id}^{new} = w * v_{id}^{old} + c_1 r_1 \left(pb_{id}^{old} - x_{id}^{old}\right) + c_2 r_2 (gb_{id}^{old} - x_{id}^{old}) \quad (3)$$

Keterangan:

v_{id}^{new} : *velocity* partikel baru

v_{id}^{old} : velocity partikel lama

x_{id}^{old} : titik partikel sebelumnya

r_1 dan r_2 : nilai acak antara 0 dan 1

c_1 : *factor cognitive learning*

c_2 : *factor social learning*

w : bobot inersia

gb : *global best*

pb : *personal best*

g) Setelah didapatkan nilai kecepatan pada tiap partikel, dilanjutkan dengan penentuan posisi baru. Inisialisasi posisi partikel pada PSO dibentuk berdasarkan nilai *random*. Sedangkan pada *feature selection,* penggunaan nilai *random* pada representasi posisi tidak dapat dilakukan karena tidak dapat memperlihatkan fitur apa saja yang digunakan. Oleh karena itu, PSO harus diubah menjadi ke bentuk biner yaitu BPSO. Posisi partikel pada BPSO direpresentasikan ke dalam nilai dengan interval [0,1]. Untuk membatasi nilai kecepatan tiap partikel dilakukan dengan melakukan transformasi *limiting* dengan persamaan berikut [11] :

$$x_{id}^{new} = \begin{cases} 1, & sigmoid(v_{id}^{new}) > rand \\ 0, & sigmoid(v_{id}^{new}) \leq rand \end{cases} \quad (4)$$

rand adalah nilai acak dalam rentang 0 dan 1, persamaan sigmoid menggunakan rumus berikut :

$$sigmoid(v_{id}^{new}) = \frac{1}{1+e^{-v_{id}^{new}}} \quad (5)$$

h) Ketika sudah mencapai iterasi maksimal maka akan didapatkan hasil seleksi fitur terbaik dari partikel *Gbest* pada iterasi terakhir.

2.4. Algoritma C4.5

Dalam membangun sebuah *decision tree* atau pohon keputusan, salah satunya dapat menggunakan algoritma C4.5. Algoritma C4.5 menggunakan *gain ratio* sebagai kriteria *splitting* untuk memilih atribut dengan informasi terpenting dalam membangun pohon keputusan [11]. Berikut merupakan tahapan dari algoritma C4.5 :

a) Langkah pertama yaitu menyeleksi data *training* berdasarkan fitur-fitur yang terpilih pada tahapan seleksi fitur sebelumnya.

b) Selanjutnya hitung *gain ratio* dari setiap atribut. Dalam menentukan *gain ratio,* tentukan lebih dahulu nilai *entropy* total dan *entropy* setiap nilai atribut dengan menggunakan persamaan (6) [4].

$$Entropy\ (S) = \sum_{i=1}^{n} - pi\ * \log_2 pi \quad (6)$$

Keterangan:

S : kumpulan kasus

pi : perbandingan dari Si terhadap S

n : jumlah partisi S

Dilanjutkan dengan perhitungan *information gain* dari setiap atribut dengan persamaan (7).

$$Gain\ (S, A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy\ (Si) \quad (7)$$

Keterangan:

S : kumpulan kasus

A : atribut

|*Si*| : banyak kasus pada partisi ke-i

|*S*| : banyak kasus dalam S

n : banyak partisi atribut A

Lalu lakukan perhitungan *split info* dengan persamaan (2.3) pada tiap atribut.

$$Split\ Info\ (S, A) = -\sum_{i=1}^{n} \frac{S_i}{S} log_2 \frac{S_i}{S} \quad (8)$$

Keterangan:

S : kumpulan kasus

A : Atribut

S_i : banyak sampel pada atribut i

Dilanjutkan dengan perhitungan *gain ratio* dengan persamaan (2.4) pada tiap atribut.

$$Gain\ Ratio\ (a) = \frac{gain(a)}{split(a)} \quad (9)$$

Keterangan:

Gain(a) : Nilai gain

Split(a) : Nilai split

c) Tentukan *root node* atau *node* yang terdapat pada bagian teratas dari pohon keputusan dengan cara mencari atribut yang mempunyai nilai gain ratio tertinggi. Kemudian bangun rule berdasarkan atribut yang terpilih tersebut.

