

Klasifikasi Penyakit Ginjal Kronis (CKD) dengan Algoritma KNN dan Decision Tree ID3

Maulana Fikri Ahmadi-1, Nabil Fahlevi Abdi-2

Teknik Informatika

Universitas Pancasila, Jakarta Selatan 4520210062@univpancasila.ac.id-1, 4520210072@univpancasila.ac.id-2

Abstract— Penyakit Ginjal Kronis merupakan masalah kesehatan global yang memerlukan diagnosis untuk pencegahan komplikasi. Menurut Direktur Pencegahan dan Pengendalian Penyakit Tidak Menular Kemenkes RI, di Indonesia sendiri, Penyakit Gagal Ginjal Kronis menjadi penyebab kematian urutan ke-10 dengan angka kematian lebih dari 42 ribu per tahun. Gagal ginjal kronis adalah kondisi ketika fungsi ginjal menurun secara bertahap. Penyakit gagal ginjal bisa terjadi karena berbagai faktor, termasuk hipertensi, diabetes, penyakit autoimun, infeksi ginjal, serta batu ginjal yang tidak diobati secara tepat. Langkah yang dapat digunakan untuk pencegahan yaitu dengan identifikasi penyakit dengan klasifikasi data mining. Banyak metode yang sudah digunakan untuk memprediksi penyakit gagal ginjal kronis, diantaranya adalah metode K-Nearest Neighbor (KNN) & Decision Tree ID3. Pada penelitian ini dilakukan klasifikasi menggunakan metode KNN dan ID3 dengan menguji data dengan berbagai persentase data uji, yaitu 10%, 20%, 30% dan 40%. Setelah dilakukan pengujian, hasil perhitungan tertinggi dari metode KNN terdapat pada data uji persentase 30% dengan nilai $k=3$, akurasi yang didapat mencapai 99.16%. Sedangkan pada metode Decision tree ID3, nilai akurasi tertinggi terdapat pada persentase data uji 30% dengan nilai akurasi 98.33%.

Kata Kunci : agal Ginjal Kronis; Klasifikasi; K-Nearest Neighbor; Decision Tree ID3

1. PENDAHULUAN

Ginjal merupakan organ vital di dalam tubuh manusia. Ginjal berfungsi sebagai organ ekskresi yang menyaring kotoran dari darah dan dikeluarkan dalam bentuk air bersama dengan urin. Jika ginjal mengalami gangguan maka tubuh akan terserang penyakit karena cairan metabolisme tubuh tidak dibersihkan oleh ginjal. Salah satu gangguan yang dapat dialami ginjal adalah gagal ginjal. Penyakit Gagal Ginjal Kronis (Chronic Kidney Disease) merupakan penyakit dimana ginjal tidak dapat berfungsi dengan baik. Penyakit ini dapat disebabkan oleh beberapa faktor, peneliti lain yang relevan dengan penelitian yang sedang dilakukan, diantaranya hipertensi, diabetes, maupun infeksi pada ginjal. Gejala yang ditimbulkan penyakit ini muncul secara bertahap, namun seringkali gejala tersebut tidak terlihat dengan jelas. Oleh karena itu, perlu identifikasi yang akurat untuk memprediksi penyakit ginjal. Teknik yang dapat

digunakan untuk mengidentifikasi penyakit gagal ginjal adalah dengan klasifikasi data mining.

Pada penelitian ini, akan dilakukan klasifikasi penyakit gagal ginjal kronis dengan metode K-Nearest Neighbor dan Decision tree ID3. Tujuan penelitian ini adalah untuk mengetahui tingkat akurasi dari kedua metode dan menentukan metode yang lebih baik digunakan untuk memprediksi penyakit gagal ginjal kronis.

a. Penyakit Gagal Ginjal Kronis

Penyakit Gagal Ginjal Kronis adalah kondisi ketika fungsi ginjal menurun secara bertahap akibat kerusakan jaringan ginjal yang dipicu oleh penyakit jangka panjang seperti diabetes, hipertensi, dan infeksi ginjal.

