IMPLEMENTASI ALGORITMA C4.5 DAN K-MEANS PADA DIAGNOSIS PENYAKIT GINJAL KRONIS

I Gede Aditya Mahardika Pratama[1)], **I Made Widiartha**[2)]
Program Studi Ilmu Komputer[1) 2)]
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, Badung, Bali[1) 2)]
adityamahardikapratama17@gmail.com[1)], madewidiartha@unud.ac.id[2]

ABSTRACT

The kidneys function to maintain the stability of the body by regulating the balance of electrolytes, body fluids and expenditure of metabolic products. Chronic kidney disease is a form of kidney disorder. This disease is a deadly disease, but it can be avoided with proper precautions. The prevalence of chronic kidney disease is increasing as a person ages. Classification technique is one technique that can be used in diagnosing chronic kidney disease. One of the machine learning algorithms that can be used for classification is the C4.5 algorithm. The C4.5 algorithm is one of the algorithms that can be used in the decision tree method (decision tree). The C4.5 algorithm can only use categorical data, so data of numerical type needs to be discretized. K-Means Clustering is one method that can be used in data discretization. Elbow method is used in determining the optimal number of k on K-Means by comparing the SSE value of each number of k. System testing was carried out using the confusion matrix and the values of accuracy, recall and precision were 97.92%, 94.44% and 100%.

Keywords : *Chronic Kidney Disease, Classification, C4.5, K-means, Discretization of Data, Elbow Method.*

ABSTRAK

Ginjal berfungsi untuk mempertahankan stabilitas tubuh dengan mengatur keseimbangan elektrolit, cairan tubuh dan pengeluaran hasil metabolisme. Penyakit ginjal kronis adalah salah satu bentuk gangguan pada ginjal. Penyakit ini merupakan penyakit yang mematikan, namun hal ini dapat dihindari dengan tindakan pencegahan yang tepat. Prevalensi penyakit ginjal kronis menjadi kian meningkat, seiring dengan bertambahnya umur seseorang. Teknik klasifikasi merupakan salah satu teknik yang dapat digunakan dalam mendiagnosis penyakit ginjal kronis. Salah satu algoritma *machine learning* yang dapat digunakan untuk klasifikasi adalah algoritma C4.5. Algoritma C4.5 adalah salah satu algoritma yang dapat digunakan dalam metode *decision tree* (pohon keputusan). Algoritma C4.5 hanya dapat menggunakan data kategorikal, sehingga data yang bertipe numerikal perlu dilakukan diskritisasi data. *K-Means Clustering* merupakan salah satu metode yang dapat digunakan dalam diskritisasi data. Metode *elbow* digunakan dalam penentuan jumlah k optimal pada *K-Means* dengan membandingkan nilai SSE masing-masing jumlah k. Pengujian sistem dilakukan dengan menggunakan *confusion matrix* dan didapatkan nilai *accuracy, recall* dan *precision* yaitu 97.92%, 94.44% dan 100%.

Kata Kunci : Penyakit Ginjal Kronis, Klasifikasi, C4.5, *K-Means*, Diskritisasi Data, Metode *Elbow*.

PENDAHULUAN

Ginjal merupakan organ tubuh yang berfungsi mempertahankan stabilitas tubuh dengan mengatur keseimbangan elektrolit, cairan tubuh dan pengeluaran hasil metabolisme (Amalia, 2018). Ginjal memilki fungsi yang sangat penting bagi kesehatan tubuh, sehingga akan sangat riskan apabila mengalami gangguan pada ginjal. Gangguan tersebut dapat dideteksi dengan melihat adanya kelainan pada darah, urin atau melalui prosedur biopsi ginjal (Kurnianto, et al., 2018). Penyakit ginjal kronis atau *Chronic Kidney Disease* (CKD) merupakan salah satu bentuk gangguan pada ginjal. Penyakit ini dapat mengakibatkan ketidakmampuan ginjal untuk melakukan fungsinya dengan baik yang disebabkan oleh penurunan kinerja organ ginjal (Kurnianto, et al., 2018). Menurut Kementerian Kesehatan RI pada tahun 2013, terdapat sebanyak 499.800 penduduk atau 2 per 1000 penduduk di Indonesia mengalami penyakit ginjal kronis.

