Perbandingan Metode Klasterisasi Data Bertipe Campuran: *One-Hot-Encoding*, *Gower Distance*, dan *K-Prototype* Berdasarkan Akurasi
(Studi Kasus: *Chronic Kidney Disease Dataset*)

Zahra Rizky Fadilah [1]*, Arie Wahyu Wijayanto [2]*
* Politeknik Statistika STIS
212011430@stis.ac.id [1], ariewahyu@stis.ac.id [2]

Article Info

Article history:

Received 2023-07-12
Revised 2023-07-28
Accepted 2023-07-30

Keyword:

Clustering,
Gower Distance,
K-Prototype,
Mixed-Data Type,
One-Hot-Encoding.

ABSTRACT

This study aims to compare the one-hot-encoding method, Gower distance combined with k-means, DBSCAN, and OPTICS algorithms, and k-prototype for clustering mixed data types based on accuracy. The dataset used in this research is the chronic kidney disease (CKD) dataset sourced from the UCI Machine Learning Repository. Based on the evaluation using the silhouette index, it is found that k-prototype with the number of clusters k=2 is the most optimal clustering method because it provides the highest silhouette index value compared to the other four methods, with a value of 0,3796. Cluster 1 contains 175 observations, while cluster 2 contains 225 observations. When associated with the labels on the dataset, the clustering results provide an accuracy value of 81,25 percent.

I. PENDAHULUAN

Clustering merupakan proses pengelompokan sekumpulan objek menjadi beberapa kelompok atau *cluster* sehingga objek-objek dalam *cluster* tersebut memiliki kesamaan yang tinggi tetapi sangat berbeda dengan objek-objek dalam *cluster* lain [1]. Dua karakteristik utama yang diinginkan dari hasil *clustering* adalah homogenitas atau memaksimalkan kesamaan intra-*cluster*, serta heterogenitas atau memaksimalkan perbedaan antar-*cluster* [2]. Perbedaan dan kesamaan tersebut diukur berdasarkan kemiripan atribut yang seringkali melibatkan pengukuran jarak. Semakin dekat jarak, maka semakin mirip suatu objek dengan objek lain [3].

Pada umumnya, *clustering* hanya diterapkan pada *dataset* dengan atribut yang seluruhnya bertipe numerik atau seluruhnya bertipe kategorik. Metode *clustering* tradisional seperti *k-means* hanya cocok digunakan untuk data yang seluruh atributnya bertipe numerik dan tidak efektif jika diterapkan pada data bertipe kategorik [3]. Ketika atribut bertipe kategorik atau campuran, euclidean *distance* akan gagal dalam menangkap kemiripan objek [4]. Sementara itu, *k-modes* merupakan algoritma *clustering* yang populer digunakan untuk menangani data yang seluruh atributnya bertipe kategorik. Padahal kenyataannya seringkali ditemukan *dataset* dengan tipe data campuran, yaitu *dataset* yang memiliki atribut bertipe numerik dan kategorik. Misalnya, pada data kesehatan yang berisi atribut numerik seperti usia, tekanan darah, kadar hemoglobin, kadar gula dalam darah; serta atribut kategorik seperti jenis kelamin, golongan darah, status merokok, dll. Di sisi lain, apabila ingin menggunakan alternatif pengelompokan dengan metode klasifikasi, akan sangat tidak efisien jika harus memberi label pada setiap data dalam *dataset*, terlebih jika *dataset* tersebut berukuran besar. Oleh karena itu, diperlukan metode *clustering* yang dapat mengakomodir tipe data campuran.

K-prototype merupakan algoritma *clustering* yang dikembangkan dari *k-means* dan *k-modes* untuk menangani *clustering* pada data yang memuat atribut numerik dan kategorik. Metode ini menggabungkan ukuran jarak pada algoritma *k-means* dan *k-modes* [5]. *K-prototype* memiliki keunggulan dari sisi algoritmanya yang tidak terlalu kompleks dan kemampuannya untuk mengatasi data bertipe numerik dan kategorik. Namun sama seperti *k-means* dan *k-modes*, *k-prototype* sensitif terhadap inisialisasi pusat *cluster* sehingga menyebabkan solusi lokal optimum [6].