b. Klasifikasi

Klasifikasi merupakan proses pengelompokkan data ke dalam kategori atau kelas tertentu berdasarkan ciri-ciri atau atribut yang dimiliki oleh data tersebut. Klasifikasi data dapat dilakukan dengan menggunakan berbagai metode, seperti K-Nearest Neighbor dan Decision Tree. Tujuan dari klasifikasi data adalah untuk memudahkan analisis dan pengambilan keputusan berdasarkan data yang telah dikelompokkan ke dalam kategori atau kelas tertentu.

c. K-Nearest Neighbor

K-Nearest Neighbor adalah algoritma klasifikasi terhadap sekumpulan data yang termasuk dalam algoritma supervised learning dimana hasil dari query instance yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori. Dengan kata lain, Algoritma K-Nearest Neighbor adalah algoritma yang melakukan klasifikasi berdasarkan kedekatan jarak suatu data dengan data yang lain.

Secara garis besar, algoritma K-Nearest Neighbor adalah sebagai berikut.

- Menentukan parameter k (jumlah tetangga paling dekat).
- Menghitung jarak antara data yang akan dievaluasi terhadap data training yang diberikan.
- Mengurutkan hasil langkah ke-2 secara ascending (urut naik) dan tentukan jarak terdekat sampai urutan ke k

- Pasangkan kelas yang bersesuaian
- Cari jumlah kelas terbanyak dari tetangga terdekat tersebut dan tetapkan kelas tersebut sebagai kelas data yang dievaluasi.

d. Decision Tree ID3

Decision Tree ID3 adalah algoritma klasifikasi menggunakan pohon keputusan. Algoritma ini melakukan pencarian secara menyeluruh pada semua kemungkinan pohon keputusan. Pembentukan pohon klasifikasi dengan algoritma ID3 melalui dua langkah, yaitu menghitung nilai entropy dan menghitung nilai information gain dari setiap variabel.

2. METODE PENELITIAN

Dalam penelitian ini dilakukan percobaan menggunakan dataset dari *Kaggle.com*. Dataset tersebut digunakan untuk *latih* dan *uji* data yang berfungsi mencari tingkat akurasi metode K-Nearest Neighbour. Dalam menentukan prediksi gagal ginjal kronis peneliti menggunakan bahasa pemrograman Python serta beberapa library seperti NumPy, Pandas, dan Scikit-learn untuk mengimplementasikan metode KNN.

a. Pengumpulan Data

Pada penelitian ini dataset diperoleh dari internet melalui laman website

<https://www.kaggle.com/datasets/abhia1999/chronic-kidney-disease>. Dari dataset tersebut terdapat 400 data, 13 fitur, dan 2 kelas yang ditampilkan pada tabel.1 dan tabel.2 berikut.

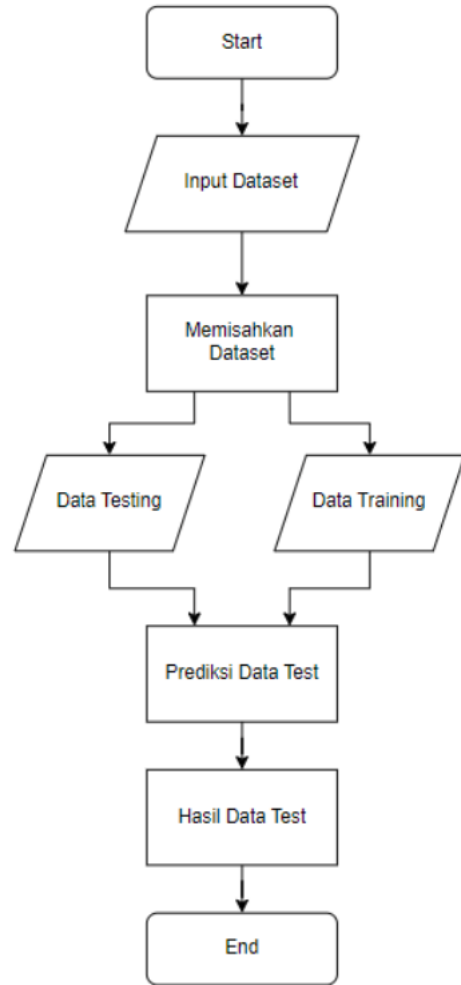
Tabel.1 Fitur Dataset

No	Fitur	Deskripsi	Values
1	Bp	Blond Pressure	Float64
2	Sg	Specific Gravity	Float64
3	Al	Album	Float64
4	Su	Sugar	Float64
5	Rbc	Red Blood Cell	Float64
6	Bu	Blood Urea	Float64
7	Sc	Serum Creatine	Float64
8	Sod	Sodium	Float64
9	Pot	Potassium	Float64
10	Hemo	Hemoglobin	Float64
11	Wbcc	White Blood Count Cell	Float64
12	Rbcc	Red Blood Count Cell	Float64
13	Htn	Hypertension	Float64

