Jurnal Elektronik Ilmu Komputer Udayana
Volume 12, No 2. November 2023

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

Implementasi Decision Tree berbasis Forward Selection untuk Klasifikasi Penyakit Ginjal Kronis

Jeremi Herodian Abednigo[a1], Made Agung Raharja[a2]

[a]Program Studi Informatika, Universitas Udayana
Kuta Selatan, Badung, Bali, Indonesia
[1]jeremi.herodian.a43@gmail.com
[2]made.agung@unud.ac.id

Abstract

Penyakit Ginjal Kronis adalah penyakit yang umum di masyarakat dengan jumlah penderita yang terus meningkat. Penyakit ini menyerang ginjal yang mengakibatkan ginjal tidak bisa berfungsi dengan dengan baik. Data mining adalah proses untuk mengekstrak data dengan tujuan mendapatkan informasi yang berharga, salah satunya metode data mining adalah klasifikasi. Algoritma Decision Tree adalah salah satu algoritma klaifikasi yang bisa digunakan untuk melakukan klasifikasi penyakit ginjal kronis. Pada penelitian ini, klasifikasi diakukan dengan Decision Tree yang digabungkan dengan seleksi fitur menggunakan Forward Selection. Forward Selection digunakan untuk mengurangi fitur yang tidak relevan terhadap target klasifikasi. Dataset yang digunakan adalah Chronic Kidney Disease dataset dari Kaggle. Pada hasil pengujian Decision Tree dengan bantuan library sklearn dari python dengan cross validation sebanyak 10 fold didapatkan bahwa seleksi fitur dengan forward selection berhasil meningkatkan hasil akurasi, presisi, recall, dan f1 score secara berurutan adalah 99.5%, 98,75%, 100%, 99,35%

Keywords: *Klasifikasi, Penyakit Ginjal Kronis, Decision Tree, Feature Selection, Forward Selection*

1. Pendahuluan

Organ ginjal merupakan salah satu organ yang penting dalam tubuh manusia yang berfungsi untuk menjaga kadar darah dengan mencegah menumpuknya limbah dan mengatur keseimbangan carian tubuh. Penyakit ginjal adalah gangguan yang mengenai organ ginjal yang banyak disebabkan oleh infeksi, tumor, kelainan bawaan, gangguan metabolic, atau degeneratif dan lain-lain [1].

Penyakit ginjal kronis merupakan salah satu penyakit yang tingkat penderitanya cukup tinggi di dunia. Menurut The United States Renal Data System (USRDS), jumlah penderita penyakit ginjal kronis diperkirakan mencapai 2.020 kasus perjuta penduduknya pada tahun 2012 dengan tingkat pertumbuhan mencapai 7% dan untuk di Amerika Serikat, hampir setiap tahunnya sekitar 70 orang meninggal dunia akibat menderita penyakit ginjal kronis[4]. Dengan laju pertumbuhan penduduk yang semakin pesat, Hal ini juga meningkatkan jumlah penderita penyakit CKD. Data yang didapatkan dari Global Burden of Disease, dilaporkan bahwa penyakit CKD menempati rangking ke-27 pada tahun 1990 dan rangking ke-18 pada tahun 2010 [3]. Menurut data Kementerian Kesehatan RI, 2 dari setiap 1.000 orang di Indonesia atau 499.800 orang menderita penyakit ginjal kronis pada tahun 2013. Prevalensi penyakit ginjal kronis meningkat seiring bertambahnya usia [2].

Data mining dapat diaplikasikan di bidang kesehatan misalnya mendiagnosis penyakit kanker payudara, penyakit jantung, penyakit diabetes dan lain-lain [5]. 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 [6].