d) Langkah selanjutnya dilakukan kembali perhitungan gain ratio pada semua atribut kecuali atribut yang sudah terpilih atau atribut yang sudah menjadi node pada perulangan sebelumnya. Untuk langkah perhitungan nilai gain ratio sama seperti langkah 2, hanya saja data yang digunakan sudah terseleksi berdasarkan rule yang sudah dibangun sebelumnya.

e) Tentukan nilai gain ratio tertinggi untuk dijadikan internal node atau node dari suatu percabangan. Kemudian bangun rule berdasarkan atribut yang terpilih tersebut.

f) Jika atribut dari internal node belum signifikan menemukan kelas prediksi atau menghasilkan nilai ambigu, maka lakukan kembali langkah 4 dan 5 sampai aturan atau rule yang dibentuk mencapai

kriteria yang ditentukan untuk mencapai kelas prediksi yang signifikan. Jika atribut sudah memenuhi kriteria maka perulangan berhenti dan pohon keputusan telah terbentuk.

2.5. Validasi dan Evaluasi

Pengujian sistem digunakan untuk melihat kinerja dari sistem itu sendiri dalam melakukan tugasnya yaitu mengklasifikasikan dataset *Chronic Kidney Disease* (CKD). Tahapan pengujian memanfaatkan metode *K-fold cross validation* yang membagi data menjadi sepuluh subset yaitu dari *fold* 1 sampai *fold* 10. Proses *training* akan dilakukan secara berulang sebanyak sepuluh kali, dimana setiap pengulangan akan terdapat sembilan *fold* yang dijadikan data latih dan satu *fold* dijadikan data uji.

Untuk mengetahui performa hasil dari sistem, diperlukan sebuah teknik untuk pengukuran evaluasi terhadap kelas aslinya. Satuan ukur evaluasi yang dapat digunakan yaitu *accuracy, precision, recall*, dan *f-measure* dengan memanfaatkan *confusion matrix*. Dalam penentuan kombinasi parameter terbaik pada BPSO menggunakan nilai rata-rata *accuracy* dan waktu komputasi BPSO sebagai satuan ukur evaluasi. Sedangkan dalam skenario klasifikasi algoritma C4.5 tanpa seleksi fitur dan dengan seleksi fitur BPSO menggunakan nilai rata-rata *accuracy, precision, recall*, *f-measure* dan waktu komputasi C4.5 sebagai satuan ukur evaluasi.

3. Hasil dan Pembahasan

Pada penelitian ini, tahapan pengujian dilakukan untuk menentukan jumlah k optimal pada K-Means, menentukan kombinasi parameter terbaik pada BPSO dan mengetahui pengaruh seleksi fitur dengan BPSO pada klasifikasi C4.5. Sehingga diperlukan skenario pengujian untuk mencapai tujuan tersebut. Pada penelitian ini terdapat enam skenario pengujian yang dilakukan.

3.1. Pengujian Jumlah K Optimal

Pengujian ini menentukan jumlah k atau klaster optimal pada atribut yang didiskritisasi dengan menggunakan K-Means *Clustering.* Jumlah klaster yang diujikan yaitu 2, 3, 4 sampai 10 klaster. Penentuan jumlah klaster optimal ditentukan dengan menggunakan metode *Silhouette Coefficient.* Jumlah klaster optimal pada tiap atribut numerik dipilih berdasarkan nilai *Silhouette Coefficient* terbesar. Pada Tabel 1 memperlihatkan hasil perhitungan *Silhouette Coefficient* di masing-masing atribut.

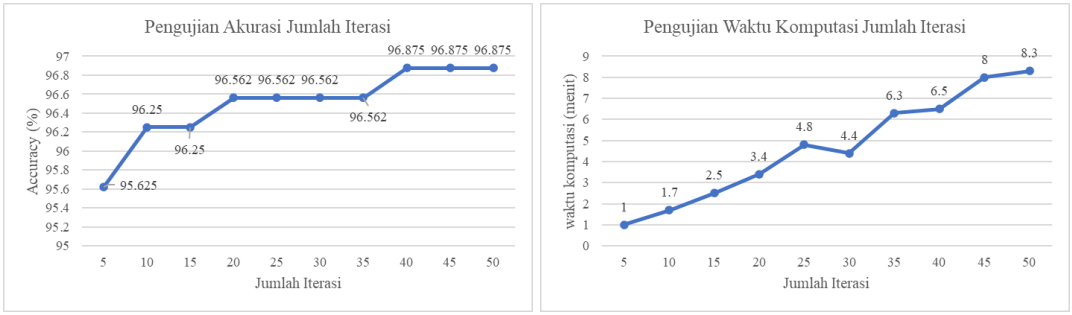
Tabel 1. Hasil Perhitungan *Silhouette Coefficient*