Tabel.1 Kelas Dataset

No	Kelas	Deskripsi	Values
1	Ckd	chronic kidney disease	Int64
2	nCkd	not chronic kidney disease	Int64

Pada penelitian ini peneliti merancang sistem prediksi gagal ginjal seperti pada Gambar 1. Pada flowchart tersebut sistem dimulai dari menginput dataset, lalu memisahkan dataset menjadi sebuah data latih dan data uji, kemudian melakukan prediksi data test dan yang terakhir dilakukan pengujian dari data uji dan dan data latih maka didapatkanlah hasil dari prediksinya.



Gambar.1 Flowchart sistem prediksi gagal ginjal

2. HASIL DAN PEMBAHASAN

a. K-Nearest Neighbour

Memuat Dataset

Hal pertama yang perlu dilakukan dalam pengujian ini yaitu memuat dataset yang berisi informasi tentang penyakit gagal ginjal kronis dari file CSV untuk memeriksa tipe data dan struktur dataset untuk memahami komposisinya.

```

# Load Dataset
df = pd.read_csv('/content/chronic kidney disease.csv')
df.head()

```

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn	Class
0	80.0	1.020	1.0	0.0	1.0	36.0	1.2	137.53	4.63	15.4	7800.0	5.20	1.0	1
1	50.0	1.020	4.0	0.0	1.0	18.0	0.8	137.53	4.63	11.3	6000.0	4.71	0.0	1
2	80.0	1.010	2.0	3.0	1.0	53.0	1.8	137.53	4.63	9.6	7500.0	4.71	0.0	1
3	70.0	1.005	4.0	0.0	1.0	56.0	3.8	111.00	2.50	11.2	6700.0	3.90	1.0	1
4	80.0	1.010	2.0	0.0	1.0	26.0	1.4	137.53	4.63	11.6	7300.0	4.60	0.0	1

Gambar.2 Memuat Dataset

Set Fitur

Pada tahap ini format data frame pandas perlu diubah menjadi sebuah array Numpy agar dapat menggunakan library scikit-learn.

```
[ ] # Fitur
X = df[['Bp', 'Sg', 'Al', 'Su', 'Rbc', 'Bu', 'Sc', 'Sod', 'Pot', 'Hemo', 'Wbcc',
'Rbcc', 'Htn']].values #.astype(float)

[ ] # Class
y = df['Class'].values
```

Gambar 2. Set Fitur

Normalisasi Data

Tahap normalisasi data berfungsi untuk membuat beberapa variabel data memiliki nilai yang sama, sehingga tidak ada data yang terlalu besar maupun terlalu kecil.

```
# Normalisasi fitur menggunakan StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Gambar.4 Normalisasi Data

Pembagian Data menjadi Data Latih dan Uji

Data yang digunakan akan dibagi menjadi 2 data, yaitu data latih dan data uji dapat dilihat pada gambar 5.

```
# Pemisahan dataset menjadi set pelatihan dan set pengujian
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)

Train set: (280, 13) (280,)
Test set: (120, 13) (120,)
```

Gambar 5 Pembagian data latih dan uji

Pada pengujian ini peneliti menggunakan 4 persentase perbandingan untuk menguji akurasi data yang digunakan. Pembagian data ditampilkan pada Tabel.2 berikut.

Tabel 2 Pembagian Data Latih dan Data Uji

No	Presentase Data Set	Data Test	Data Latih
1	10% : 90%	40	360
2	20% : 80%	80	320
3	30% : 70%	120	280
4	40% : 60%	160	240

Pelatihan Model

Tahap selanjutnya menentukan nilai k untuk melihat seberapa bagus tingkat akurasi suatu prediksi pada model.