Decision Tree Learning (DTL) merupakan salah satu teknik pembelajaran mesin (Machine Learning) yang menggunakan aturan klasifikasi berstruktur sekuensial hirarki dengan cara mempartisi himpunan data latih secara rekursif [7]. Pada peneliti terdahulu menggunakan Decision Tree sebagai algoritma

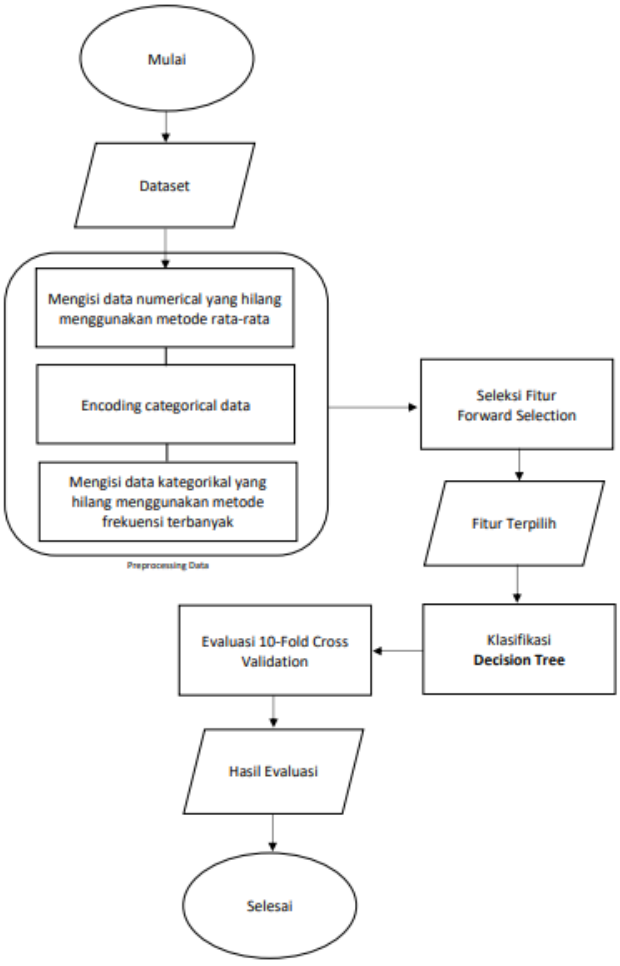
untuk prediksi metode penyakit kutil, didapatkan hasil akurasi sebesar 90% [8]. Juga pada penelitian sebelumnya mengenai perbandingan metode Decision Tree dengan Naïve Bayes untuk klasifikasi tumor otak, didapatkan akurasi algoritma Decision Tree lebih unggul dibandingkan Naïve Bayes yaitu 96% untuk Decision Tree dan 91% untuk Naïve Bayes [9].

Banyak hal yang bisa dilakukan untuk meningkatkan akurasi pada algoritma Decision Tree,salah satunya adalah dengan cara melakukan seleksi fitur. Seleksi fitur adalah cara menyeleksi subset fitur yang berlebihan dan kurang informatif pada dataset [2]. Pada penelitian terdahulu dengan menggunakan algoritma Decision Tree C4.5 dengan metode seleksi fitur Binary Particle Swarm Optimization (BPSO) berhasil meningkatkan hasil akurasi dengan hasil akhir akurasi 96,869% [2]. Penelitian yang lain menggunakan metode wrapper backward selection untuk klasifikasi penyakit diabetes, didapatkan hasil akurasi yang baik sebesar 96,7% [10].

Berdasarkan hasil uraian penelitian di atas, penelitian ini akan membahas sejauh apa hasil dari metode klasifikasi Decision Tree dan Forward Selection sebagai seleksi fitur untuk mendapatkan subset yang relevan sehingga mendapatkan kinerja yang baik. Ukuran pengujian menggunakan k-fold cross validation sebanyak 10-fold.

2. Metode Penelitian

Pada studi kasus ini, alur pemrosesan meliputi pengumpulan data, preprocessing data, seleksi fitur menggunakan Forward Selection, klasifikasi Decision Tree, dan evaluasi menggunakan k-fold validation.



Gambar 1. Alur penelitian

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

Pada langkah awal data yang didapatkan akan melalui tahap preprocessing. Tahap preprocessing yang dilakukan meliputi pengisian *missing values* dan encoding data categorical. Metode missing value dibedakan berdasarkan jenis fitur, fitur numerical menggunakan mean dan fitur kategorikal menggunakan modus/frekuensi kemunculan data terbanyak. Data yang sudah diolah, siap untuk masuk seleksi fitur forward selection. Subset data yang terbaik kemudian masuk ke dalam model klasifikasi Decision Tree sekali lagi untuk dibandingan dengan hasil sebelum dilakukan seleksi fitur. Terakhir divalidasi menggunakan k-fold validation sebanyak 10-fold untuk divalidasi apakah model yang dihasilkan tidak terjadi overfitting. Hasil akhirnya adalah data evaluasi model Decision Tree berupa akurasi, presisi, recall, dan f1 score.