Atribut	Nilai *Silhouette Coefficient* Pada Jumlah Klaster									Klaster Optimal
	2	3	4	5	6	7	8	9	10	
Age	**0,582**	0,559	0,548	0,533	0,539	0,514	0,534	0,539	0,525	2
Bp	0,653	0,694	**0,865**	-0,41	-0,41	-0,08	-0,41	-0,01	-0,27	4
Bgr	**0,745**	0,734	0,575	0,564	0,508	0,577	0,525	0,531	0,549	2
Bu	**0,752**	0,664	0,555	0,589	0,541	0,542	0,563	0,53	0,532	2
Sc	**0,852**	0,798	0,747	0,664	0,517	0,543	0,561	0,538	0,505	2
Sod	0,497	0,523	0,536	0,524	0,474	0,475	0,548	**0,56**	0,557	9
Pot	**0,977**	0,601	0,501	0,453	0,556	0,548	0,554	0,599	0,579	2
Hemo	**0,608**	0,54	0,547	0,529	0,543	0,531	0,573	0,564	0,55	2
Pcv	**0,614**	0,528	0,582	0,548	0,568	0,534	0,584	0,586	0,562	2
Wc	0,574	**0,593**	0,548	0,538	0,547	0,544	0,553	0,554	0,518	3
Rc	0,558	**0,572**	0,53	0,506	0,539	0,533	0,552	0,543	0,556	3

3.2. Pengujian Jumlah Iterasi

Pengujian jumlah iterasi pada BPSO digunakan untuk mendapatkan jumlah iterasi optimal dalam menentukan kombinasi fitur yang optimal. Jumlah iterasi yang diujikan yaitu 5, 10, sampai 50 iterasi dengan dengan jumlah partikel adalah 25, bobot inersia (w) adalah 0,9, c1 yaitu 1 dan c2 yaitu 1,2. Jumlah iterasi terbaik ditentukan berdasarkan nilai rata-rata *accuracy* dari pengujian *10-fold cross validation.* Pada Gambar 2 memperlihatkan hasil pengujian jumlah iterasi.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