```
# Mendefinisikan dan melatih model KNeighborsClassifier
k_value = 3 # Gantilah dengan nilai k yang sesuai
knn_model = KNeighborsClassifier(n_neighbors=k_value)
knn_model.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

Gambar.6 Inisialisasi Model

Data Prediksi

Data ini merupakan data uji yang digunakan untuk melihat prediksi klasifikasi berdasarkan dataset uji. Hasil data prediksi dapat dilihat pada gambar 7.

```
# Prediksi pada set pengujian
y_pred = knn_model.predict(X_test)
y_pred[0:5]

array([1, 0, 0, 1, 1])
```

Gambar.7 Data Prediksi

Evaluasi Kinerja Model

Tahap ini merupakan penilaian kinerja model terhadap data uji yang mencakup berbagai metrik seperti akurasi, presisi, recall dan confusion matrix. Berikut hasil pengujian dapat dilihat pada gambar 8

```
Accuracy: 0.9916666666666667
Precision: 0.9918367346938776
Recall: 0.9916666666666667

Confusion Matrix:
[[48  0]
 [ 1 71]]
True Positives: 71
True Negatives: 48
False Positives: 0
False Negatives: 1
```

Gambar.8 Hasil Evaluasi Kinerja Model

Berdasarkan hasil pengujian nilai akurasi yang diuji pada persentase data pelatihan (70%) dan data uji (30%) serta nilai k=3 yaitu sebesar 99.16%

Tingkat Akurasi

Berdasarkan hasil keseluruhan pengujian tingkat akurasi terbaik mencapai 99.16% berada pada persentase data pelatihan (70%) dan data uji (30%). Sedangkan untuk hasil akurasi persentase lainnya dapat dilihat pada tabel 3 berikut.

Tabel.3 Perbandingan Hasil Tingkat Akurasi Untuk Persentase Lainnya

No	Pembagian Data		Hasil Akurasi		Nilai K
	Uji	Latih	Uji	Latih	
1	10%	90%	97.5%	98.61%	k =1
2	20%	80%	97.5%	98.75%	k=1
3	30%	70%	99.16%	99.28%	k=3
4	40%	60%	98.12%	99.58%	k=2

Menghitung Nilai K Terbaik

Tahapan ini dilakukan agar dapat menemukan nilai k yang optimal pada sebuah model, sehingga mampu memperoleh hasil akurasi yang terbaik. Misalnya, pada perhitungan persentase data pelatihan (70%) dan data uji (30%), nilai k terbaik berada pada k=3 dengan tingkat akurasi sebesar 99.16%. Berikut hasil pengujian yang dapat dilihat pada gambar 9.

```
[21] #Berikut pengujian nilai k dari k=1 s/d k=40
Ks = 40
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
ConfusionMx = [];
for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    y_pred=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, y_pred)

    std_acc[n-1]=np.std(y_pred==y_test)/np.sqrt(y_pred.shape[0])

mean_acc[0:10]

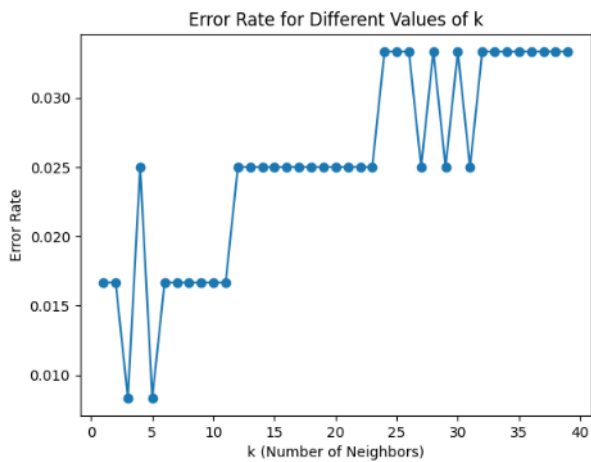
print("The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()

The best accuracy was with 0.991666666666667 with k= 3
```

Gambar.9 Hasil Nilai K Terbaik

Menghitung Error Rate

Tahapan ini bertujuan untuk mengukur seberapa baik model dalam menghasilkan prediksi yang benar. Pada pengujian ini tingkat error terkecil terdapat pada nilai k=3 dan k=5, yaitu sebesar 0.0083. Dapat dilihat pada gambar 10 bahwa semakin besar nilai k maka semakin tinggi tingkat kesalahan dalam model.