2.1 Preprocessing Data

Seperti yang dijelaskan sebelumnya, pada tahapan preprocessing data ini dilakukan penanganan pada missing value. Proses pengisian missing value berdasarkan pada jenis data. Untuk data numerikal menggunakan metode mean dan untuk kategorikal menggunakan metode frekuensi kemunculan nilai terbanyak.Berikut implementasi untuk pengisian missing value.

a. Mengisi Missing Value Data Kategorikal
Pengisian data yang hilang dengan tipe data kategorikal bisa menggunakan rumus modus sebagai berikut

$$Mo = Tb + (\frac{d1}{d1 + d2})c$$

Untuk implementasi di sini mengguanakn bantuan dari library sklean yaitu SimpleImputer dengan memberikan parameter “most_frequent” sehingga data kategorikal yang hilang akan terisi sesuai dengan kemunculan data terbanyak.

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='most_frequent')
imputer = imputer.fit(data[cat_cols])
data[cat_cols] = imputer.transform(data[cat_cols])
```

Gambar 2. Penerapan missing value kategorikal dengan modus

b. Encoding
Pada tahapan encoding, data yang bersifat kategorikal string diubah menjadi data numerik yang mewakili data kategorikal tersebut.

	red_blood_cells	pus_cell	pu
0.0	normal	normal	
0.0	normal	normal	
5.0	normal	normal	
0.0	normal	abnormal	

Gambar 3. Sebelum encoding data

```
if (data.loc[idx, 'pus_cell'] == "normal"):
    data.loc[idx, 'pus_cell'] = 1
elif (data.loc[idx, 'pus_cell'] == "abnormal"):
    data.loc[idx, 'pus_cell'] = 0

if (data.loc[idx, 'pus_cell_clumps'] == "present"):
    data.loc[idx, 'pus_cell_clumps'] = 1
elif (data.loc[idx, 'pus_cell_clumps'] == "notpresent"):
    data.loc[idx, 'pus_cell_clumps'] = 0
```

Gambar 4. Penerapan missing value data numerikal

red_blood_cells	pus_cell
1	1
1	1
1	1
1	0

Gambar 5. Setelah encoding data

Seperti terlihat pada gambar di atas, data telah berubah dari kategorikal menjadi numerikal. Data encoding sangat penting karna model Decision Tree hanya mengenali data yang numerik.

c. Mengisi Missing Value Data Numerikal
Data yang hilang menggunakan rumus mean.

$$Xrata - rata = \frac{\sum_{i=1}^{N} Xi}{N}$$

Berikut ini adalah implementasi missing value menggunakan bantuan dari library sklearn.

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
imputer = imputer.fit(data)
all_col = data.columns
data = pd.DataFrame(data=imputer.transform(data), columns=all_col)
df = data
```

Gambar 6. Penerapan missing value data numerikal

Dengan SimpleImputer, penerapan mean untuk missing value bisa digunakan dengan memberikan parameter ‘mean’ sehingga data-data yang hilang pada semua kolom yang bertipe numerikal akan terisi.

2.2 Decision Tree

Algortima Decision Tree merupakan suatu metode pengklasifikasian yang menggunakan contoh pohon, menyatakan node yang menggambarkan tiap atribut, yang mana daun menggambarkan tiap kelas, juga setiap cabangnya menggambarkan nilai dari tiap kelas. Node akar menyatakan node yang berada paling atas dari pohon. Setiap node ini menggambarkan node pembagi, yang mana tiap node ini merupakan satu masukan dan memiliki sedikitnya dua keluaran [11]. Leaf node adalah node terakhir, hanya mempunyai satu masukan, dan tidak mempunyai keluaran. Pohon keputusan pada tiap leaf node menyatakan label tiap kelas. Pohon keputusan pada tiap cabangnya menyatakan keadaan yang harus diisi dan tiap puncak pohonnya menggambarkan nilai kelas data [12].