Prevalensi penyakit ginjal kronis menjadi kian meningkat, seiring dengan bertambahnya umur seseorang (Yulianti, et al., 2020).

Teknik klasifikasi merupakan salah satu teknik yang dapat digunakan dalam mendiagnosis penyakit ginjal kronis. Klasifikasi merupakan salah satu teknik dari *data mining*. Dimana *data mining* merupakan suatu cara yang bertujuan dalam penemuan pola dari data yang dimanfaatkan untuk menyelesaikan suatu masalah melalui berbagai aturan proses (Handayani, 2019). Definisi dari klasifikasi adalah pengelompokan yang didasarkan pada data-data yang ada (Fadilla, et al., 2018).

Salah satu algoritma machine learning yang dapat digunakan untuk klasifikasi adalah algoritma C4.5. Algoritma C4.5 adalah salah satu algoritma yang dapat digunakan dalam metode *decision tree* (pohon keputusan). Penelitian terdahulu tentang klasifikasi telah dilakukan dengan membandingkan algoritma C4.5 dengan Naïve Bayes untuk memprediksi penyakit diabetes. Penelitian ini menunjukan algoritma C4.5 lebih unggul dengan akurasi sebesar 82,74% (Pujianto, et al., 2019). Kemudian penelitian lainnya melakukan komparasi algoritma C4.5, Naïve Bayes, k-NN, Log-R, dan *Deep learning* dalam prediksi penyakit liver. Dari perbandingan kelima algoritma, *Decision Tree* (C4.5) merupakan algoritma dengan hasil paling baik, dengan tingkat akurasi 2,56% dan AUC 0,594 (Fahdia, 2020).

Dalam pengklasifikasian dengan C4.5, diperlukan diskritisasi dari suatu kumpulan data. Teknik diskritisasi merupakan teknik yang mengkonversi bilangan numerikal menjadi bilangan kategorikal berdasarkan label interval atau label konseptual. Hal ini dilakukan dikarenakan C4.5 hanya mampu menggunakan data kategorikal dalam proses pelatihan dan pengujian data (Rochman, et al., 2019). Salah satu metode diskritisasi yang dapat digunakan adalah *K-Means* (Pradana, et al., 2018). Berdasarkan uraian permasalahan tersebut, maka penelitian ini berfokus untuk mengimplementasikan algoritma C4.5 dan diskritisasi dengan algoritma *K-Means Clustering* pada diagnosis penyakit ginjal kronis.

TINJAUAN PUSTAKA

K-Means

K-Means adalah salah satu algoritma pengklasteran yang cukup sederhana untuk mempartisi data ke dalam beberapa klaster. Algoritma ini cukup mudah untuk diimplementasikan dan dijalakan, relatif cepat, mudah disesuaikan dan banyak digunakan. Namun kelemahan dari algoritma ini yaitu pada inisialisasi *cluster* yang sangat sensitif. Berikut urutan proses algoritma *K-Means* (Rahman, et al., 2017) :

a. Tentukan jumlah k (*cluster*) yang akan dibuat.
b. Bangkitkan titik pusat (*centroid*) setiap *cluster* secara acak.
c. Hitung jarak setiap *centroid* terhadap masing-masing data dengan menggunakan perhitungan jarak *Euclidean Distance* dengan 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 (jarak data)

x_1 : *centroid*

x_2 : posisi objek data

d. Kelompokkan data dengan mencari jarak terdekat antara *centroid* dengan data.
e. Tentukan nilai *centroid* yang baru dengan mencari rata-rata dari setiap *cluster* yang bersesuaian menggunakan rumus berikut :

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

Keterangan:

n_k : jumlah data dalam *cluster* k

d_i : jumlah dari nilai jarak yang masuk dalam masing-masing *cluster*

f. Lakukan perulangan dari langkah c sampai e hingga anggota tiap *cluster* tidak ada yang berubah.