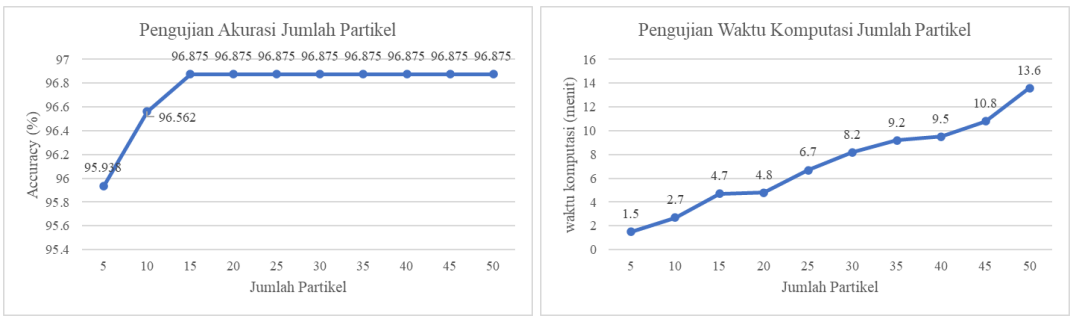


Gambar 2. Pengujian Jumlah Iterasi BPSO

Pada Gambar 2, menunjukan bahwa jumlah iterasi pada BPSO dapat mempengaruhi nilai akurasi yang dihasilkan. Seperti yang terlihat pada pengujian jumlah iterasi dari 5 sampai 40 iterasi mengalami peningkatan nilai akurasi. Hal ini dapat terjadi karena semakin banyak jumlah iterasi yang digunakan maka semakin banyak juga proses pencarian solusi pada BPSO untuk mencapai solusi terbaik. Sedangkan ketika jumlah iterasi terlalu besar akan menghasilkan peningkatan akurasi yang tidak signifikan seperti yang ditunjukan pada hasil pengujian jumlah iterasi 40 sampai 50 iterasi. Hal ini dapat terjadi karena penggunaan jumlah iterasi yang terlalu besar akan mengakibatkan solusi yang dihasilkan hampir sama dengan solusi pada iterasi sebelumnya. Selain nilai akurasi, jumlah iterasi yang digunakan juga dapat mempengaruhi waktu komputasi yang dibutuhkan dalam proses seleksi fitur dengan BPSO. Seperti yang terlihat pada Gambar 2, dimana semakin banyak jumlah iterasi maka waktu komputasi yang dibutuhkan semakin lama. Sehingga dari pengujian jumlah iterasi yang telah dilakukan, 40 iterasi dipilih sebagai jumlah iterasi terbaik dengan nilai akurasi yaitu 96,875% dan waktu komputasi selama 6,5 menit.

3.3. Pengujian Jumlah Partikel

Pengujian ini digunakan untuk memperoleh jumlah partikel optimal dalam menentukan kombinasi fitur yang optimal. Jumlah partikel yang diujikan yaitu 5, 10, sampai 50 partikel dengan bobot inersia (w) yaitu 0,9, c1 yaitu 1, c2 yaitu 1,2 dan jumlah iterasi yang digunakan merupakan jumlah iterasi terbaik pada pengujian sebelumnya yaitu 40 iterasi. Jumlah partikel terbaik ditentukan berdasarkan nilai rata-rata *accuracy* dari pengujian *10-fold cross validation.*

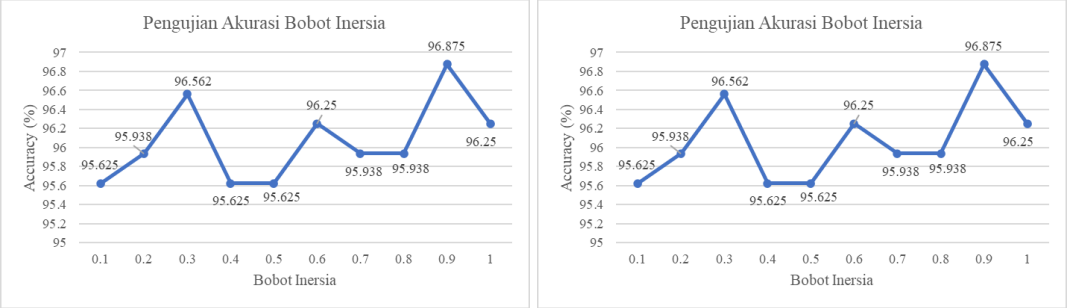


Gambar 3. Pengujian Jumlah Partikel BPSO

Pada Gambar 3, menunjukan bahwa jumlah partikel pada BPSO dapat mempengaruhi hasil akurasi yang didapatkan. Seperti yang terlihat pada hasil pengujian pada jumlah partikel 5 sampai 15 partikel menghasilkan nilai akurasi yang semakin meningkat. Hal ini dapat terjadi karena semakin banyak jumlah partikel yang digunakan maka akan menghasilkan variasi solusi yang lebih beragam sehingga probabilitas dalam menemukan solusi terbaik juga lebih besar. Namun, ketika jumlah partikel yang digunakan terlalu banyak maka nilai akurasi yang dihasilkan tidak mengalami peningkatan yang signifikan seperti yang ditunjukan pada hasil pengujian dengan jumlah partikel 15 sampai 50 partikel. Semakin banyak penggunaan jumlah partikel dapat menghasilkan beberapa partikel akan memiliki solusi yang sama. Selain nilai akurasi, banyaknya jumlah partikel juga dapat mempengaruhi lama waktu komputasi dalam proses seleksi fitur dengan BPSO. Dimana semakin besar jumlah partikel yang digunakan akan memperbesar lama waktu komputasi. Sehingga dari pengujian jumlah iterasi yang telah dilakukan, 15 iterasi dipilih sebagai jumlah partikel terbaik dengan nilai akurasi yaitu 96,875% dan waktu komputasi selama 4,7 menit.