Selain *k-prototype*, *clustering* pada data campuran dapat ditangani dengan mengubah metrik jarak yang digunakan. Salah satu metrik jarak yang paling banyak digunakan untuk *clustering* data bertipe campuran adalah *Gower distance* [7].

Gower distance dapat dikombinasikan dengan metode *clustering* seperti *k-medoids*, DBSCAN, dan OPTICS dikarenakan ketiga metode tersebut memiliki kemampuan dalam memproses *precomputed distance matrix*. Dengan menerapkan metrik yang berbeda dalam menghitung jarak atribut bertipe numerik dan atribut bertipe kategorik, *Gower distance* digunakan untuk memastikan bahwa *k-medoids*, DBSCAN, dan OPTICS dapat diterapkan pada data bertipe campuran.

Di samping itu, terdapat cara yang sangat sederhana untuk menghitung jarak pada data bertipe campuran, yaitu dengan mengubah data kategorik menjadi *dummy* dan memperlakukannya sebagai numerik. Popoola *et al*. [2] menyebutkan bahwa metode tersebut dikenal sebagai *one-hot-encoding* dimana setiap kategori dalam variabel kategorik dikonversi menjadi variabel *biner*. Nilai 1 diberikan untuk suatu kategori tertentu, sementara nilai 0 diberikan pada kategori sisanya. Setelah itu, seluruh data dapat distandardisasi dan *Euclidean distance* dapat digunakan. Namun, metode tersebut dianggap naif karena tidak jarang atribut numerik hasil konversi dari kategorik kurang signifikan secara statistik. Selain itu, atribut hasil konversi cenderung didominasi oleh atribut kontinu selama *clustering* dan secara umum tidak ada skema pembobotan yang dapat mengatasi hal tersebut [2].

Oleh karena itu, penelitian ini bertujuan untuk melakukan perbandingan metode dalam menangani *clustering* data bertipe campuran berdasarkan akurasi. Adapun metode yang akan dibandingkan meliputi konversi variabel kategorik menjadi numerik (*one-hot-encoding*); kombinasi *Gower distance* dengan algoritma *k-medoids*, DBSCAN, dan OPTICS; serta metode *k-prototype*. Perbandingan metode *clustering* dalam penelitian ini dilakukan menggunakan metrik *silhouette index* yang mengukur seberapa baik objek masuk ke dalam *cluster*. Adapun *dataset* yang akan digunakan sebagai studi kasus adalah *chronic kidney disease* (CKD) yang memuat atribut campuran numerik dan kategorik.

Terdapat beberapa penelitian sebelumnya yang juga menggunakan *dataset* CKD tetapi menerapkan algoritma yang berbeda. Penelitian Revathy *et al*. [8] melakukan prediksi CKD menggunakan tiga algoritma *supervised learning* yaitu *decision tree*, *random forest*, dan *support vector machine*. Hasil penelitian menunjukkan bahwa *random forest* lebih baik dalam memprediksi CKD dibandingkan dua model lainnya. Kemudian, Antony *et al*. [9] menerapkan lima algoritma *unsupervised*, yaitu *k-means*, DBSCAN, *IForest*, dan *Autoencoder* untuk mengelompokkan pasien CKD dan non-CKD dengan menerapkan *one-hot-encoding*. Diperoleh hasil bahwa *k-means* memiliki performa terbaik dibandingkan tiga metode lainnya. Dalam penelitian Gopika *et al*. [10] juga diterapkan *one-hot-encoding* serta dilakukan perbandingan metode *k-means*, *k-medoids*, dan *fuzzy c-means* dengan jarak *Euclidean* untuk mengelompokkan *dataset* CKD. Ditemukan bahwa *fuzzy c-means* memberikan hasil yang lebih baik dibandingkan *k-means* dan *k-medoids*. Namun demikian, masih sedikit penelitian yang membandingkan antara *one-hot-encoding*, *Gower distance*, dan *k-prototype*. Oleh karena itu, peneliti tertarik untuk melakukan perbandingan ketiga metode tersebut dalam penelitian ini.