Pada kebanyakan kasus, decision tree bisa melakukan klasifikasi yang targetnya bersiifat biner atau disebut klasifikasi biner, contoh untuk memprefiksi jawaban ya atau tidak.

Berikut adalah beberapa jenis klasifikasi biner:

a. Classification Trees (*Yes/No types*)
   Jenis klasifikasi ini adalah contoh klasifikasi untuk memberikan prediksi jawaban ya atau tidak, hujan atau tidak, terinfeksi atau tidak, dan lain sebagainya.
b. Regression Tree (*Continuous data types*)
   Jenis klasifikasi ini adalah klasifikasi untuk tipe data kontiyu seperti angka bilangan riil.

Dari sekian banyak pohon keputusan yang dibuat, salah satu algoritma pohon keputusan yang populer adalah Algoritma ID3. ID3 adalah singkatan dari Iterative Dichotomiser 3. Ada beberapa definisi yang membangun algoritma ID3.

a. Entropy
   Entropy atau juga disebut Shannon Entropy dilambangkan dengan H(S) untuk himpunan S yang terbatas adalah ukuran jumlah ketidakpastian atau suatu nilai acak dalam data.

$$H(S) = \sum_{x \in X} p(x) log_{2 \frac{1}{p(x)}}$$

b. Information Gain
   Nama lain dari Information Gain adalah Kullback-Leibler divergence dilambangkan IG(S,A) untuk semua himpunan S adalah peubah efektif dalam entropi setelah memutuskan atribut A tertentu. Pengukuran perubahan ini relative dalam entropi sehubung dengan variable independent.

$$IG(S, A) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

2.3 Feature Selection

Feature Selection atau yang sering disebut juga sebagai attribute selection merupakan proses menemukan subset hasil seleksi fitur dari suatu dataset. Feature selection dipakai pada bidang statistika, dan data mining[13].

Forward selection adalah salah satu metode dalam Feature Selection. Feature selection adalah metode yang penting untuk menghasilkan klasifikasi yang baik. Tanpa feature selection proses komputasi dan performansi model menjadi buruk. Tujuan dari feature selection adalah membuang atribut yang tidak relevan[13].

a. Forward Selection
   Metode Forward Selection mengadopsi prinsip regresi Linear. Forward Selection adalah salah satu model wrapper yang digunakan mereduksi atribut dataset [13].
   Proses pencarian attribute dengan forward selection diawali dengan empty model, selanjutnya tiap variabel dimasukan hingga kriteria kombinasi model attribute terpenuhi dengan baik. Berikut adalah pseudo code dari forward selection:
   1. Membuat empty set: $Y_k = \{\emptyset\},\ k = 0$
   2. Memilih feature terbaik: $X^+ = \arg max_{x+ \in Yk}[J(Y_k + X^+)]$
   3. Jika $[J(Y_k + X^+)] > J(Y_k)$:
      - Update $Y_k + 1 = Y_k + X^+$
      - $k = k + 1$

- Kembali ke step-2

2.4 K-Fold Cross Validation

Cross validation atau disebut validasi silang adalah salah satu metrik untuk mengukur hasil dari algoritma klasifikasi. Sedangkan K-fold validation adalah salah satu cara untuk mengetahui rata-rata keberhasilan sebuah sistem klasifikasi. K-fold validation akan mengacak sebuah dataset secara silang sehingga sistem diuji untuk beberapa dataset yang sudah diacak. Tujuanya adalah untuk menghindari dominasi data pada pembelajaran model klasifikasi. Validasi k-fold akan dimulai dengan membagi beberapa data menjadi n-fold yang diinginkan. Demikian jika data dibagi menjadi 5 akan menghasilkan 5 partisi data dengan ukuran yang sama D1, D2, D3. Kemudian dilakukan proses test dan training sebanyak jumlah fold. Setiap iterasi ke-i, data partisi n akan menjadi dataset uji dan sisanya menjadi dataset pelatihan[15].