Dalam menentukan jumlah k optimal dapat menggunakan metode *Elbow*. Dalam menentukan jumlah *cluster* yang tepat, metode *elbow* akan mencari suatu titik yang membentuk siku dengan membandingkan hasil persentase berdasarkan jumlah setiap *cluster*. Ketika nilai suatu *cluster* dengan nilai *cluster* selanjutnya mengalami penurunan paling besar

atau menghasilkan sudut pada grafik maka jumlah nilai *cluster* tersebut yang optimal. Perbandingan didapatkan dengan menghitung *Sum of Square Error* (SSE) dari setiap nilai *cluster*. Nilai SSE akan semakin kecil ketika nilai *cluster* semakin besar. Perhitungan SSE menggunakan persamaan berikut (Dewi & Pramita, 2019) :

$$SSE = \sum_{K=1}^{K} \sum_{x_i} |x_i - c_k|^2 \quad (3)$$

Keterangan :
K = *cluster* ke-c
x_i = jarak data ke-i
c_k = *centroid* ke-i

Algoritma C4.5

Dalam membangun sebuah pohon keputusan, salah satunya dapat menggunakan algoritma C4.5. C4.5 telah diterapkan dalam kasus pada dunia nyata terutama pembuatan keputusan dalam hal medis, karena dapat menghasilkan akurasi klasifikasi yang baik dan dapat direpresentasikan secara sederhana. Kriteria *splitting* pada algoritma C4.5 menggunakan *gain ratio*, pemilihan atribut pertama dipilih berdasarkan atribut yang memiliki informasi penting yang dikomputasi pada *training set*, kemudian menyeleksi atribut tersebut dan seterusnya (Pradana, et al., 2018). Ada beberapa tahapan dalam membuat sebuah *decision tree* dengan algoritma C4.5 yaitu (Yulianti, et al., 2020) :

a. Information Gain

Information Gain dihitung dengan menggunakan persamaan berikut:

$$Gain\ (S, A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy\ (Si) \quad (4)$$

Keterangan:
S : himpunan kasus
A : atribut
$|S_i|$: jumlah kasus pada partisi ke-i
|S| : jumlah kasus dalam S
n : jumlah partisi atribut A

b. Entropy

Dalam menentukan infomation gain diperlukan nilai entropy dengan menggunakan persamaan berikut:

$$Entropy\ (S) = \sum_{i=1}^{n} - pi\ * \log_2 pi \quad (5)$$

Keterangan:
S : himpunan kasus
Pi : proporsi dari Si terhadap S
n : jumlah partisi S

c. Gain Ratio

Gain ratio adalah modifikasi dari information gain yang bertujuan untuk mengurangi bias atribut yang pada algoritma C4.5, dengan persamaan sebagai berikut:

$$Gain\ Ratio\ (a) = \frac{gain(a)}{split(a)} \quad (6)$$

Keterangan:
a : Atribut
Gain(a) : Nilai gain pada atribut a
Split(a) : Nilai split pada atribut a

d. Split Info

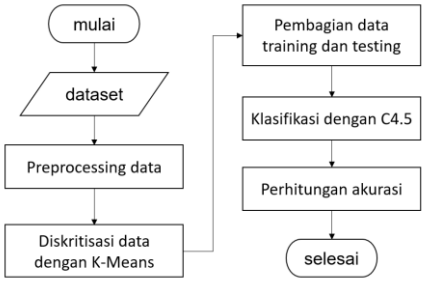
Dalam pemilihan akar (*node*) terdapat perhitungan split info terlebih dahulu sebelum mencari nilai gain ratio. Perhitungan split info menggunakan persamaan berikut:

$$Split\ Info\ (S, A) = -\sum_{i=1}^{n} \frac{S_i}{S} log_2 \frac{S_i}{S} \quad (7)$$

