

IMPLEMENTATION OF PARTICLE SWARM OPTIMIZATION IN K-NEAREST NEIGHBOR ALGORITHM AS OPTIMIZATION HEPATITIS C CLASSIFICATION

Susi Setianingsih*¹, Maria Ulfa Chasanah², Yogiek Indra Kurniawan³, Lasmedi Afuan⁴

^{1,2,3,4}Informatika, Fakultas Teknik, Universitas Jenderal Soedirman, Indonesia

Email: ¹susi.setianingsih@mhs.unsoed.ac.id, ²maria.chasanah@mhs.unsoed.ac.id, ³yogiek@unsoed.ac.id,
⁴lasmedi.afuan@unsoed.ac.id

(Naskah masuk: 01 April 2023, Revisi: 13 April 2023, Diterbitkan: 20 April 2023)

Abstract

Hepatitis has become a public health problem that is generally caused by infection with the hepatitis virus. One type of hepatitis caused by a virus is Hepatitis C. This disease can cause patients to experience inflammation of the liver. In the worst conditions, it can even lead to death. Initial predictions need to be made to increase the awareness of each individual against the threat of Hepatitis C by using the K-Nearest Neighbor method. K-Nearest Neighbor is a classification method that can give a pretty good percentage result in classifying, especially when using large training data. However, K-Nearest Neighbor still has a weakness, namely the determination of the value of K that is less precise so that it can reduce classification performance. To overcome these shortcomings, the researchers used the implementation of Particle Swarm Optimization on K-Nearest Neighbor to find the optimal K value. The existence of this implementation is expected to be able to increase the value of accuracy in classification and overcome solutions to weaknesses in the K-Nearest Neighbor algorithm. From the results of the K-Nearest Neighbor test, the accuracy value is 97.24% at K=5 and K=3. As for the results of testing the implementation of Particle Swarm Optimization on the K-Nearest Neighbor, there was an increase in the accuracy value of 2.07% to 99.31%. This test shows that the implementation of PSO can overcome the shortcomings of KNN and this model can be used as the best solution to determine the classification of Hepatitis C disease.

Keywords: Classification, Hepatitis, KNN, PSO

IMPLEMENTASI PARTICLE SWARM OPTIMIZATION PADA ALGORITMA K-NEAREST NEIGHBOR UNTUK OPTIMASI PENENTUAN KLASIFIKASI PENYAKIT HEPATITIS C

Abstrak

Penyakit hepatitis sudah menjadi masalah kesehatan masyarakat yang umumnya disebabkan oleh infeksi virus hepatitis. Salah satu jenis hepatitis yang disebabkan oleh virus yaitu Hepatitis C. Penyakit ini dapat menyebabkan penderita mengalami peradangan pada hati. Dalam kondisi terburuk bahkan dapat menyebabkan kematian. Prediksi awal perlu dilakukan guna meningkatkan kewaspadaan setiap individu terhadap ancaman penyakit Hepatitis C dengan menggunakan metode K-Nearest Neighbor. K-Nearest Neighbor merupakan salah satu metode klasifikasi yang mampu memberikan hasil persentase yang cukup bagus dalam melakukan klasifikasi, terutama pada penggunaan data training yang besar. Namun, K-Nearest Neighbor masih memiliki kelemahan yaitu penentuan nilai K yang kurang tepat sehingga dapat menurunkan kinerja klasifikasi. Untuk mengatasi kekurangan tersebut, peneliti menggunakan implementasi Particle Swarm Optimization pada K-Nearest Neighbor untuk mencari nilai K optimal. Adanya implementasi tersebut diharapkan mampu meningkatkan nilai akurasi dalam klasifikasi dan mengatasi solusi untuk kelemahan pada algoritma K-Nearest Neighbor tersebut. Dari hasil pengujian K-Nearest Neighbor diperoleh nilai akurasi 97,24% pada K=5 dan K=3. Adapun untuk hasil pengujian implementasi Particle Swarm Optimization pada K-Nearest Neighbor terjadi peningkatan nilai akurasi sebesar 2,07% menjadi 99,31%. Pengujian ini menunjukkan bahwa implementasi PSO mampu mengatasi kekurangan KNN dan model ini dapat dijadikan sebagai solusi terbaik untuk menentukan klasifikasi penyakit Hepatitis C.

Kata kunci: Hepatitis, Klasifikasi, KNN, PSO

1. PENDAHULUAN

Akhir-akhir ini, dunia sedang digemparkan dengan munculnya wabah virus hepatitis akut yang menyerang anak-anak pada umumnya. Menurut laporan yang dikeluarkan World Health Organization (WHO) pada 17 Mei 2022, setidaknya ada 429 kasus hepatitis akut yang tidak diketahui asalnya telah dilaporkan dari 22 negara di dunia, termasuk 26 kasus yang membutuhkan transplantasi hati [1]. Di Indonesia sendiri, kasus yang sama telah ditemukan. Hal tersebut dilaporkan oleh Kementerian Kesehatan pada 11 Mei 2022 yang menemukan bahwa terdapat 18 orang dengan dugaan kasus hepatitis akut yang belum diketahui penyebabnya dan 7 diantaranya meninggal dunia.

Pada dasarnya hepatitis akut bukanlah suatu fenomena baru dalam dunia medis. Penyakit hepatitis sudah menjadi masalah kesehatan masyarakat yang umumnya disebabkan oleh infeksi virus hepatitis [2]. Selain hepatitis akut yang baru-baru muncul ini, terdapat setidaknya lima jenis hepatitis akut lain seperti Hepatitis A, Hepatitis B, Hepatitis C, Hepatitis D, dan Hepatitis E [3].