II. TINJAUAN PUSTAKA

A. Data Bertipe Campuran

Data bertipe campuran (*mixed data type*) merupakan *dataset* yang memiliki atribut bertipe numerik dan kategorik. Data bertipe campuran banyak ditemukan diantaranya pada data pasien maupun data nasabah yang mengajukan kredit ke bank. *Clustering* pada data bertipe campuran memerlukan metode khusus karena *clustering* tradisional seperti *k-means* hanya dapat digunakan untuk data yang seluruh atributnya bertipe numerik. Alternatif metode *clustering* yang dapat digunakan untuk menangani data bertipe campuran adalah *one-hot-encoding*, *Gower distance*, dan *k-prototype*. *One-hot-encoding* mengkonversi atribut kategorik menjadi numerik (*dummy*), sementara *Gower distance* dan *k-prototype* membedakan atau memisahkan penghitungan jarak untuk atribut numerik dan kategorik.

B. K-NN Imputation

K-Nearest Neighbour (k-NN) merupakan salah satu algoritma *supervised learning* untuk mengklasifikasikan data berdasarkan k objek terdekat. Selain digunakan untuk klasifikasi, k-NN juga dapat digunakan untuk menangani *missing data*. Proses tersebut diawali dengan menentukan jumlah tetangga atau observasi terdekat yang disimbolkan dengan k. Kemudian menghitung jarak terhadap setiap observasi yang tidak mengandung *missing data* menggunakan jarak *Euclidean*. Selanjutnya, menentukan k observasi terdekat berdasarkan nilai jarak yang terpendek. Imputasi *missing data* dilakukan dengan menghitung nilai estimasi rata-rata tertimbang pada k observasi terdekat yang tidak mengandung *missing data* menggunakan formula berikut.

$$\bar{x}_j = \frac{\sum_{k=1}^{K} w_k v_k}{\sum_{k=1}^{K} w_k} \qquad (1)$$

dimana $\bar{x}_j$ adalah nilai estimasi rata-rata tertimbang, v_k adalah nilai observasi terdekat ke-k pada variabel yang mengandung *missing data*, K adalah jumlah observasi terdekat yang digunakan, w_k adalah bobot observasi terdekat ke-K dengan rumus $w_k = 1/(d(x_{ak}x_{bk})^2)$ dimana $d(x_{ak}x_{bk})$ adalah jarak observasi K [11].

C. Standardisasi Data

Standardisasi dilakukan ketika atribut-atribut numerik dalam *dataset* memiliki satuan yang berbeda secara signifikan. Proses standardisasi dilakukan dengan mengkonversi data asli menjadi *z-score*, dengan rumus sebagai berikut.

$$z = \frac{x-\mu}{\sigma} \qquad (2)$$

dimana z adalah data hasil standardisasi, x adalah data asli, μ adalah rata-rata sampel, dan σ adalah standar deviasi sampel

[9]. Dengan demikian, data hasil standardisasi akan memiliki nilai rata-rata 0 dan standar deviasi 1.

D. Elbow Method

Elbow method merupakan metode yang digunakan untuk menentukan jumlah *cluster* optimal dengan melihat pada jumlah *cluster* berapa akan terbentuk siku. Hal tersebut ditentukan dengan menghitung SSE (*sum of square*) dari masing-masing jumlah *cluster*, dimana semakin besar jumlah *cluster* k maka nilai SSE akan semakin kecil. Nilai SSE untuk setiap jumlah *cluster* dapat disajikan secara visual dalam bentuk grafik. Jumlah *cluster* dikatakan optimum ketika grafik menunjukkan penurunan paling tajam sehingga membentuk sebuah siku. Adapun formula untuk menghitung SSE dapat dirumuskan sebagai berikut.

$$SSE = \sum_{k=1}^{k} \sum_{i=1}^{n} \|x_i - u_k\|^2 \tag{3}$$

Keterangan:
k = jumlah *cluster*
x_i = nilai atribut data ke-i
u_k = nilai atribut titik pusat cluster i

E. K-Means

K-Means merupakan salah satu algoritma *partitioning-based clustering*, yaitu metode *clustering* yang mengelompokkan objek-objek ke dalam beberapa *cluster* sedemikian rupa sehingga setiap *cluster* harus berisi minimal satu objek. Umumnya, metode ini melakukan pengelompokan secara eksklusif, artinya setiap objek hanya boleh menjadi anggota pada satu *cluster* [1]. Sesuai dengan namanya, algoritma *k-means* mendefinisikan pusat (*centroids*) dari sebuah *cluster* sebagai nilai rata-rata (*mean*) dari objek-objek di dalam *cluster*. Tahapan *clustering* dengan metode k-means adalah sebagai berikut.