Keterangan:
S : Himpunan kasus
A : Atribut
Si : Jumalah sampel untuk atribut i

METODEOLOGI PENELITIAN

Model Yang Diusulkan



Gambar 1. *Flowchart* Model yang Diusulkan

Data sekunder penyakit ginjal kronis yang diambil dari *UCI Machine Learning* akan melewati tahapan *preprocessing data* terlebih dahulu. Terdapat beberapa tahapan dalam preprocessing data seperti data cleaning yang digunakan untuk membersihkan data dari *missing value*, transformasi data untuk mengubah data masih bersifat nominal menjadi numerik serta proses normalisasi data agar rentang data disemua atribut menjadi sama. Selanjutnya dilakukan proses diskritisasi data menggunakan K-Means pada data bertipe numerik (kontinu) seperti Usia, *Blood*

Pressure, Blood Glucose Random, Blood Urea, Serum Creatinine, Sodium, Potassium, Haemoglobin, Packed Cell Volume, White Blood Cell Count, dan *Red Blood Cell Count.* Setelah data diskritisasi akan dilakukan pembagain data *training* dan *testing* lalu dilanjutkan dengan klasifikasi dengan menggunakan algoritma C4.5. Dan yang terakhir akan dilakukan perhitungan akurasi.

Pengumpulan Data

Pada penelitian ini menggunakan data sekunder yaitu dataset *Chronic Kidney Disease* (CKD) yang didapatkan dari *UCI Machine Learning Repository*. Dataset ini memiliki 400 sampel data dengan pembagian jumlah data untuk setiap kelas yaitu CKD sebanyak 250 data dan NOTCKD sebanyak 150 data. Dataset ini memilki 25 atribut dimana 11 atribut bertipe numerikal dan 14 atribut bertipe kategorikal. Tabel 1 menunjukan deskripsi dari dataset *Chronic kidney disease*.

Tabel 1. Deskripsi atribut

No	Fitur	Deskripsi	Domain
1	*Age*	Usia	2 – 90
2	*Bp*	*Blood Pressure*	50 – 180
3	*Sg*	*Specific Gravity*	1,005 – 1,025
4	*Al*	*Albumin*	0 – 5
5	*Su*	*Sugar*	0 – 5
6	*Rbc*	*Red Blood Cells*	*Normal/Abnormal*
7	*Pc*	*Pus Cell*	*Normal/Abnormal*
8	*Pcc*	*Pus Cell Clumps*	*Present/NotPresent*
9	*Ba*	*Bacteria*	*Present/NotPresent*
10	*Bgr*	*Blood Glucose Random*	22 – 490
11	*Bu*	*Blood Urea*	1,5 – 391
12	*Sc*	*Serum Creatinine*	0,4 – 76
13	*Sod*	*Sodium*	4,5 – 163
14	*Pot*	*Potassium*	2,5 – 47
15	*Hemo*	*Haemoglobin*	3,1 – 17,8
16	*Pcv*	*Packed Cell Volume*	9 – 54
17	*Wc*	*White Blood Cell Count*	2200 – 26400
18	*Rc*	*Red Blood Cell Count*	2,1 – 8
19	*Htn*	*Hypertension*	*Yes/No*
20	*Dm*	*Diabetes Mellitus*	*Yes/No*
21	*Cad*	*Coronary Artery Disease*	*Yes/No*
22	*Appet*	*Appetite*	*Good/Poor*
23	*Pe*	*Pedal Edema*	*Yes/No*
24	*Ane*	*Anemia*	*Yes/No*
25	*Class*	*Class*	*Ckd/Nockd*

Pengolahan Data Awal

Sebelum data diproses akan dilakukan pengolahan data awal atau *preprocessing data* dimulai dari *data cleaning*, transformasi data dan tahapan diskritisasi data dengan *K-Means*.

a. *Data Cleaning*

Data cleaning dilakukan untuk menghapus *missing value* pada dataset. Pada dataset *Chronic Kidney Disease*, terdapat 242 data yang mengandung *missing value* dari 400 data yang ada. Penanganan terhadap missing value pada dataset dilakukan dengan menghapus data-data yang mengandung *missing value*. Sehingga didapatkan total data yang tidak mengandung *missing value* sebanyak 158 data dengan pembagian jumlah data untuk setiap kelasnya yaitu CKD sebanyak 43 data dan NOTCKD sebanyak 115 data.