Beberapa dari virus hepatitis memiliki tingkat infeksi yang cukup tinggi, salah satunya ialah virus Hepatitis C. Hepatitis C merupakan jenis virus hepatitis yang paling banyak menginfeksi masyarakat Indonesia, nomor dua setelah virus Hepatitis B. Penyakit ini biasanya ditandai dengan suatu peradangan yang terjadi pada organ tubuh seperti hati [4]. Hingga pada tahun 2018, diperkirakan terdapat sekitar 2,5 juta masyarakat Indonesia yang terinfeksi virus Hepatitis C dengan angka kematian yang meningkat dari infeksi tersebut [5]. Oleh karena itu, diperlukan adanya prediksi awal untuk penentuan klasifikasi penyakit Hepatitis C guna meningkatkan kewaspadaan setiap individu terhadap ancaman penyakit tersebut [6].

Pada pembuatan model penentuan klasifikasi, peneliti menggunakan kumpulan dataset pasien hasil diagnosis penyakit Hepatitis C yang diperoleh dari UCI Machine Learning Repository. Dari data tersebut, selanjutnya dilakukan proses *mining* dengan menggunakan sebuah metode klasifikasi. Salah satu teknik yang dapat digunakan untuk proses klasifikasi penyakit Hepatitis C adalah metode K-Nearest Neighbor (KNN).

K-Nearest Neighbor merupakan salah satu metode klasifikasi terhadap sekumpulan data yang berdasarkan mayoritas dari kategori dan tujuannya untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel-sampel dari *training data* [7]. Pemilihan K-Nearest Neighbor sebagai metode dalam penelitian ini tidak lain karena metode ini mampu memberikan hasil persentase yang cukup bagus dalam melakukan klasifikasi, terutama pada penggunaan data training yang besar [8]. Namun, metode ini masih memiliki beberapa kelemahan, diantaranya yaitu kurang optimal dalam menentukan

nilai K dan untuk pemilihan atribut guna mendapatkan hasil terbaik [2], [9], [10].

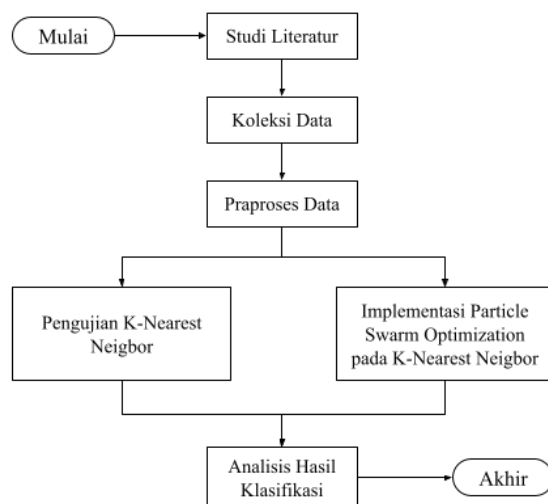
Dalam proses perhitungan K-Nearest Neighbor terdapat parameter K yang dapat mempengaruhi hasil akurasi dari klasifikasi [11]. Permasalahan dalam menentukan nilai K optimal dapat diatasi dengan menggunakan pendekatan metaheuristik karena dapat menghasilkan solusi optimal dari suatu masalah [12]. Pendekatan metaheuristik yang dapat digunakan diantaranya Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), dan Genetic Algorithm (GA) [13].

Pada penelitian ini, dipilih implementasi Particle Swarm Optimization pada K-Nearest Neighbor untuk mencari nilai K optimal. Pemilihan ini didasarkan pada konsep PSO yang sederhana, mudah diimplementasikan, dan memiliki fleksibilitas untuk mengontrol keseimbangan antara pencarian lokal dan global pada ruang pencarian [14]. Alasan lainnya yaitu Particle Swarm Optimization mampu digunakan untuk memecahkan masalah optimasi dan masalah *feature selection* [15].

Adanya implementasi PSO pada pengujian ini diharapkan mampu meningkatkan nilai akurasi dalam klasifikasi dan mengatasi solusi untuk kelemahan pada algoritma K-Nearest Neighbor tersebut.

2. METODE

Metodologi yang dilakukan dalam penelitian ini terdiri dari beberapa tahapan yang ditunjukkan seperti pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1. Studi Literatur

Penelitian diawali dengan melakukan studi literatur terkait topik penelitian yang dilakukan, meliputi *data mining*, klasifikasi, algoritma K-Nearest Neighbor, maupun Particle Swarm Optimization. Pencarian literatur didasarkan pada penelitian-penelitian terdahulu terkait teori yang berkaitan dengan penelitian yang dilakukan dan teori-teori yang tengah berkembang saat ini.

2.2. Koleksi Data

Tahapan koleksi data bertujuan untuk mendapatkan data yang akan digunakan dalam penelitian. Data yang diambil adalah data HCV (Hepatitis C Virus) yang diperoleh dari UCI Machine Learning Repository [16]. Data yang diperoleh berjumlah 615 data yang terdiri dari 14 atribut dengan rincian 12 atribut numerik dan 2 atribut kategorikal (termasuk atribut kelas). Pada dataset ini masih ditemukan *missing value*. Kelas target dari data ini berupa Kategori Blood Donors (BD) dan Hepatitis C (yang di dalamnya termasuk Hepatitis C, Fibrosis, dan Cirrhosis).

2.3. Praproses Data

Praproses data bertujuan untuk mempersiapkan data mentah sebelum dilakukan proses lain yang dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem [17]. Adapun kegiatan yang dilakukan dalam tahap ini yaitu:

- Menghapus atribut yang tidak diperlukan dalam proses penelitian.
- Mengubah tipe data atribut yang belum bertipe numerik menjadi bertipe numerik.
- Menghapus data yang memiliki *missing value*.
- Mengambil data untuk digunakan dalam proses mining.