Gambar.10 Error Rate

Memuat Dataset

Memuat dataset yang bertujuan untuk memeriksa tipe data dan struktur dataset serta memahami komposisinya.

```
# Load Dataset
df = pd.read_csv('/content/chronic kidney disease.csv')
df.head()

Bp Sg Al Su Rbc Bu Sc Sod Pot Hemo Wbcc Rbcc Htn Class
0 80.0 1.020 1.0 0.0 1.0 36.0 1.2 137.53 4.63 15.4 7800.0 5.20 1.0 1
1 50.0 1.020 4.0 0.0 1.0 18.0 0.8 137.53 4.63 11.3 6000.0 4.71 0.0 1
2 80.0 1.010 2.0 3.0 1.0 53.0 1.8 137.53 4.63 9.6 7500.0 4.71 0.0 1
3 70.0 1.005 4.0 0.0 1.0 56.0 3.8 111.00 2.50 11.2 6700.0 3.90 1.0 1
4 80.0 1.010 2.0 0.0 1.0 26.0 1.4 137.53 4.63 11.6 7300.0 4.60 0.0 1
```

Gambar.11 Memuat Dataset

Persiapan Data

Tahap selanjutnya memisahkan variabel target ('Class') dari variabel fitur dan membuat dua data frame : cls_data untuk target dan atr_data untuk fitur. Fitur-fitur ini

merupakan parameter-parameter penting terkait kesehatan ginjal.

```
# Memisahkan kolom 'Class' sebagai target
cls_data = df['Class']

# Memisahkan kolom lainnya sebagai fitur
atr_data = df.drop(columns='Class')
atr_data.head()

Bp Sg Al Su Rbc Bu Sc Sod Pot Hemo Wbcc Rbcc Htn
0 80.0 1.020 1.0 0.0 1.0 36.0 1.2 137.53 4.63 15.4 7800.0 5.20 1.0
1 50.0 1.020 4.0 0.0 1.0 18.0 0.8 137.53 4.63 11.3 6000.0 4.71 0.0
2 80.0 1.010 2.0 3.0 1.0 53.0 1.8 137.53 4.63 9.6 7500.0 4.71 0.0
3 70.0 1.005 4.0 0.0 1.0 56.0 3.8 111.00 2.50 11.2 6700.0 3.90 1.0
4 80.0 1.010 2.0 0.0 1.0 26.0 1.4 137.53 4.63 11.6 7300.0 4.60 0.0
```

Gambar.12 Persiapan Data

Pembagian Data menjadi Data Latih dan Data Uji Pada tahap ini data dibagi menjadi dua bagian: data pelatihan (70%) dan data uji (30%). Proses ini membantu untuk melatih model pada satu set data dan menguji performanya pada set data terpisah.

```
# Pisahkan data menjadi set pelatihan dan set pengujian
xtrain, xtest, ytrain, ytest = train_test_split(atr_data, cls_data, test_size=0.3, random_state=1)

# Melihat jumlah data training dan testing
print('Jumlah data training:', len(xtrain))
print('Jumlah data testing:', len(xtest))

Jumlah data training: 280
Jumlah data testing: 120
```

Gambar.13 Pembagian data latih dan uji

Pada pengujian ID3 dilakukan pula pengujian menggunakan 4 persentase perbandingan untuk menguji akurasi data yang digunakan. Pembagian data ditampilkan pada Tabel.4 berikut.

Tabel.4 Pembagian Data Latih dan Data Uji

No	Presentasi Data Test	Data Test	Data Latih
1	10% : 90%	40	360
2	20% : 80%	80	320
3	30% : 70%	120	280
4	40% : 60%	160	240

Inisialisasi Model dan Pelatihan Model Decision Tree diinisialisasi dengan beberapa parameter tertentu, seperti kedalamanmaksimumpohon (max_depth), dll. Model Decision Tree dilatih menggunakan data pelatihan (fitur dan target). Ini melibatkan proses pembentukan pohon keputusan berdasarkan pola yang ditemukan dalam data pelatihan.

```
# Inisialisasi model Decision Tree dengan penyesuaian parameter
dt_model = DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_split=2, min_samples_leaf=1, random_state=1)

# Melatih model
dt_model.fit(xtrain, ytrain)

DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=1)
```