b. Tranformasi Data

Tranformasi dilakukan dengan mengubah data yang bertipe nominal menjadi numerik. Data yang diubah yatitu atribut *specific gravity, red blood cells, pus cell, pus cell clumps, bacteria, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema* dan *anemia*. Setelah data yang bertipe nominal diubah menjadi numerik, dilanjutkan dengan proses normalisasi data. Pada dataset *Chronic kidney disease*, rentang nilai pada setiap atribut masih belum sama. Sehingga diperlukan normalisasi agar rentang nilai atau domain tiap atribut menjadi sama dengan rentang nilai [0,1] dengan menggunakan persamaan *Min-Max Normalization* dengan rumus sebagai berikut.

$$X_{new} = \frac{X - \min(x)}{\max(x) - \min(x)} \quad (8)$$

Evaluasi

Tahap evaluasi sistem dilakukan dengan menggunakan *confusion matrix*. Perhitungan akan dilakukan dengan membagi data menjadi data *training* dan data *testing* dengan perbandingan 70:30. Lalu akan dilanjutkan dengan pengukuran evaluasi dengan *confusion matrix.*

a. *Accuracy* merupakan ketepatan dari jumlah data yang diprediksi secara benar dengan menggunakan persamaan (9).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

b. *Precision* (P) merupakan ketepatan dari kasus yang positif yang diprediksi secara benar dengan menggunakan persamaan (10).

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

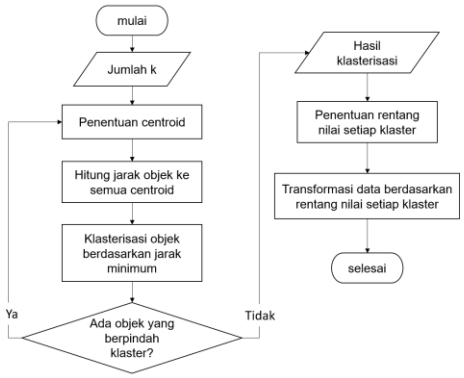
c. *Recall* (R) merupakan ketepatan dari kasus yang positif yang diidentifikasi secara benar dengan menggunakan persamaan (11).

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

IMPLEMENTASI SISTEM

Proses K-Means

Diskritisasi dilakukan dengan menggunakan *K-Means* pada atribut-atribut yang memilki data kontinu. Dimana diskritisasi akan dilakukan pada setiap atribut yang artinya proses *K-Means* akan dilakukan pada satu dimensi. Alur dari diskritiasasi dengan menggunakan *K-Means* dapat dilihat pada gambar 2.



Gambar 2. *Flowchart* Diskritisasi dengan *K-Means*

Langkah pertama dalam *K-Means* yaitu menentukan jumlah k atau klaster. Dimana jumlah klasiter ini nantinya akan menjadi jumlah *instance* suatu atribut yang akan digunakan dalam perhitungan C4.5. Selanjutnya menentukan *k-centroid* atau titik pusat klaster dengan mengambil data dari dataset secara acak lalu menghitung jarak setiap data pada masing-masing centroid menggunakan rumus *Euclidean Distance* dengan persamaan (1). Mengelompokan setiap data ke masing-masing klaster denga mencari jarak terdekat antara *centroid* dengan data dan menentukan nilai *centroid* baru berdasarkan nilai rata-rata semua data pada masing-masing klaster dengan menggunakan persamaan (2). Hal yang sama akan dilakukan ketika terdapat data yang berpindah klaster dengan membandingkan nilai centroid baru dengan nilai centroid sebelumnya. Jika nilainya berbeda maka lakukan perulangan, jika tidak perulangan berhenti. Hasil dari proses tersebut akan menghasilkan data-data yang sudah terklaster. Dari hasil klaster tersebut akan dibentuk rentang nilai di setiap klasternya dan dilanjutkan dengan mentransformasi data-data kontinu berdasarkan rentang tiap klaster yang bersesuaian.