2.4. Pengujian K-Nearest Neighbor

Tahapan ini merupakan tahapan mining data dengan menggunakan metode KNN. KNN merupakan algoritma yang termasuk dalam kelompok *instance-based learning* dan tergolong dalam teknik *lazy learning* [15]. KNN dilakukan dengan mencari kelompok K objek dalam *training data* yang paling dekat atau mirip dengan objek pada data baru atau data uji [18]. Adapun langkah-langkah dalam menghitung metode KNN antara lain:

- Menentukan parameter K.
- Menghitung jarak antara *training data* dengan *testing data*. Perhitungan yang paling umum digunakan adalah dengan menghitung jarak Euclidean, dengan menggunakan rumus di bawah ini.

$$d_i = \sqrt{\sum_{i=1}^n (p_{2i} - q_{1i})^2} \quad (1)$$

Dimana:

d_i = jarak variabel ke- i

i = variabel data

n = dimensi data

p = data uji

q = data sampel

- Mengurutkan jarak yang terbentuk.
- Menentukan jarak terdekat sampai urutan K.
- Memasangkan kelas yang bersesuaian.

- Mencari jumlah kelas dari tetangga yang terdekat dan menetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.

2.5. Implementasi Particle Swarm Optimization pada K-Nearest Neighbor

Pada tahapan ini dilakukan proses *mining* dengan menggunakan modifikasi KNN yang dikombinasikan dengan PSO. Algoritma PSO merupakan algoritma optimasi yang bertujuan untuk mencari solusi terdekat dengan global minimum atau global maksimum [19]. Masalah optimasi maupun masalah seleksi fitur banyak dipecahkan dengan menggunakan PSO [20].

Langkah-langkah yang dilakukan pada tahapan ini pun pada dasarnya sama dengan langkah yang dilakukan ketika menggunakan metode KNN murni. Akan tetapi terdapat perubahan, yaitu dengan menambahkan algoritma PSO pada proses *mining*-nya, sehingga akan menghasilkan nilai akurasi berbeda yang diharapkan dapat lebih tinggi. Adapun proses perhitungan dari implementasi PSO pada KNN adalah sebagai berikut [2], [21].

- Bangkitkan populasi yang terdiri dari 3 proses yaitu inialisasi kecepatan awal, inialisasi posisi awal, serta inialisasi $Pbest$ dan $Gbest$. Inialisasi dilakukan dengan panjang partikel dalam biner sejumlah N string, N merupakan jumlah ($n1 + n2$). Kemudian hitung *fitness* untuk mendapatkan $Gbest$ seperti pada langkah ke-d. Berikut adalah rumus untuk menghitung masing-masing inialisasi.

- Inialisasi Kecepatan Awal

$$V_{1,j(t)} = 0 \quad (2)$$

$V_{i,j}$ merupakan kecepatan, j adalah letak partikel dan i adalah letak individu dan t adalah iterasi.

- Inialisasi Posisi Awal

$$x_{(t)} = x_{min} + r(x_{max} - x_{min}) \quad (3)$$

X merupakan posisi partikel dan r adalah nilai *random*

- Inialisasi $Pbest$ dan $Gbest$

$$(Pbest_{i,j(t)} = x_{i,j(t)}) \quad (4)$$

$Pbest$ merupakan personal best pada individu ke- i dan partikel ke- j . X_{ij} merupakan posisi partikel. Sedangkan $Gbest$ merupakan $Pbest$ dengan nilai *fitness* terbaik.

- Update* kecepatan untuk menentukan arah perpindahan posisi partikel yang ada di populasi.

$$v_{i,j}^{t+1} = w.v_{i,j}^t + c_1r_1(Pbest_{i,j}^t - x_{i,j}^t) + c_2r_2(Gbest_{g,j}^t - x_{i,j}^t) \quad (5)$$

Dimana:

- $v_{i,j}^{t+1}$ = kecepatan partikel iterasi selanjutnya
- $v_{i,j}^t$ = kecepatan partikel ke saat ini (iterasi ke t)
- w = bobot inersia
- c_1c_2 = koefisien akselerasi
- r_1r_2 = bilangan acak antara 0 dan 1

c. *Update* posisi untuk menentukan posisi terbaru dari setiap partikel berdasarkan hasil *update* kecepatan sebelumnya.

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{6}$$

$x_{i,j}^{t+1}$ merupakan posisi partikel hasil *update*.

- d. Hitung *fitness* (evaluasi akurasi KNN)
Hitung nilai *fitness* setiap individu pada populasi, menggunakan nilai akurasi KNN berdasarkan rata-rata dari nilai klasifikasi yang benar atau sesuai dengan data asli dan berdasarkan nilai *k* dari setiap individu.
- e. *Update Pbest* yaitu dengan membandingkan nilai *fitness* dari *Pbest* pada iterasi sebelumnya dengan *fitness* dari *update Posisi*. Nilai yang terbaik akan menjadi *Pbest* yang baru pada iterasi selanjutnya.
- f. *Termination test*
Ulangi langkah b sampai e hingga kondisi berhenti terpenuhi.

2.6. Analisis Hasil Klasifikasi

Pada tahapan ini dilakukan analisis dan pembahasan terhadap hasil *mining* yang telah dilakukan. Analisis pertama yang dilakukan adalah mengenai nilai K yang digunakan pada metode KNN. Analisis selanjutnya adalah mengenai perbandingan nilai akurasi, presisi, dan *recall* antara penggunaan metode KNN murni dan penggunaan modifikasi KNN yang dikombinasikan dengan PSO.