Gambar.14 Pelatihan Model

Evaluasi Kinerja Model

Tahap berikutnya Model kemudian diuji pada data uji, dan hasil prediksinya dievaluasi menggunakan metrik akurasi. Akurasi diukur untuk kedua set data pelatihan dan uji untuk memahami sejauh mana model dapat mempelajari dan menggeneralisasi pola dalam data. Hasil pengujian dapat dilihat pada gambar 15.

```
# Evaluasi kinerja model pada set pengujian
accuracy_test = accuracy_score(ytest, y_pred_test)
print("Test set Accuracy:", accuracy_test)

# Evaluasi kinerja model pada set pelatihan
accuracy_train = accuracy_score(ytrain, y_pred_train)
print("Training set Accuracy:", accuracy_train)

Test set Accuracy: 0.9833333333333333
Training set Accuracy: 0.9857142857142858
```

Gambar.15 Hasil Akurasi

Berdasarkan hasil pengujian pada persentase data pelatihan (70%) dan data uji (30) memiliki nilai akurasi sebesar 98.33%

Tingkat Akurasi

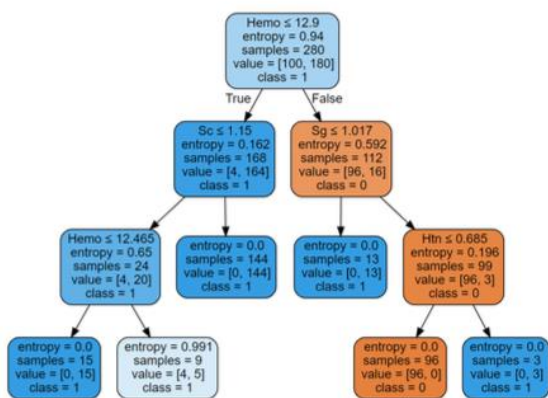
Berdasarkan hasil perhitungan keseluruhan tingkat akurasi terbaik mencapai 98.33% terdapat pada persentase data pelatihan (70%) dan data uji (30%). Sedangkan hasil akhir akurasi pada tiap persentase lainnya dapat dilihat pada tabel 3 berikut.

Tabel.3 Perbandingan Hasil Tingkat Akurasi Untuk Persentase Lainnya

No	Pembagian Data		Hasil Akurasi	
	Uji	Latih	Uji	Latih
1	10%	90%	95%	98.88%
2	20%	80%	97.5%	98.75%
3	30%	70%	98.33%	98.57%
4	40%	60%	96.25%	98.75%

Visualisasi Pohon ID3

Tahap terakhir yaitu menampilkan struktur pohon dan bagaimana model membuat keputusan. Model pohon ditampilkan pada gambar berikut.



2. KESIMPULAN DAN SARAN

Hasil pengujian menggunakan metode K-Nearest Neighbor dan Decision Tree ID3 didapat nilai akurasi yang sangat tinggi dari kedua metode. Nilai akurasi tertinggi dari metode KNN sebesar

99,16%. Sedangkan pada metode Decision Tree ID3, nilai akurasi tertinggi adalah 98,33%.

Berdasarkan hasil pengujian klasifikasi penyakit gagal ginjal kronis di atas, nilai akurasi tertinggi didapat dengan metode K-Nearest Neighbor dengan nilai 99,16%.

Berdasarkan hasil penelitian, beberapa hal perlu dikembangkan lebih lanjut lagi pada pengujian dataset yang berbeda untuk mengetahui sejauh mana model ini dapat diterapkan pada situasi yang lebih luas. Selain itu, perlu juga melakukan perbandingan kinerja metode KNN dan Decision Tree ID3 dengan metode klasifikasi lainnya untuk melihat apakah ada metode yang lebih baik dalam memprediksi penyakit ginjal kronis.