Sebagai contoh, akan dilakukan proses diskritisasi pada atribut *age*. Atribut *age* memilki data kontinu dengan rentang nilai dari 6 sampai 83 setelah dilakukan *preprocesing data*. Selanjutnya data-data dari atribut tersebut akan masuk ke tahapan *K-Means* dengan nilai k = 3. Sehingga didapatkan hasil klasterisasi yang dapat dilihat pada tabel 2.

Tabel 2. Hasil Klasterisasi

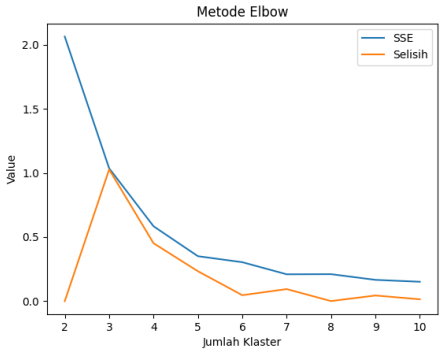
Klaster	Data	Rentang
Klaster 1	6, 12, 15, 17, 20, 21, ..., 34, 35, 35, 36	6 ≥ data ≤ 36
Klaster 2	37, 37, 38, 38, 39, 39, ..., 52, 52, 52, 53	37 ≥ data ≤ 53
Klaster 3	55, 55, 55, 56, 56, 58, ..., 75, 79, 80, 83	55 ≥ data ≤ 83

Dari hasil klasterisasi pada tabel 2, penentuan rentang nilai tiap klaster dilakukan dengan memilih nilai terbesar dan terkecil dari data pada masing-masing klaster. Selanjutnya akan dilakukan transformasi data pada atribut *age* berdasarkan rentang-rentang yang sudah didapatkan pada tabel 2. Sehingga pada atribut *age* kini memliki 3 buah instance yaitu 1, 2, dan 3.

Untuk menentukan jumlah k yang optimal dari setiap atribut, akan dilakukan perhitungan dengan metode *elbow*. Metode ini melakukan perbandingan berdasarkan nilai *Sum of Square Error* (SSE) dari masing-masing jumlah klaster dengan menggunakan persamaan (3). Ketika nilai suatu *cluster* dengan nilai *cluster* selanjutnya mengalami penurunan paling besar atau menghasilkan sudut pada grafik maka jumlah nilai *cluster* tersebut akan digunakan dalam proses klasifikasi. Perhitungan nilai SSE akan dilakukan pada jumlah klaster dari 2 sampai 10. Sebagai contoh, tabel 3 dan gambar 3 akan memperlihatkan nilai SSE beserta selisihnya pada atribut *age*.

Tabel 3. Hasil Perhitungan SSE

Jumlah k	SSE	Selisih
2	2.0826	0
3	1.0363	**1.0463**
4	0.5794	0.4569
5	0.3911	0.1883
6	0.2682	0.1228
7	0.2096	0.0586
8	0.1916	0.018
9	0.1842	0.0074
10	0.1361	0.0481



Gambar 3. Grafik Hasil Perhitungan SSE

Berdasarkan hasil perhitungan pada tabel 3 dan gambar 3, bahwa selisih terbesar didapatkan pada jumlah k adalah 3 dengan selisih sebesar 1.0463. Sehingga pada atribut *age* akan dilakukan proses diskritisasi pada jumlah k = 3. Perhitungan yang sama dilakukan pada seluruh atribut yang memiliki data numerikal (kontinu). Tabel 4 akan memperlihatkan nilai k optimal beserta rentang dari setiap klasternya pada seluruh atribut yang memiliki data kontinu.