Pengertian akurasi (*accuracy*) dalam klasifikasi merupakan persentase ketepatan *record* data yang diklasifikasikan dengan benar. Sementara itu, presisi (*precision*) merupakan proporsi kasus yang diprediksi positif yang juga positif benar pada data sebenarnya. Adapun *recall* merupakan proporsi kasus positif yang sebenarnya diprediksi positif secara benar.

Nilai akurasi, presisi, dan *recall* dapat diperoleh dengan menggunakan metode Confusion Matrix [22]. Metode Confusion Matrix merepresentasikan hasil evaluasi model dengan menggunakan tabel matriks dimana kelas prediksi akan dibandingkan dengan kelas asli [23]. Tabel 1 berikut menyajikan Confusion Matrix untuk klasifikasi 2 kelas.

Tabel 1. Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Perhitungan nilai akurasi, presisi, dan *recall* berdasarkan Tabel 1 ditunjukkan pada Persamaan (7), (8), dan (9) berikut.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \tag{7}$$

$$precision = \frac{TP}{TP+FP} \times 100\% \tag{8}$$

$$recall = \frac{TP}{TP+FN} \times 100\% \tag{9}$$

Keterangan:

- TP : True Positive
- TN : True Negative
- FP : False Positive
- FN : False Negative

Terakhir, akan diambil kesimpulan dari hasil analisis penelitian yang telah dilakukan.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Praproses Data

Setelah melakukan tahapan praproses data, dihasilkan sebuah set data yang terdiri dari 589 data bersih dengan 13 atribut yang bertipe numerik. Di dalam penelitian ini, peneliti memutuskan untuk menggunakan 580 data yang akan digunakan dalam proses *mining*. Atribut hasil praproses data dapat dilihat pada Tabel 2, sedangkan hasil praproses data dapat dilihat pada Tabel 3.

Tabel 2. Atribut Hasil Praproses Data

Simbol	Atribut	Tipe	Informasi
Y	Category	Integer	0=BD, 1=Hepatitis C
X1	Age	Integer	Umur (dalam tahun)
X2	Sex	Integer	0=laki-laki, 1=perempuan
X3	ALB	Real	Data laboratorium
X4	ALP	Real	Data laboratorium
X5	ALT	Real	Data laboratorium
X6	AST	Real	Data laboratorium
X7	BIL	Real	Data laboratorium
X8	CHE	Real	Data laboratorium
X9	CHOL	Real	Data laboratorium
X10	CREA	Integer	Data laboratorium
X11	GGT	Real	Data laboratorium
X12	PROT	Real	Data laboratorium

Tabel 3. Hasil Praproses Data

No	Y	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
1	0	32	0	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106	12.1	69
2	0	32	0	38.5	70.3	18	24.7	3.9	11.17	4.8	74	15.6	76.5
3	0	32	0	46.9	74.7	36.2	52.6	6.1	8.84	5.2	86	33.2	79.3
4	0	32	0	43.2	52	30.6	22.6	18.9	7.33	4.74	80	33.8	75.7
5	0	32	0	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76	29.9	68.7
6	0	32	0	41.6	43.3	18.5	19.7	12.3	9.92	6.05	111	91	74
7	0	32	0	46.3	41.3	17.5	17.8	8.5	7.01	4.79	70	16.9	74.5
8	0	32	0	42.2	41.9	35.8	31.1	16.1	5.82	4.6	109	21.5	67.1
9	0	32	0	50.9	65.5	23.2	21.2	6.9	8.69	4.1	83	13.7	71.3
10	0	32	0	42.4	86.3	20.3	20	35.2	5.46	4.45	81	15.9	69.9
...													
579	1	64	1	24	102.8	2.9	44.4	20	1.54	3.02	63	35.9	71.3
580	1	64	1	29	87.3	3.5	99	48	1.66	3.63	66.7	64.2	82

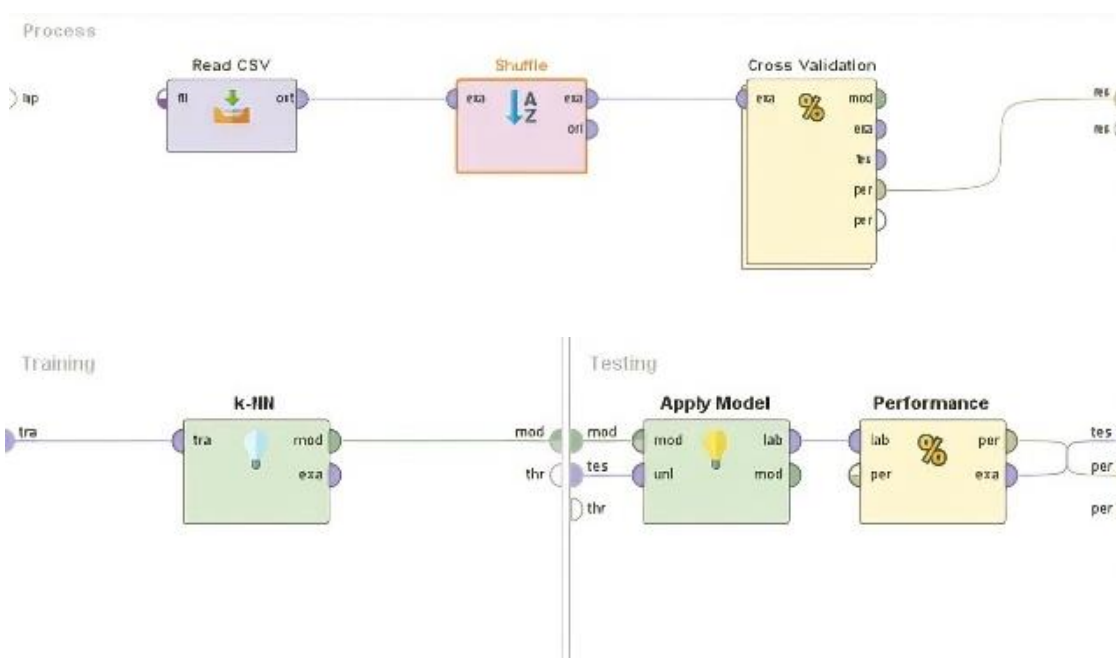
Pada Tabel 2, simbol Y merupakan atribut kategori untuk klasifikasi hepatitis C. Sementara itu, simbol X menunjukkan atribut yang menentukan hasil klasifikasi simbol Y. Selain itu, pada Tabel 3 ditampilkan data hasil praproses yang akan digunakan selama proses pengujian.