3.4. Pengujian Bobot Inersia

Pengujian ini digunakan untuk memperoleh bobot inersia terbaik dalam menentukan kombinasi fitur yang optimal. Bobot inersia yang diujikan yaitu 0,1, 0,2, sampai 1,0 dengan c1 adalah 1, c2 adalah 1,2 dan jumlah iterasi serta jumlah partikel yang digunakan merupakan nilai terbaik yang didapatkan pada pengujian sebelumnya yaitu 40 partikel dan 15 iterasi. Bobot inersia terbaik ditentukan berdasarkan nilai rata-rata *accuracy* dari pengujian *10-fold cross validation.*



Gambar 4. Pengujian Bobot Inersia BPSO

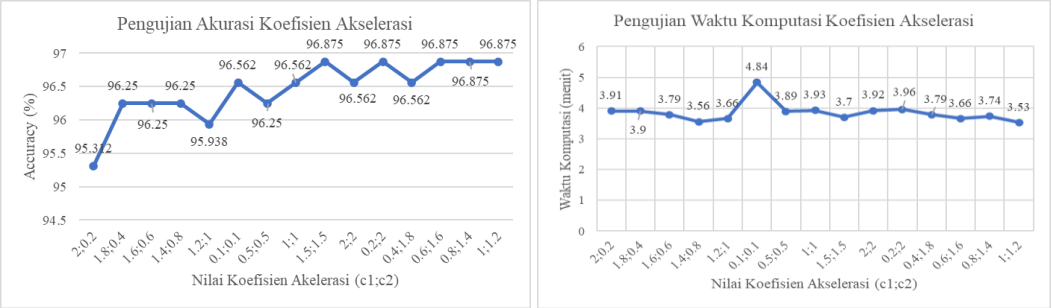
Dari hasil pengujian pada Gambar 4, didapatkan nilai *accuracy* terbesar yaitu 96% pada bobot inersia dengan nilai 0,9. Bobot inersia dapat mempengaruhi nilai akurasi yang didapatkan, dimana semakin banyak bobot inersia akan memperlambat kecepatan partikel pada titik awal pencarian solusi. Sehingga hal ini akan memberikan kesempatan partikel untuk melakukan eksploitasi lokal yang digunakan untuk mencapai solusi terbaik di wilayahnya sendiri sebelum eksplorasi ke wilayah lain. Ketika bobot inersia semakin kecil maka partikel akan cenderung melakukan eksplorasi yang mengakibatkan partikel akan kehilangan kesempatan untuk mencari solusi optimal pada wilayahnya sendiri atau eksploitasi lokal [15]. Selanjutnya dari hasil pengujian pada Gambar 4, peningkatan nilai bobot inersia yang digunakan tidak memberikan peningkatan atau penurunan waktu komputasi secara signifikan. Dari hal tersebut menunjukan bahwa. bobot inersia tidak mempengaruhi lama waktu komputasi yang dibutuhkan.

3.5. Pengujian Koefisien Akselerasi

Pengujian koefisien akselerasi pada BPSO dilakukan untuk mendapatkan nilai faktor *cognitive learning* (c1) dan faktor *social learning* (c2) terbaik dalam menghasilkan fitur-fitur yang paling optimal. Nilai c1 dan c2 yang akan diujikan pada penelitian ini yaitu dengan mengkombinasikan nilai c1 dan c2 pada rentang nilai yaitu 0,1 sampai 2,0. Lalu untuk nilai parameter lainnya yang digunakan merupakan nilai terbaik yang didapatkan pada pengujian sebelumnya yaitu 15 partikel, 40 iterasi serta bobot inersia (w) yaitu 1,2. Nilai c1 dan c2 terbaik ditentukan berdasarkan nilai rata-rata *accuracy* dari pengujian *10-fold cross validation.*

Dari hasil pengujian pada Gambar 5, menunjukan bahwa nilai c1 dan c2 dapat mempengaruhi nilai akurasi yang dihasilkan. Hal tersebut dapat dilihat pada penggunaan nilai c2 yang lebih besar dari nilai c1 memberikan nilai akurasi yang lebih baik dibandingkan penggunaan nilai c1 yang lebih besar dari c2. Hal tersebut dapat terjadi karena, penggunaan nilai c1 yang lebih besar dari c2 mengakibatkan kecepatan partikel akan dominan ke arah *Pbest,* sehingga partikel dengan nilai *fitness* yang rendah hanya akan bergerak pada area *Pbest* dan tidak bergerak menuju *Gbest.* Sedangkan pada penggunaan niali c2 yang lebih besar dari c1 akan mengakibatkan kecepatan partikel akan dominan ke arah *Gbest,* sehingga partikel akan lebih condong bergerak ke arah *Gbest* dibandingkan ke arah *Pbest* [16]. Selain itu dari hasil pengujian, menunjukan bahwa penggunaan nilai c1 dan c2 tidak berdampak secara signifikan terhadap waktu komputasi yang dibutuhkan. Berdasarkan hasil pengujian yang telah dilakukan, nilai akurasi terbesar didapatkan pada kombinasi (1,5;1,5), (0,2;2), (0,6;1,6), (0,4;1,8), (0,8;1,4) dan (1;1,2) dengan nilai akurasi sebesar 96,875%. Untuk menentukan kombinasi c1 dan c2 yang optimal, pemilihan parameter akan dipilih berdasarkan nilai akurasi terbesar dan waktu komputasi terkecil. Sehingga nilai c1 yaitu 1 dan c2 yaitu 1,2 dipilih sebagai kombinasi parameter terbaik dengan nilai akurasi sebesar 96,875% dan waktu komputasi selama 3,53 menit.