1. Memilih k objek secara acak dimana masing-masing objek mewakili rata-rata atau pusat *cluster*.
2. Mengelompokkan setiap objek yang tersisa ke dalam *cluster* yang paling mirip berdasarkan jarak antara objek dengan pusat *cluster* yang telah ditetapkan sebelumnya. Terdapat beberapa jarak yang dapat digunakan, seperti *Euclidean*, *Manhattan*, *Minkowski*, dll. Adapun dalam penelitian ini digunakan jarak *Euclidean* dengan rumus sebagai berikut.

$$d(X_j, Z_i) = \sqrt{\sum_{l=1}^{p} (x_{jl} - z_{il})^2} \tag{4}$$

Keterangan:
x_{jl} = nilai atribut l pada data ke-j
z_{il} = nilai atribut l pada *centroid* ke-i

3. Memperbarui *centroids* dengan menghitung nilai rata-rata dari objek-objek di setiap *cluster*.
4. Ulangi langkah 2-3 hingga keanggotaan objek dalam *cluster* tidak berubah.

F. One-Hot-Encoding

One-hot-encoding merupakan metode yang paling sederhana untuk mengklasterkan data bertipe campuran. Pada prinsipnya, metode ini dilakukan dengan mengkonversi setiap atribut kategorik menjadi *dummy* dimana nilai 1 diberikan untuk kategori tertentu sedangkan nilai 0 diberikan pada kategori sisanya sehingga seluruh atribut hasil konversi dapat diperlakukan sebagai numerik. Dengan demikian, metode *k-means* dengan penghitungan jarak menggunakan *Euclidean distance* dapat diterapkan pada data hasil konversi tersebut [2].

G. Gower Distance

Gower [7] mengembangkan cara yang paling populer untuk mengukur kesamaan antarobjek dengan atribut bertipe data campuran. *Gower distance* menghitung jarak untuk atribut numerik dan kategorik secara terpisah kemudian menjumlahkan hasilnya, atau dapat dirumuskan sebagai berikut.

$$\delta_G(X, Y) = \frac{\sum_{j=1}^{m} w_j f_j(x_j, y_j)}{\sum_{j=1}^{m} w_j} \tag{5}$$

dimana $f_j(x_j, y_j) = |x_j - y_j|/r_j$ jika j kontinu (r_j adalah *range* sampel dari atribut *j*), dan *simple matching* jika *j* kategorik, m adalah jumlah atribut, dan w_j adalah pembobot untuk masing-masing atribut. Untuk keperluan *clustering* data bertipe campuran, *Gower distance* dapat dikombinasikan dengan metode *clustering* seperti *k-medoids*, DBSCAN, dan OPTICS. Dalam hal ini, penghitungan jarak dengan *Gower distance* berperan dalam memastikan bahwa metode *clustering* seperti *k-medoids*, DBSCAN, dan OPTICS dapat diaplikasikan pada data bertipe campuran.

H. K-Medoids (PAM)

K-Medoids atau yang dikenal juga dengan *Partitioning Around Medoids* (PAM) merupakan solusi dari algoritma *k-means* yang sensitif terhadap *outlier*, dimana ketika *outlier* tersebut dikelompokkan ke dalam *cluster* maka nilai rata-rata *cluster* akan berubah secara drastis dan memengaruhi pengelompokan objek lain. Dibandingkan menggunakan nilai rata-rata sebagai pusat *cluster*, algoritma *k-medoids* menggunakan objek pada kumpulan objek untuk merepresentasikan *cluster*, masing-masing *cluster* diwakili oleh satu objek [1]. Objek yang terpilih untuk merepresentasikan *cluster* disebut *medoids*. Setiap objek yang tersisa dikelompokkan ke dalam *cluster* berdasarkan kemiripannya dengan *medoid*. Proses ini diulang hingga keanggotaan objek dalam *cluster* tidak berubah.

I. DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) merupakan salah satu *density-based clustering* yang mengidentifikasi *cluster* berdasarkan parameter *Eps* dan *MinPts*, dimana *Eps* adalah radius yang menentukan batas *cluster*, sedangkan *MinPts* adalah jumlah minimal objek untuk bisa dianggap sebagai *cluster*. Objek yang memenuhi *Eps* dan *MinPts* disebut *core*. Sementara *border* bukanlah *core*, tetapi objek yang masih cukup dekat dengan *core*. Sedangkan objek-objek yang tidak masuk dalam *cluster* dan tersisa di akhir pengelompokan disebut *noise* [12]. DBSCAN tidak memerlukan input jumlah *cluster* k dan *robust* terhadap *noise*, tetapi tidak cocok untuk data berdimensi tinggi atau untuk *cluster* dengan kepadatan (*density*) yang berbeda-beda [13].

J. Tipping Point

Tipping point merupakan suatu teknik yang bertujuan untuk mendapatkan nilai parameter *Eps* yang optimal pada DBSCAN. Ide utama teknik ini adalah menghitung jarak rata-rata dari setiap objek ke k tetangga terdekatnya. Adapun nilai k akan ditentukan oleh peneliti. Nilai tersebut nantinya akan menjadi parameter *MinPts* dalam DBSCAN. Berdasarkan jarak yang telah dihitung sebelumnya, dapat dibentuk *kNN-distance plot* dimana sumbu x merepresentasikan objek yang telah diurutkan berdasarkan jarak ke k tetangga terdekat dan sumbu y merepresentasikan besarnya jarak tersebut. Suatu titik yang mengalami perubahan kemiringan plot akan menjadi nilai *Eps* optimal yang signifikan [14]. Titik tersebut ditentukan berdasarkan posisi *knee*, yaitu posisi dimana perubahan tajam terjadi di sepanjang plot.

K. OPTICS

Ordering points to identify the clustering structure (OPTICS) merupakan *density-based clustering* yang juga mengidentifikasi *cluster* berdasarkan parameter *Eps* dan *MinPts*. Namun, OPTICS mampu mendeteksi *cluster* dengan kepadatan yang beragam dan cocok untuk data berdimensi tinggi. Untuk setiap objek yang diproses, OPTICS mencatat urutan pemrosesannya dan menghitung *reachability distances* ke *core* terdekat. OPTICS dapat menghasilkan *reachability plot* dimana setiap palung atau lembah pada plot tersebut merepresentasikan suatu *cluster* [13].

L. K-Prototype

K-Prototype merupakan salah satu algoritma *partitioning-based clustering*. Algoritma ini merupakan pengembangan dari *k-means* untuk menangani *clustering* pada data dengan atribut bertipe campuran numerik dan kategorik. Pengembangan mendasar pada algoritma *k-prototype* terletak pada pengukuran kesamaan antar objek dengan *centroid*-nya (*prototype*), dimana data numerik dihitung menggunakan jarak *Euclidean* sedangkan data kategorik dihitung menggunakan jarak *k-modes*. Ukuran jarak yang digunakan pada *k-prototype* adalah sebagai berikut.

$$d_{ij} = \sum_{k=1}^{p}\left(x_{ik} - x_{jk}\right)^2 + \gamma \sum_{k=p+1}^{p+m} \delta(x_{ik}, x_{jk}) \quad (6)$$

Keterangan:
d_{ij} = ukuran jarak antara objek i dan j
$\sum_{k=1}^{p}\left(x_{ik} - x_{jk}\right)^2$ = ukuran jarak untuk data numerik
γ = penyeimbang jarak
$\sum_{k=p+1}^{p+m} \delta(x_{ik}, x_{jk})$ = ukuran jarak untuk data kategorik

Tahapan algoritma *k-prototype* dimulai dengan menentukan banyaknya *cluster* k yang akan dibentuk. Kemudian, menentukan k *prototype* awal sebagai pusat pada masing-masing *cluster*. Selanjutnya, dilakukan penghitungan jarak setiap observasi dalam *dataset* terhadap *cluster* awal dan mengalokasikannya berdasarkan jarak terdekat. Berikutnya, dilakukan penghitungan pusat *cluster* (*prototype*) baru dan merealokasikan setiap data terhadap *prototype* baru berdasarkan jarak terdekat. Proses ini diulang hingga tidak ada perubahan keanggotaan dalam *cluster* [15].