Setiap iterasi dihitung Accuracy, Presisi, Recall, dan F1 Scorenya menggunakan rumus berikut:

$$Accuracy = \frac{\sum ccorrect\ classification\ test\ data}{\sum total\ test\ data}$$

$$Precision = \frac{\sum ccorrect\ classification\ test\ data}{\sum total\ data\ predicted}$$

$$Recall = \frac{\sum ccorrect\ classification\ test\ data}{\sum total\ test\ data\ on\ a\ specifi\ c\ class}$$

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

3. Hasil dan Pembahasan

Pada hasil pengujian, data yang digunakan berjumlah 400 entitas dengan 24 fitur dan 1 label dengan 2 class yaitu CKD (Chronic Kidney Disease) dan Not CKD (Not Chronic Kidney Disease). Pada proses seleksi fitur forward selection yang dibantu dengan library mlxtend dari python, setiap fitur yang ditambahkan kepada subset dilakukan perhitungan cross validation sebanyak 10-Fold. Seleksi melakukan iterasi sebanyak 24 kali sesuai dengan jumlah fiturnya. Pada setiap iterasi, kombinasi fitur yang menghasilkan akurasi terbaik dari cross validation akan dipilih hingga pada akhirnya semua fitur berhasil dikombinasikan. Tahap akhir dari semua iterasi akan menghasilkan nilai rata-rata cross validation yang kemudian akan di seleksi dengan mengambil nilai rata-rata cross validation tertinggi. Hasil fitur-fitur yang terpilih menggunakan forward selection seperti pada Tabel 1.

Table 1. Fitur terpilih

Algoritma	Index Features
Decision Tree	**0, 2, 3, 5, 11, 13, 14, 15, 18**

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

Table 2. Keterangan Fitur

Index	Deskripsi
0	**age**
2	**Specific gravity**
3	**Albumin**
5	**Red blood cells**
11	**Serum creatinine**
13	**Potassium**
14	**Haemogllobin**
15	**Packed cell volume**
18	**Hypertension**

Untuk pengujian Decision Tree tanpa menggunakan seleksi fitur, didapatkan nilai akurasi, Presisi, Recall, F1 Score tanpa seleksi fitur, hasil cross validation seperti tertera pada Tabel 3.

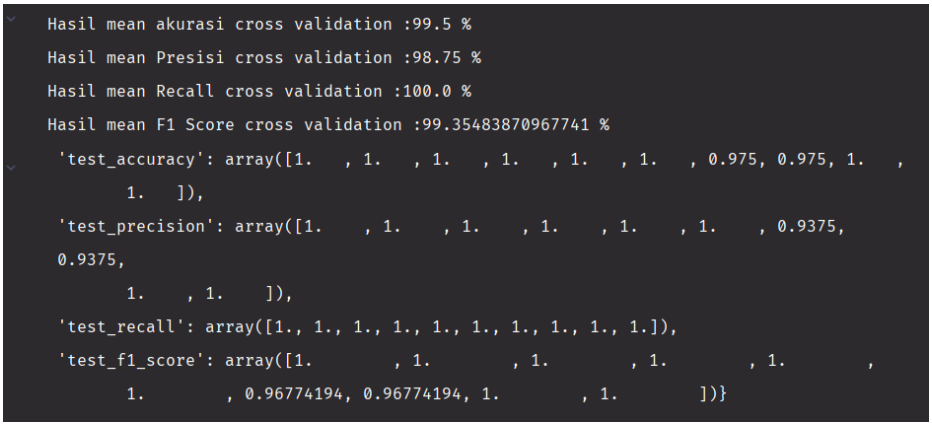
```
Hasil mean akurasi cross validation :97.25 %
Hasil mean Presisi cross validation :96.16666666666666 %
Hasil mean Recall cross validation :96.66666666666666 %
Hasil mean F1 Score cross validation :96.31723078552889 %

        0.00298953, 0.00299048, 0.00199318, 0.00199366, 0.00299001]),
 'test_accuracy': array([0.975, 0.975, 0.975, 1.   , 0.95 , 0.95 , 1.   , 0.95 , 0.975,
        0.975]),
 'test_precision': array([0.9375    , 1.        , 0.9375    , 1.        , 1.        ,
        0.93333333, 1.        , 0.93333333, 0.9375    , 0.9375    ]),
 'test_recall': array([1.        , 0.93333333, 1.        , 1.        , 0.86666667,
        0.93333333, 1.        , 0.93333333, 1.        , 1.        ]),
 'test_f1_score': array([0.96774194, 0.96551724, 0.96774194, 1.        , 0.92857143,
        0.93333333, 1.        , 0.93333333, 0.96774194, 0.96774194])}
```