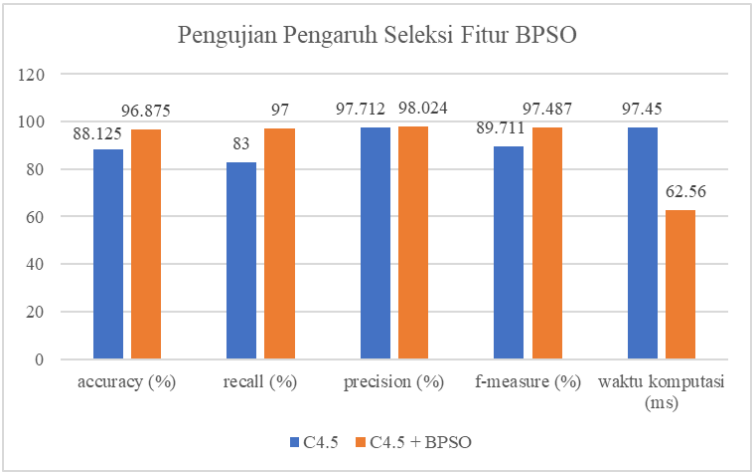
p-ISSN: 2301-5373
e-ISSN: 2654-5101



Gambar 5. Pengujian Koefisien Akselerasi BPSO

3.6. Pengujian Pengaruh Seleksi Fitur BPSO

Pengujian ini digunakan untuk melihat pengaruh seleksi fitur dengan BPSO pada klasifikasi C4.5 dengan membandingkan kinerja dari skema klasifikasi algoritma C4.5 tanpa seleksi fitur dan skema klasifikasi algoritma C4.5 dengan seleksi fitur BPSO. Dimana parameter BPSO yang digunakan merupakan nilai terbaik yang didapatkan pada pengujian sebelumnya yaitu 15 partikel, 40 iterasi, bobot inersia (w) 0,9, nilai c1 yaitu 1 dan c2 yaitu 1,2. Pengaruh seleksi fitur BPSO pada klasifikasi C4.5 dilihat berdasarkan rata-rata nilai *accuracy, precision, recall*, *f-measure* dan waktu komputasi dari pengujian *10-fold cross validation.*



Gambar 6. Pengujian Pengaruh Seleksi Fitur BPSO

Dari hasil pengujian pada Gambar 6, model C4.5 dengan BPSO memberikan hasil evaluasi yang lebih baik dari model C4.5 tanpa BPSO. Model C4.5 dengan BPSO mendapatkan nilai evaluasi yang lebih besar dibandingkan model C4.5 tanpa BPSO dengan nilai rata-rata *accuracy* sebesar 96,875%, *precision* sebesar 97%, *recall* sebesar 96,869% dan *f-measure* sebesar 97,487%. Sedangkan pada model C4.5 tanpa BPSO mendapatkan nilai rata-rata *accuracy* sebesar 88,125%, *precision* sebesar 97,712%, *recall* sebesar 83% dan *f-measure* sebesar 89,711%. Hal ini menunjukan bahwa seleksi fitur dengan BPSO terbukti mampu meningkatkan hasil evaluasi dari klasifikasi algoritma C4.5. Karena penerapan seleksi fitur dengan BPSO dapat menyeleksi data yang kurang informatif dan juga berpengaruh dalam proses pembentukan pohon keputusan serta dapat meminimalisir data *noise.*