3.2. Proses Klasifikasi dengan Pengujian K-Nearest Neighbor

Setelah melalui praproses data, kemudian dilakukan pengujian dengan menggunakan metode K-Nearest Neighbor. Sebelum melakukan pengujian, harus ditentukan model validasi yang digunakan. Adapun model validasi yang dapat digunakan pada penelitian ini adalah dengan model pengujian Cross Validation.

Cross Validation pada dasarnya digunakan untuk memprediksi model dan memperkirakan seberapa akurat sebuah model prediktif dijalankan [15]. Teknik dari Cross Validation yang akan dipakai adalah K-Fold Cross Validation dengan nilai K=10. Nantinya, dataset akan secara acak dibagi menjadi 10 bagian dengan ukuran yang sama. Selama 10 kali, 9 bagian akan menjadi *training data* dan 1 bagian akan menjadi *testing data* setiap kali dilakukan pengujian.

Sementara itu, untuk mengetahui nilai K maksimal pada K-Nearest Neighbor, peneliti menggunakan nilai 3, 5, 7, 9, 11, 13, 15, 17, 19, dan 21 sebagai bahan uji. Pembuatan model KNN dilakukan dengan menggunakan aplikasi Rapidminer dapat dilihat pada Gambar 2.



Gambar 2. Pemodelan KNN dengan Aplikasi Rapidminer

Pada Gambar 2, model KNN dibuat dengan beberapa operator. Operator Read CSV digunakan untuk membaca ExampleSet dari file csv yang telah ditentukan. Shuffle digunakan untuk mengacak ExampleSet dan membuat salinan baru di memori. Cross Validation digunakan untuk melakukan validasi pengujian. Di dalam operator Cross Validation terdapat Training View yang berisi operator KNN dan Testing View yang berisi operator Apply Model dan Performance. Adapun operator KNN merupakan operator yang menghasilkan model klasifikasi KNN. Sementara itu, Apply Model berfungsi menerapkan model pada ExampleSet sedangkan Performance digunakan untuk evaluasi kerja.

Berdasarkan pengujian dengan K-Nearest Neighbor, diperoleh nilai akurasi, presisi, dan recall untuk setiap nilai K. Hasilnya ditunjukkan pada Tabel 4 berikut.

Tabel 4. Hasil Pengujian KNN

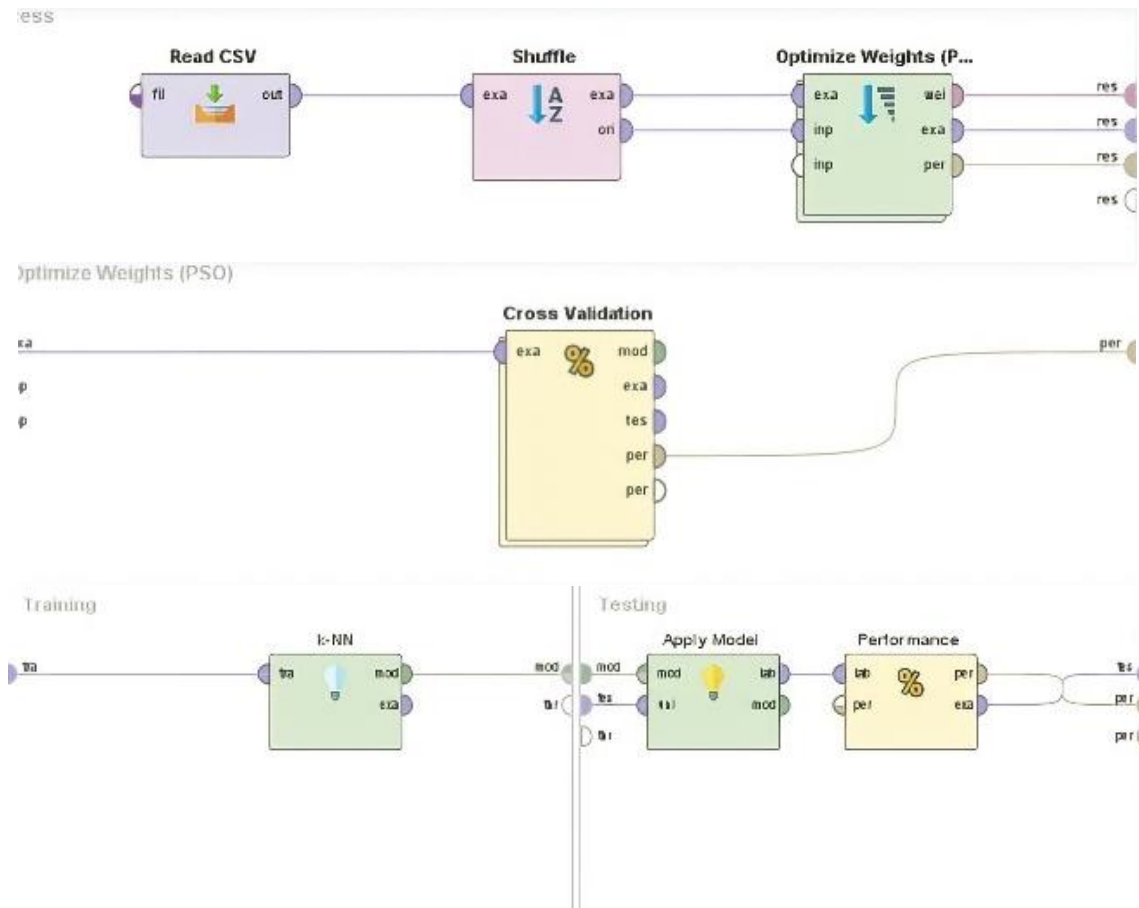
Uji	K	Accuracy	Precision	Recall
1	3	97.24%	97.74%	99.24%
2	5	97.24%	97.57%	99.43%
3	7	96.90%	97.03%	99.62%
4	9	96.21%	96.48%	99.43%
5	11	96.55%	96.49%	99.81%
6	13	96.03%	95.96%	99.81%
7	15	96.03%	95.96%	99.81%
8	17	95.86%	95.79%	99.81%
9	19	95.69%	95.61%	99.81%
10	21	95.17%	95.09%	99.81%