Gambar 7. Output 10-Fold Validation **sebelum** seleksi fitur

Table 3. Akurasi Decision Tree **sebelum** seleksi fitur

Fold	**Ukuran evaluasi (Rata-Rata Fold)**			
	Akurasi	Presisi	Recall	F1 Score
1	97,5	93,75	100	96,77
2	97,5	100	93,33	96,55
3	97,5	93,75	100	96,77
4	100	100	100	100
5	95	100	86,67	92,85
6	95	93,33	93,33	93,33
7	100	100	100	100
8	95	93,33	09,33	93,33
9	97,5	93,75	100	96,77
10	97,5	93,75	100	96,77
Rata-Rata	**97,25%**	**96,16%**	**96.66%**	**96.31%**

Pada hasil keseluruhan fitur sebelum diseleksi mendapatkan hasil yang lumayan baik, akurasi rata-rata yang diperoleh dari setiap fold adalah 97,25 %. Kemudian pada Tabel 4 di bawah ini adalah hasil dari Decision Tree dengan seleksi fitur forward selection.



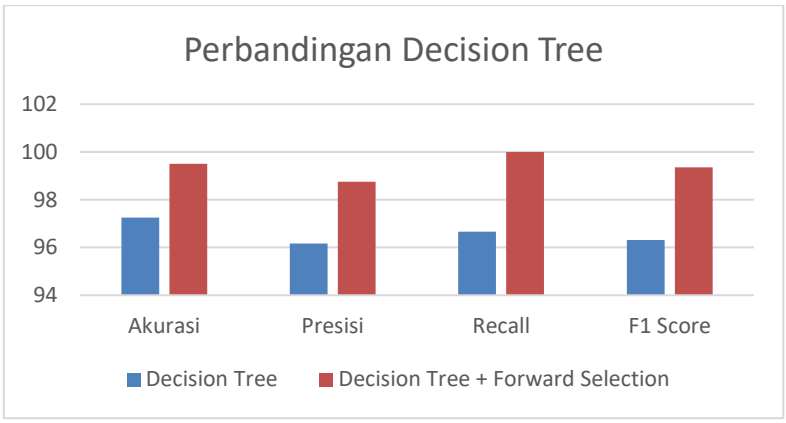
Gambar 8. Output 10-Fold Validation **setelah** seleksi fitur

Table 4. Akurasi Decision Tree **sesudah** seleksi fitur

Fold	**Ukuran evaluasi (Rata-Rata Fold)**			
	Akurasi	Presisi	Recall	F1 Score
1	100	100	100	100
2	100	100	100	100
3	100	100	100	100
4	100	100	100	100
5	100	100	100	100
6	100	100	100	100
7	97,5	93,75	100	96,77
8	97,5	93,75	100	96,77
9	100	100	100	100
10	100	100	100	100
Rata-Rata	**99.5%**	**98,75%**	**100%**	**99,35%**

Untuk melihat hasil perbandingan antara Decision Tree sebelum dan sesudah dilakukan seleksi fitur, bisa dilihat pada histogram pada Gambar 7.

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



Gambar 7. Perbandingan Decision Tree

Pada gambar histogram di atas bisa terllihat bahwa metode seleksi fitur bisa meningkatkan akurasi Decision Tree lebih baik daripada hanya menggunakan Decision Tree. Jika dilihat dari perbandingan akurasi, Decision Tree tanpa seleksi fitur mendapatkan nilai sebesar 97,25%, sedangkan setelah dilakukan seleksi fitur hasilnya meningkat sebsar 2,25% menjadi 99,5% di mana nilai akurasi ini termasuk ke dalam *excellent accuracy*. Terbukti metode seleksi fitur forward selection berhasil meningkatkan akurasi Decision Tree pada kasus klasifikasi penyakit ginjal kronis.