REFERENSI

Yunus, W. (2018). Algoritma K-Nearest Neighbor Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Ginjal Kronik. *Jurnal Cosphi*, 2(2). <https://cosphijournal.unisan.ac.id/index.php/cosphihome/article/view/43>

Arifin, T., & Ariesta, D. (2019). PREDIKSI PENYAKIT GINJAL KRONIS MENGGUNAKAN ALGORITMA NAIVE BAYES CLASSIFIER BERBASIS PARTICLE SWARM OPTIMIZATION. *Jurnal Tekno Insentif*; <https://www.semanticscholar.org/paper/PREDIKSI-PENYAKIT-GINJAL-KRONIS-MENGGUNAKAN-NAIVE-Arifin-Ariesta/54f2549b0542f9b6c0cc7eaf8eacea31ebf88e2>

Dudi irawan, Hardian Oktavianto, & moh khoirul anam. (2019). ANALISIS PENERAPAN ALGORITMA NAIVE BAYES UNTUK KLASIFIKASI PENYAKIT GINJAL KRONIS. 1(2), 127–134. <https://doi.org/10.32528/jasie.v1i2.3606>

Dimas, Imam Cholisoddin, & Santoso, E. (2018). Klasifikasi Risiko Gagal Ginjal Kronis Menggunakan Extreme Learning Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(11), 5220–5228. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/3237>

Rusda Wajhillah. (2019). PENERAPAN METODE ALGORITMA ID3 UNTUK PREDIKSI DIAGNOSA GAGAL GINJAL KRONIS (STUDI KASUS: RSUD SEKARWANGI SUKABUMI). *Jurnal Ilmiah KLIK*, 6(1), 97–97. <https://doi.org/10.20527/klik.v6i1.211>

Ikhsan Wisnuadji Gamadarena, & Indra Waspada. (2020). Implementasi Data Mining untuk Deteksi Penyakit Ginjal Kronis (PGK) menggunakan K-Nearest Neighbor (KNN) dengan Backward Elimination. <https://www.semanticscholar.org/paper/Implementasi-Data-Mining-untuk-Deteksi-Penyakit-Gamadarena-Waspada/b8f640448b3d42b2300bd20121b10a2a26d17b41>

Ariani, A., & Samsuryadi, S. (2020). Klasifikasi Penyakit Ginjal Kronis menggunakan K-Nearest Neighbor. core.ac.uk/oai:seminar.ilkom.unsri.ac.id:article/2129

Alvi Norma Utami. (2020, June 26). Studi Komparasi Klasifikasi Gagal Ginjal Kronis Menggunakan Algoritma SVM, KNN dan MLP. *ResearchGate*; Universitas Negeri Surabaya. <https://www.researchgate.net/publication/>

361835732_Studi_Komparasi_Klasifikasi_Gagal_Ginjal_Kronis_Menggunakan_Algoritma_SVM_KNN_dan_MLP

I. G. A. Mahardika Pratama, Astuti, L. G., Widiartha, I. M., Putra, C., Rai, C., & I. M. B. A. Atmaja Darmawan. (2022). Diagnosis Penyakit Ginjal Kronis dengan Algoritma C4.5, K-Means dan BPSO. *Jurnal Elektronik Ilmu Komputer Udayana*; <https://www.semanticscholar.org/paper/Diagnosis-Penyakit-Ginjal-Kronis-dengan-Algoritma-Pratama-Astuti/84c5f086bb66abbf7e24df1fb92a349bc7480071>

Akbar, A., Novanto Yudistira, & Achmad Ridok. (2023). Identifikasi Gagal Ginjal Kronis dengan Mengimplementasikan Metode Support Vector Machine beserta K-Nearest Neighbour (SVM-KNN). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(2), 301–308. <https://doi.org/10.25126/jtiik.20231026059>

Nandi Prabu Nugraha, Azim, R., Syauqi Zalffa Daffa, & Putri Salma Ningayu. (2023). Perbandingan Akurasi Metode Naïve Bayes dan Metode KNN untuk Memprediksi Gagal Ginjal Kronis. *Jurnal Rekayasa Elektro Sriwijaya*, 5(1), 1–10. <https://doi.org/10.36706/jres.v5i1.63>

Yogi, M. (2021). Penerapan Algoritma Iterative Dichotomiser 3 (ID3) Untuk Klasifikasi Penyakit Tifoid. <http://repository.unmuhjember.ac.id/8330/9/9.%20Jurnal.pdf>