Pada Tabel 4 menunjukkan hasil pengujian dengan KNN dimana akurasi tertinggi mencapai 97.24% pada K=3 dan K=5. Sementara itu, presisi tertinggi dihasilkan oleh K=3 dengan nilai 97.74%. Nilai recall tertinggi yaitu 99.81% pada K yang bernilai 11, 13, 15, 17, 19, dan 21.

3.3. Proses Klasifikasi Menggunakan Particle Swarm Optimization pada K-Nearest Neighbor

Pada pengujian dengan implementasi PSO pada KNN memiliki proses yang sama dengan pengujian KNN murni. Perbedaannya hanya terletak pada penggunaan PSO selama proses pengujian. Sebelum memulai pengujian, parameter-parameter dalam PSO perlu ditentukan terlebih dahulu.

Dalam pengujian ini, peneliti menggunakan parameter *default* diantaranya yaitu Population Size = 5, Max of Generation = 30, Inersia Weight = 1.0, Local Best Weight = 1.0, dan Global Best Weight = 1.0. Pembuatan model PSO + KNN dilakukan dengan menggunakan aplikasi Rapidminer dapat dilihat pada Gambar 3 berikut.



Gambar 3. Pemodelan Implementasi PSO pada KNN dengan Aplikasi Rapidminer

Pada Gambar 2, pemodelan implementasi PSO pada KNN dengan aplikasi RapidMiner menggunakan operator yang hampir sama seperti pada pengujiannya sebelumnya. Hanya saja pada pengujian ini menambahkan operator Optimize Weights (PSO) yaitu metode pengoptimalan yang menggunakan sekumpulan partikel untuk menghasilkan bobot atribut. Dalam operator ini terdapat Optimize Weights (PSO) View yang berisi operator Cross Validation dengan operator KNN, Apply Model, dan Performamance didalamnya. Model operator Cross Validation ini memiliki bentuk yang sama seperti model pengujian KNN murni.

Berdasarkan pengujian dengan implementasi PSO pada KNN, diperoleh nilai akurasi, presisi, dan *recall* untuk setiap nilai K. Hasilnya ditunjukkan pada Tabel 5 berikut.

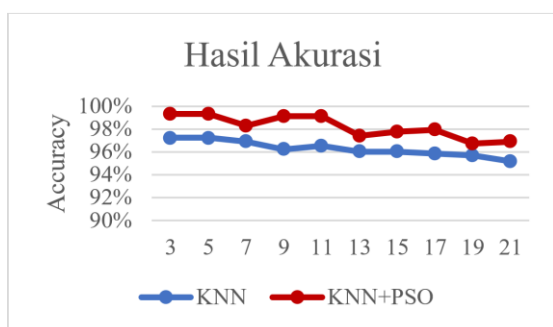
Tabel 5. Hasil Pengujian implementasi PSO pada KNN

Uji	K	Accuracy	Precision	Recall
1	3	99.31%	99.62%	99.62%
2	5	99.31%	99.62%	99.62%
3	7	98.28%	98.31%	99.81%
4	9	99.14%	99.43%	99.62%
5	11	99.14%	99.43%	99.62%
6	13	97.41%	97.57%	99.62%
7	15	97.76%	97.94%	99.62%
8	17	97.93%	97.94%	99.81%
9	19	96.72%	97.02%	99.43%
10	21	96.90%	97.03%	99.62%

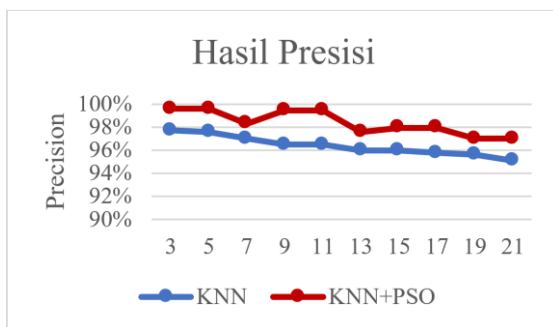
Pada Tabel 5 menunjukkan hasil pengujian dengan implementasi PSO pada KNN dimana akurasi tertinggi mencapai 99.31% pada K=3 dan K=5. Sementara itu, presisi tertinggi dihasilkan oleh K yang sama yang memiliki akurasi tertinggi dengan nilai 99.62%. Nilai *recall* tertinggi yaitu 99.81% pada K=7 dan K=17.