4. Kesimpulan

Hasil kesimpulan pada penelitian ini bahwa metode seleksi fitur forward selection telah berhasil meningkatkan akurasi dari algoritma Decision Tree dengan hasil akurasi, presisi, recall, dan f1 score secara berurutan adalah 99.5%, 98,75%, 100%, 99,35% dengan kombinasi subset fitur yang terbaik sebanyak 9 fitur dari 24. Dari penelitian ini, peneliti juga menyimpulkan bahwa peran dari preprocessing data berpengaruh dalam mendapatkan hasil akurasi karna terdapat missing value. Jadi peneliti menyimpulkan bahwa pada studi kasus ini metode preprocessing data untuk missing value sangat penting untuk dilakukan.

Saran untuk penelitian selanjutnya yaitu bisa menerapkan metode missing value lainnya dan membandingkan hasilnya. Juga bisa menerapkan metode seleksi fitur lainnya seperti ekstraksi fitur (PCA) atau Seleksi fitur berdasarkan korelasi.

References

[1] I.W. Gamadarenda, and I.Waspada, "IMPLEMENTASI DATA MINING UNTUK DETEKSI PENYAKIT GINJAL KRONIS (PGK) MENGGUNAKAN K-NEAREST NEIGHBOR (KNN) DENGAN BACKWARD ELIMINATION" *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* Vol. 7, No. 2. 417-426, 2020.
[2] I.G.A.Mahardika Pratama, dkk, “Diagnosis Penyakit Ginjal Kronis dengan Algoritma C4.5, K-Means dan BPSO” Jurnal-Elektronik-Ilmu-Komputer-Udayana, Volume 10, No 4, 2022.
[3] 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.
[4] E.A.Kurnianto, dkk, “Klasifikasi Penderita Penyakit Ginjal Kronis Menggunakan Algoritme Support Vector Machine (SVM)” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 12, 2018.
[5] D. T. Larose, Discovering Knowledge in Data: An Introduction to Data Mining. United States of America: John Wiley & Sons, Inc, 2005.
[6] I. Handayani, “Penyakit Disk Hernia Dan Spondylolisthesis Dalam Kolumna Vertebralis,” vol. 1, no. 2, pp. 83–88, 2019, doi: 10.12928/JASIEK.v13i2.xxxx.
[7] Suyanto, Machine Learning Tingkat Dasar dan Lanjut, 1st ed. Bandung: Informatika Bandung, 2018.
[8] Fitriyani, and T.Arifin, “IMPLEMENTASI GREEDY FORWARD SELECTION UNTUK PREDIKSI METODE PENYAKIT KUTIL MENGGUNAKAN DECISION TREE”, Jurnal Sains dan Teknologi, Vol.9 No. 1, 2020.

[9] S.D.Kamil, “PERBANDINGAN METODE DECISION TREE DENGAN NAIVE BAYES DALAM KLASIFIKASI TUMOR OTAK CITRA MRI”, UPV Veteran Jakarta, 2022.

[10] iratama, M.Abid, “OPTIMASI ALGORITMA DATA MINING MENGGUNAKAN BACKWARD ELIMINATION UNTUK KLASIFIKASI PENYAKIT DIABETES”, Universitas Amikom, 2022.

[11] H.Ferdian, and H.Seng. (2017). Penerapan Algoritma C4.5 untuk Memprediksi Penerimaan Calon Pegawai Baru di PT WISE. Palembang: Jatisi, Vol. 3, No.2.H

[12] Robianto, dkk, “PENERAPAN METODE DECISION TREE UNTUK MENGKLASIFIKASIKAN MUTU BUAH JERUK BERDASARKAN FITUR WARNA DAN UKURAN”, Coding:Jurnal Komputer dan Aplikasi, Volume 09, No. 01, 2021

[13] H.B.Sasongko , and O.Arifin, “IMPLEMENTASI METODE FORWARD SELECTION PADA ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN NAIVE BAYES CLASSIFIER KERNEL DENSITY (STUDI KASUS KLASIFIKASI JALUR MINAT SMA)”, Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), Vol. 6, No. 4, 2019

[14] Pedregosa *et al, “Scikit-learn: Machine Learning in Python”,* JMLR 12, pp. 2825-2830, 2011.

[15] I K.S Putri Rahayua1, and I K.A. Mogi, “Implementation of K-Nearest Neighbor Algorithm in Heart Disease Classification”, Jurnal Elektronik Ilmu Komputer Udayana, Volume 10 No. 1, 2021.