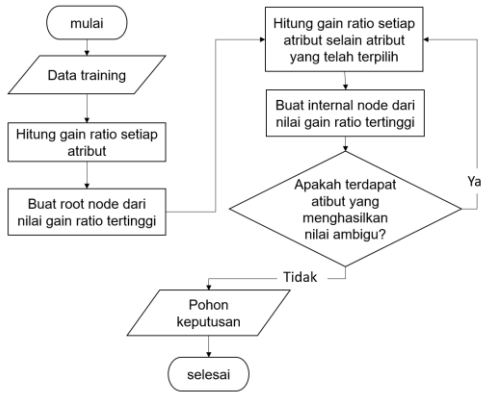
Tabel 4. Hasil Diskritisasi

Fitur	K	Rentang
Age	3	6-36, 37-53, 55-83
Bp	3	50-60, 70-70, 80-110
Bgr	6	70-87, 88-106, 107-125, 127-176, 210-303, 380-490
Bu	4	10-33, 34-73, 82-158, 309-163
Sc	3	0.4-2.7, 3.2-8.5, 9.2-15.2
Sod	3	111-125, 130-141, 142-150
Pot	3	2.5-4.2, 4.2-7.6, 47-47
Hemo	3	3.1-11.5, 12-15.2, 15.2-17.8
Pcv	4	9-26, 28-37, 39-46, 47-54
Wc	3	3800-8200, 8300-12800, 14600-26400
Rc	3	2.1-4, 4.1-5.3, 5.4-8

Tabel 4 memperlihatkan hasil diskritisasi dari setiap atribut, dimana jumlah k optimal pada atribut Usia, *Blood Pressure*, *Blood Glucose Random*, *Blood Urea*, *Serum Creatinine*, *Sodium*, *Potassium*, *Haemoglobin*, *Packed Cell Volume*, *White Blood Cell Count*, dan *Red Blood Cell Count* dengan masing-masing nilainya yaitu 3, 3, 6, 4, 3, 3, 3, 3, 4, 3, 3.

Proses C4.5

Setelah dilakukan diskritisasi, selanjutnya adalah proses klasifikasi dengan algoritma C4.5 untuk membentuk pohon keputusan berdasarkan data *training* yang digunakan untuk memprediksi kelas dari data testing. Adapun tahapan-tahapan klasifikasi menggunakan algoritma C4.5 yang dijelaskan pada gambar 4.



Gambar 4. *Flowchart* Algoritma C4.5

Langkah pertama yaitu menentukan data yang akan digunakan dalam proses pembentukan pohon keputusan. 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 (5). Dilanjutkan dengan perhitungan *information gain* dari setiap atribut dengan persamaan (4). Lalu lakukan perhitungan *split info* dengan persamaan (7) dan *gain ratio* dengan persamaan (6) pada tiap atribut. Selanjutnya

tentukan *root node* atau *node* yang terletak pada bagian paling atas dari pohon keputusan dengan cara mencari atribut yang mempunyai nilai *gain ratio* tertinggi. Kemudian bangun *rule* berdasarkan atribut yang terpilih tersebut. 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. Lalu tentukan nilai *gain ratio* tertinggi untuk dijadikan *internal node* atau *node* dari suatu percabangan. Kemudian bangun rule berdasarkan atribut yang terpilih tersebut. Jika atribut dari *internal node* belum signifikan menemukan kelas prediksi atau menghasilkan nilai ambigu, maka lakukan kembali perhitungan *gain ratio* dan penentuan *internal node* sampai *rule* yang dibangun memenuhi kriteria untuk menemukan kelas prediksi yang signifikan. Jika atribut sudah memenuhi kriteria maka perulangan berhenti dan pohon keputusan telah terbentuk.

Pengujian Sistem

Pengujian sistem dilakukan membagi data menjadi data *training* dan data *testing* dengan perbandingan 70:30. Dari 158 data yang digunakan dari hasil *preprocessing*, jumlah data *training* yang digunakan yaitu 110 data dan data *testing* 48 data. Pengujian sistem dilakukan dengan menggunakan *confusion matrix* dengan menghitung nilai *accuracy*, *recall* dan *precision*. Hasil pengujian dari sistem yang dibangun dapat dilihat pada tabel 5.