3.4. Analisis Hasil Klasifikasi

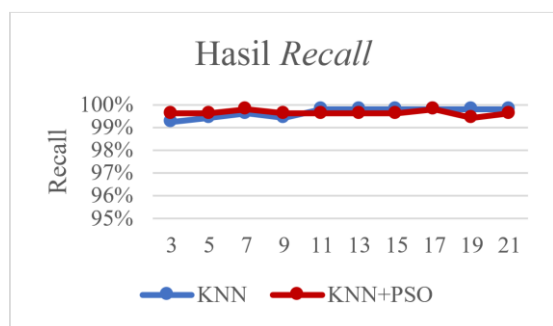
Hasil dari pengujian antara penggunaan K-Nearest Neighbor dengan implementasi PSO pada KNN selanjutnya dibandingkan. Perbandingan hasil akurasi, presisi, dan *recall* pada KNN dengan KNN yang diimplementasikan dengan menggunakan PSO ditunjukkan pada Gambar 4, Gambar 5, dan Gambar 6 berikut.



Gambar 4. Grafik Hasil Akurasi



Gambar 5. Grafik Hasil Presisi



Gambar 6. Grafik Hasil Recall

Berdasarkan Gambar 4, Gambar 5, dan Gambar 6 terlihat bahwa hasil yang diperoleh menunjukkan adanya peningkatan nilai akurasi, presisi, dan *recall* di setiap pengujian yang dilakukan setelah diimplementasikan menggunakan algoritma PSO. Nilai dengan akurasi dan presisi terbaik setelah diimplementasikan menggunakan algoritma PSO ditunjukkan oleh nilai K=3 dan K=5 dengan nilai yang sama masing-masing adalah 99,31% dan 99,62%. Sementara itu, nilai *recall* yang tertinggi mencapai 99,81% dengan nilai K dari uji 5 sampai uji 10 pada KNN serta uji 3 dan uji 8 pada KNN+PSO.

4. DISKUSI

Penggunaan PSO untuk meningkatkan tingkat akurasi memiliki dampak yang cukup signifikan. Hasil ini berkesesuaian dengan penelitian yang pernah dilakukan sebelumnya. Seperti penelitian yang dilakukan oleh Mahardika, dkk. (2018) tentang Optimasi K-Nearest Neighbour Menggunakan Particle Swarm Optimization pada Sistem Pakar untuk Monitoring Pengendalian Hama pada Tanaman Jeruk. Pada penelitian ini, terjadi peningkatan nilai akurasi dari 90% ke 96,25% setelah KNN diimplementasikan dengan PSO [2]. Pada penelitian yang lain mengenai Kombinasi KNN dan Particle Swarm Optimization (PSO) pada Prediksi Kualitas Udara oleh Yahdin, dkk. (2022) menunjukkan peningkatan nilai akurasi setelah adanya implementasi PSO yaitu dari 97.12% ke 98.98% [24]. Dari kedua penelitian tersebut menunjukkan bahwa optimasi KNN dengan menggunakan PSO merupakan salah satu alternatif yang cukup baik untuk meningkatkan akurasi dari hasil klasifikasi. Oleh karena itu, peneliti mencoba menguji model klasifikasi dengan dan tanpa implementasi PSO pada KNN untuk studi kasus Hepatitis C. Analisis yang akan dilakukan pada penelitian ini meliputi perbandingan nilai akurasi, presisi, dan *recall* tertinggi hasil setiap pengujian yang dilakukan.

5. KESIMPULAN

Perbedaan nilai akurasi, presisi, dan *recall* di setiap pengujian dengan nilai K berbeda menunjukkan bahwa nilai K memiliki pengaruh terhadap akurasi KNN. Semakin tinggi nilai K maka akurasi cenderung akan menurun, yang disebabkan oleh semakin banyak tetangga yang digunakan dalam proses klasifikasi sehingga *noise* semakin tinggi. Ditambah lagi dengan adanya dominasi atau frekuensi kelas *training data* yang tidak seimbang dari suatu kelas tertentu sehingga data yang dihasilkan cenderung diklasifikasikan pada data yang mendominasi.

Dalam penelitian ini ditemukan bahwa nilai K optimal untuk akurasi adalah 3 dan 5 dengan hasil nilai akurasi menggunakan KNN adalah sama yaitu sebesar 97,24%. Adanya implementasi PSO dapat meningkatkan kinerja KNN dengan menghasilkan nilai akurasi yang lebih baik dibandingkan dengan pengujian KNN murni. Hal ini dapat ditunjukkan dengan adanya peningkatan nilai akurasi pada K=3 dan K=5 sebesar 2,07% menjadi 99,31%. Sementara itu, nilai presisi juga mengalami peningkatan pada K=3 dan K=5 yaitu dari masing-masing 97.74% dan 97.57% menjadi 99.62%. Adapun nilai *recall* tertinggi selama pengujian yaitu sebesar 99,81%. Dari hasil pengujian dapat disimpulkan bahwa implementasi PSO mampu mengatasi kekurangan KNN dan model ini dapat dijadikan sebagai solusi terbaik untuk menentukan klasifikasi penyakit Hepatitis C.