Pada Gambar 6 juga terdapat pengujian waktu komputasi yang dihitung berdasarkan berapa lama proses algoritma C4.5 dalam melakukan proses klasifikasi dari tahap *training* sampai *testing*. Dari hasil pengujian yang dilakukan, menunjukan bahwa rata-rata waktu komputasi pada model C4.5 dengan BPSO lebih singkat dibandingkan dengan model C4.5 tanpa BPSO dengan selisih waktu yaitu 34,89 ms. Hal ini terjadi karena fitur-fitur yang diseleksi oleh metode BPSO dapat mengurangi jumlah data yang digunakan dalam proses pembentukan aturan dalam *decision tree*. Sehingga hal ini akan mengurangi beban komputasi pada tahapan *training* algoritma C4.5. Dari hasil pengujian yang telah dilakukan, berdasarkan nilai rata-rata *accuracy, precision, recall*, *f-measure* dan waktu komputasi memperlihatkan metode seleksi fitur BPSO dapat meningkatkan kinerja algoritma C4.5 pada klasifikasi penyakit ginjal kronis.

4. Kesimpulan

Berdasarkan dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Implementasi diskritisasi data dengan K-Means dan seleksi fitur dengan *Binary Particle Swarm Optimization* (BPSO) pada algoritma C4.5 memberikan hasil evaluasi dengan nilai rata-rata *accuracy* sebesar 96,875%*, precision* sebesar 97%*, recall* sebesar 96,869% dan *f-measure* sebesar 97,487%. Penerapan seleksi fitur dengan BPSO mampu meningkatkan kinerja algoritma C4.5 yang terbukti dari hasil evaluasi yang diperoleh pada model C4.5 tanpa BPSO mendapatkan nilai rata-rata *accuracy* sebesar 88,125%*, precision* sebesar 97,712%*, recall* sebesar 83% dan *f-measure* sebesar 89,711%. Hal ini menunjukan bahwa teknik seleksi fitur dengan BPSO pada klasifikasi mampu untuk menghindari data yang bersifat *noise* dalam pembentukan pohon keputusan serta mampu mencari fitur-fitur yang paling informatif. Penerapan seleksi fitur dengan BPSO juga mampu meningkatkan efesiensi waktu komputasi dalam klasifikasi C4.5, dimana rata-rata waktu komputasi pada model C4.5 dengan BPSO lebih singkat dibandingkan dengan model C4.5 tanpa BPSO dengan selisih waktu yaitu 34,89 ms. Hal ini terjadi karena fitur-fitur yang diseleksi dapat mengurangi beban komputasi pada proses pembentukan pohon keputusan.
2. Dari hasil pengujian parameter terbaik pada BPSO, didapatkan bahwa jumlah iterasi terbaik yaitu 40 iterasi dengan *accuracy* sebesar 96,875%, jumlah partikel terbaik yaitu 15 partikel dengan *accuracy* sebesar 96,875%, nilai bobot inersia (w) terbaik yaitu 0,9 dengan *accuracy* sebesar 96,875% dan nilai c1 dan c2 terbaik yaitu 1 dan 1,2 dengan *accuracy* sebesar 96,875%.
3. Berdasarkan hasil pengujian dengan *Silhouette Coefficient* untuk menentukan nilai parameter k optimal pada K-Means *Clustering* dalam proses diskritisasi data bertipe numerik, didapatkan bahwa jumlah k optimal pada atribut *Age, Bp, Bgr, Bu, Sc, So, Po, Hemo, Pcv, Wc,* dan *Rc* berturut-turut yaitu 2, 4, 2, 2, 2, 9, 2, 2, 2, 3 dan 3 klaster.

References

[1] H. Amalia, “Perbandingan Metode Data Mining Svm Dan Nn Untuk Klasifikasi Penyakit Ginjal Kronis,” *Maret*, vol. 14, no. 1, p. 1, 2018, [Online]. Available: www.bsi.ac.id.

[2] I. Fadilla, P. P. Adikara, and R. Setya Perdana, “Klasifikasi Penyakit Chronic Kidney Disease (CKD) Dengan Menggunakan Metode Extreme Learning Machine (ELM),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3397–3405, 2018, [Online]. Available: https://www.researchgate.net/publication/323365845.

[3] E. A. Kurnianto, I. Cholissodin, and E. Santoso, “Klasifikasi Penderita Penyakit Ginjal Kronis Menggunakan Algoritme Support Vector Machine (SVM),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 6597–6602, 2018.