Tabel 5. Akurasi Sistem

Accuracy	97.92%
Recall	94.44%
Precision	100%

SIMPULAN

Peneltian ini membangun sistem untuk mendiagnosis penyakit ginjal kronis dengan mengggunakan algoritma C4.5 dan *K-Means Clustering* sebagai diskritisasi data. Diskritisasi dilakukan pada atribut yang memilki data numerik (kontinu). Dalam penentuan k optimal pada *K-Means* digunakan metode *elbow* dengan membandingkan nilai SSE dari setiap jumlah k dengan nilai yaitu 2 sampai 10. Sehingga didapatkan jumlah k optimal pada atribut Usia, *Blood Pressure*, *Blood Glucose Random*, *Blood Urea*, *Serum Creatinine*, *Sodium*, *Potassium*, *Haemoglobin*, *Packed Cell Volume*, *White Blood Cell Count*, dan *Red Blood Cell Count* dengan masing-masing nilainya adalah 3, 3, 6, 4, 3, 3, 3, 3, 4, 3, 3. Pengujian sistem dilakukan dengan menggunakan *confusion matrix* dan didapatkan nilai *accuracy*, *recall* dan *precision* yaitu 97.92%, 94.44% dan 100%.

DAFTAR PUSTAKA

[1] Amalia, H., 2018. PERBANDINGAN METODE DATA MINING SVM DAN NN UNTUK KLASIFIKASI PENYAKIT GINJAL KRONIS. Jurnal PILAR Nusa Mandiri, 14(1), pp. 1-6.

[2] Dewi, D. A. I. C. & Pramita, D. A. K., 2019. Analisis Perbandingan Metode Elbow dan Sillhouette pada Algoritma Clustering K-Medoids dalam Pengelompokan Produksi Kerajinan Bali. *JURNAL MATRIX*, 9(3), pp. 102-109.

[3] Fadilla, I., Adikara, P. P. & Perdana, R. S., 2018. Klasifikasi Penyakit Chronic Kidney Disease (CKD) Dengan Menggunakan Metode Extreme Learning Machine (ELM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , 2(10), pp. 3397-3405.

[4] Fahdia, M. R., 2020. PERBANDINGAN ALGORITMA KLASIFIKASI UNTUK PREDIKSI PENYAKIT LIVER. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 1(2), pp. 82-88.

[5] Handayani, I., 2019. PENERAPAN ALGORITMA C4.5 UNTUK KLASIFIKASI PENYAKIT DISK HERNIA DAN SPONDYLOLISTHESIS DALAM KOLUMNA VERTEBRALIS. *JASIEK*, 1(2), p. 83~88.

[6] Kurnianto, E. A., Cholissodin, I. & Santoso, E., 2018. Klasifikasi Penderita Penyakit Ginjal Kronis Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan*

Ilmu Komputer, 2(12), pp. 6597-6602.

[7] Pradana, A. C., Adiwijaya & Aditsania, A., 2018. Implementasi Algoritma Binary Particle Swarm Optimization (BPSO) dan C4.5 Decision Tree untuk Deteksi Kanker Berdasarkan Klasifikasi Microarray Data. *e-Proceeding of Engineering*, 5(3), pp. 7665-7682.

[8] Pujianto, U., Setiawan, A. L., Rosyid, H. A. & Salah, A. M. M., 2019. Comparison of Naïve Bayes Algorithm and Decision Tree C4.5 for Hospital Readmission Diabetes Patients using HbA1c Measurement. *Knowledge Engineering and Data Science (KEDS)*, 2(2), p. 58–71.

[9] Rahman, A. T., Anggrainingsih, R. & Wiranto, 2017. Coal Trade Data Clusterung Using K-Means (Case Study PT. Global Bangkit Utama). *ITSMART: Jurnal Ilmiah Teknologi dan Informasi*, 6(1), pp. 24-31.

[10] Rochman, M. I. A., Ratnawati, D. E. & Anam, . S., 2019. Penerapan Algoritme C4.5 untuk Klasifikasi Fungsi Senyawa Aktif Menggunakan Kode Simplified Molecular Input Line System (SMILES). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , 3(1), pp. 761-769.

[11] Yulianti, I., Saputra, R. A., Mardiyanto, M. S. & Rahmawati, A., 2020. Optimasi Akurasi Algoritma C4.5 Berbasis Particle Swarm Optimization dengan Teknik Bagging pada Prediksi Penyakit Ginjal Kronis. *Techno.COM*, 19(4), pp. 411-421.