DAFTAR PUSTAKA

- [1] W. H. Organization, "Acute, severe hepatitis of unknown origin in children," 2022.
- [2] K. Wiliam 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. Informassi dan Ilmu Komput.*, vol. 2, no. 9, pp. 3333–3344, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [3] Siswanto, *Epidemiologi Penyakit Hepatitis*. Mulawarman University Press, 2020.
- [4] Alhawaris, "Hepatitis C: Epidemiologi, Etiologi, dan Patogenitas," *J. Sains dan Kesehat.*, vol. 2, no. 2, pp. 139–150, Dec. 2019, doi: 10.25026/jsk.v2i2.132.
- [5] S. Jonathan, W. Jack, and W. Suzanne, "Situation Analysis of Viral Hepatitis in Indonesia : A Policy Report," 2018. [Online]. Available: http://www.healthpolicypartnership.com/wp-content/uploads/hepatitis/Situation_analysis_of_viral_hepatitis_in_Indonesia.pdf
- [6] S. Sulastri, K. Hadiono, and M. T. Anwar, "Analisis Perbandingan Klasifikasi Prediksi Penyakit Hepatitis Dengan Menggunakan Algoritma K-Nearest Neighbor, Naïve Bayes Dan Neural Network," *J. Din.*, vol. 24, no. 2, pp. 82–91, 2020, doi: 10.35315/dinamik.v24i2.7867.
- [7] P. Putra, A. M. H Pardede, and S. Syahputra, "ANALISIS METODE K-NEAREST NEIGHBOUR (KNN) DALAM KLASIFIKASI DATA IRIS BUNGA," *J. Tek. Inform. Kaputama*, vol. 6, no. 1, 2022.
- [8] M. Kumari and S. Soni, "A Review of classification in Web Usage Mining using K-Nearest Neighbour," 2017. [Online]. Available: <http://www.ripublication.com>
- [9] W. Nugroho, "Optimasi Metode K-Nearest Neighbours dengan Backward Elimination Menggunakan Dataset Software Effort Estimation Bianglala Informatika," *Bianglala Inform.*, vol. 8, no. 2, pp. 129–133, 2020.
- [10] A. Bode, "K-Nearest Neighbor Dengan Feature Selection Menggunakan Backward Elimination Untuk Prediksi Harga Komoditi Kopi Arabika," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 188–195, 2017, doi: 10.33096/ilkom.v9i2.139.188-195.
- [11] D. Untuk, M. Skripsi, T. Zadlyka, and J. Teknik Informatika, "Optimasi Metode K-Nearest Neighbor (KNN) Menggunakan Particle Swarm Optimization (PSO) untuk Diagnosis Penyakit Hati. Oleh," 2021.
- [12] H. Febriyanto, "PENERAPAN ALGORITMA PARTICLES SWARM OPTIMIZATION DALAM PENYELESAIAN GATE ASSIGNMENT PROBLEM (STUDI KASUS : BANDARA SOEKARNO-HATTA)," Institut Teknologi Sepuluh Nopember, 2018.
- [13] K. Sasirekha and K. Thangavel, "Optimization of K-nearest neighbor using particle swarm optimization for face recognition," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7935–7944, 2019, doi: 10.1007/s00521-018-3624-9.
- [14] I. Assayyis, I. Cholissodin, and Tibyani, "Optimasi Travelling Salesman Problem Pada Angkutan Sekolah Menggunakan Algoritme Genetika (Studi Kasus : Sekolah MI Salafiyah Kasim Blitar)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 3, no. 1, pp. 454–461, 2020.
- [15] W. Yunus, "Algoritma K-Nearest Neighbor Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Ginjal Kronik," *J. Tek. Elektro CosPhi*, vol. 2, no. 2, pp. 51–55, 2018.
- [16] R. Lichtinghagen, F. Klawonn, and G. Hoffmann, *Hepatitis C Prediction Dataset Laboratory values of blood donors and Hepatitis C patients*. 2020. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/HCV+data>
- [17] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.
- [18] D. Cahyanti, A. Rahmayani, and S. A. Husniar, "Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 39–43, 2020, doi: 10.33096/ijodas.v1i2.13.
- [19] T. Hidayat, A. F. Habibi, and U. L. Yuhana, "Software Defect Prediction Menggunakan Algoritma K-Nn Yang Dioptimasi Dengan Pso," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 15, no. 1, pp. 16–21, 2020, doi: 10.33005/scan.v15i1.1848.
- [20] D. Pajri, Y. Umaidah, and T. N. Padilah, "K-Nearest Neighbor Berbasis Particle Swarm Optimization untuk Analisis Sentimen Terhadap Tokopedia," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 242–253, 2020, doi: 10.28932/jutisi.v6i2.2658.
- [21] R. D. Liklikwatil, E. Noersasongko, C. Supriyanto, S. Dipanegara, and U. D.

- Nuswantoro, "Optimasi K-Nearest Neighbor Dengan Particle Swarm Optimization Untuk Memprediksi Harga Komoditi Karet," 2018.
- [22] T. A. Yoga and Prihandoko, "Penerapan Optimasi Berbasis Particle Swarm Optimization (Pso) Algoritma Naïve Bayes Dan K-Nearest Neighbor Sebagai Perbandingan Untuk Mencari Kinerja Terbaik Dalam Mendeteksi Kanker Payudara," *J. Bangkit Indones.*, vol. 7, no. 2, p. 1, 2018, [Online]. Available: <http://journal.universitasmulia.ac.id/index.php/metik/article/view/62>
- [23] V. Cherian and M. S. Bindu, "Heart Disease Prediction Using Naïve Bayes Algorithm and Laplace Smoothing Technique," *Int. J. Comput. Sci. Trends Technol.*, vol. 5, no. 2, pp. 68–73, 2017.
- [24] S. Yahdin *et al.*, "COMBINATION OF KNN AND PARTICLE SWARM OPTIMIZATION (PSO) ON AIR QUALITY PREDICTION," *J. Ilmu Mat. dan Terap.*, vol. 16, no. 1, pp. 7–14, 2022.