[4] I. Yulianti, R. A. Saputra, M. S. Mardiyanto, and A. Rahmawati, “Optimasi Akurasi Algoritma C4.5 Berbasis Particle Swarm Optimization dengan Teknik Bagging pada Prediksi Penyakit Ginjal Kronis,” *Techno.Com*, vol. 19, no. 4, pp. 411–421, 2020, doi: 10.33633/tc.v19i4.3579.

[5] I. Handayani, “Penyakit Disk Hernia Dan Spondylolisthesis Dalam Kolumna Vertebralis,” vol. 1, no. 2, pp. 83–88, 2019, doi: 10.12928/JASIEK.v13i2.xxxx.

[6] U. Pujianto, A. L. Setiawan, H. A. Rosyid, and A. M. M. Salah, “Comparison of Naïve Bayes Algorithm and Decision Tree C4.5 for Hospital Readmission Diabetes Patients using HbA1c Measurement,” *Knowl. Eng. Data Sci.*, vol. 2, no. 2, p. 58, 2019, doi: 10.17977/um018v2i22019p58-71.

[7] P. S. Oktaviani, R. D. Ramadhani, T. G. Laksana, and A. E. Amalia, “Komparasi Tingkat Akurasi Support Vector Machine (SVM) dan C4.5 dalam Mengklasifikasikan Keberlangsungan Hidup Pasien Hepatitis,” *Centive*, pp. 163–167, 2018.

[8] M. Iskandar, A. Rochman, D. E. Ratnawati, and S. Anam, “Penerapan Algoritme C4 . 5 untuk Klasifikasi Fungsi Senyawa Aktif Menggunakan Kode Simplified Molecular Input Line System (SMILES),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 3, no. 1, pp. 761–769, 2019.

[9] M. K. and J. P. Jiawei Han, “Data Mining: Concepts and Techniques, Third Edition - Books24x7,” *Morgan Kaufmann Publ.*, p. 745, 2012, [Online]. Available: http://library.books24x7.com/toc.aspx?bkid=44712.

[10] R. W. Sembiring Brahmana, F. A. Mohammed, and K. Chairuang, “Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 11, no. 1, p. 32, 2020, doi: 10.24843/lkjiti.2020.v11.i01.p04.

[11] A. C. Pradana and A. Aditsania, “Implementasi Algoritma Binary Particle Swarm Optimization (BPSO) dan C4 . 5 Decision Tree untuk Deteksi Kanker Berdasarkan Klasifikasi Microarray

Data," *e-Proceeding Eng.*, vol. 5, no. 3, pp. 7665–7682, 2018.
[12] A. T. Rahman, Wiranto, and A. Rini, "Coal Trade Data Clustering Using K-Means (Case Study Pt. Global Bangkit Utama)," *ITSMART J. Teknol. dan Inf.*, vol. 6, no. 1, pp. 24–31, 2017, [Online]. Available: https://jurnal.uns.ac.id/itsmart/article/download/11296/11108.
[13] F. Santoso, A. Syukur, and A. Z. Fanani, "Algoritma C4.5 Dengan Particle Swarm Optimization Untuk Klasifikasi Lama Menghafal Al-Quran Pada Santri Mahadul Quran," *J. Teknol. Inf.*, vol. 14, pp. 92–103, 2018.
[14] R. Wajhillah, "OPTIMASI ALGORITMA KLASIFIKASI C4.5 BERBASIS PARTICLE SWARM OPTIMIZATION UNTUK PREDIKSI PENYAKIT JANTUNG," *SWABUMI*, vol. 1, no. 1, pp. 26–36, 2014.
[15] N. A. Sugianto, I. Cholissodin, and A. W. Widodo, "Klasifikasi Keminatan Menggunakan Algoritme Extreme Learning Machine dan Particle Swarm Optimization untuk Seleksi Fitur (Studi Kasus: Program Studi Teknik Informatika FISugianto, N. A., Cholissodin, I., & Widodo, A. W. (2018). Klasifikasi Keminatan Mengg," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 5, pp. 1856–1865, 2018.
[16] K. W. Mahardika, Y. A. Sari, and A. Arwan, "Optimasi K-Nearest Neighbour Menggunakan Particle Swarm Optimization pada Sistem Pakar untuk Monitoring Pengendalian Hama pada Tanaman Jeruk," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 3333–3344, 2018.