M. Silhouette Index

Dalam menentukan jumlah *cluster* optimal serta memilih metode *clustering* terbaik untuk data bertipe campuran, penelitian ini melakukan perbandingan keakuratan *cluster* menggunakan *silhouette index*. *Silhouette index* merupakan salah satu teknik validasi *cluster* internal yang paling banyak digunakan. *Silhouette index* menggunakan *silhouette coefficient* yang mengombinasikan *cohesion* dan *separation*. *Cohesion* digunakan untuk mengukur jarak antar objek dalam *cluster* yang sama, sedangkan *separation* digunakan untuk mengukur jarak antar *cluster*.

Silhouette coefficient dihitung dengan persamaan berikut.

$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)} \quad (7)$$

Keterangan:
$s(i)$ = nilai *silhouette coefficient*
$a(i)$ = nilai rata-rata jarak data dengan semua data lain dalam suatu cluster
$b(i)$ = nilai rata-rata jarak data dengan semua data lain pada cluster lain

Silhouette coefficient memiliki nilai dalam rentang -1 hingga 1. Nilai *silhouette coefficient* dikategorikan baik jika bernilai positif, saat nilai $a(i) < b(i)$ dan nilai $a(i)$ mendekati nol [16]. besar nilai *silhouette coefficient*, semakin baik *cluster* yang terbentuk.

III. METODE

Penelitian ini menggunakan *dataset Chronic Kidney Disease* (CKD) yang diperoleh melalui *UCI Machine Learning Repository* [17]. *Dataset* tersebut bersumber dari Dr. P. Soundarapandian. M. D., D. M. yang *merupakan Senior Consultant Nephrologist* di Apollo Hospitals, Managiri, Madurai Main Road, Karaikudi, Tamilnadu, India (Juli 2015). *Dataset* tersebut memuat 400 observasi dengan 25 atribut, terdiri atas 11 atribut numerik, 13 atribut kategorik, serta 1 label (*class*). Namun, label tidak digunakan dalam *clustering* karena *clustering* merupakan salah satu metode *unsupervised learning* yang tidak membutuhkan label pada *dataset* yang digunakan. Adapun rincian atribut dalam *dataset* yang digunakan dapat dilihat pada Tabel I.

TABEL I
RINCIAN ATRIBUT DALAM DATASET CHRONIC KIDNEY DISEASE

No	Atribut	Keterangan	Tipe Data
1	*age*	*age*	Numerik (tahun)
2	*bp*	*blood pressure*	Numerik (mm/Hg)
3	*sg*	*specific gravity*	Kategorik (1,005; 1,010; 1,015; 1,020; 1,025)
4	*al*	*albumin*	Kategorik

			(0,1,2,3,4,5)
5	*su*	*sugar*	Kategorik (0,1,2,3,4,5)
6	*rbc*	*red blood cells*	Kategorik (*normal, abnormal*)
7	*pc*	*pus cell*	Kategorik (*normal, abnormal*)
8	*pcc*	*pus cell clumps*	Kategorik (*present, notpresent*)
9	*ba*	*bacteria*	Kategorik (*present, notpresent*)
10	*bgr*	*blood glucose random*	Numerik (mgs/dl)
11	*bu*	*blood urea*	Numerik (mgs/dl)
12	*sc*	*serum creatinine*	Numerik (mgs/dl)
13	*sod*	*sodium*	Numerik (mEq/L)
14	*pot*	*potassium*	Numerik (mEq/L)
15	*hemo*	*hemoglobin*	Numerik (gms)
16	*pcv*	*packed cell volume*	Numerik
17	*wc*	*white blood cell count*	Numerik (*cell/cumm*)
18	*rc*	*red blood cell count*	Numerik (*millions/cmm*)
19	*htn*	*hypertension*	Kategorik (*yes, no*)
20	*dm*	*diabetes mellitus*	Kategorik (*yes, no*)
21	*cad*	*coronary artery disease*	Kategorik (*yes, no*)
22	*appet*	*appetite*	Kategorik (*good, poor*)
23	*pe*	*pedal edema*	Kategorik (*yes, no*)
24	*ane*	*anemia*	Kategorik (*yes, no*)
25	*class*	*classification*	Kategorik (*ckd, notckd*)

Analisis dalam penelitian ini menerapkan proses *data mining* dengan mengacu pada enam tahapan dalam *Cross-Industry Standard Process for Data Mining* (CRISP-DM), yaitu *business understanding, data understanding, data preparation, modelling, evaluation,* dan *deployment*.

Business understanding merupakan tahapan awal dalam *data mining* untuk memahami tujuan dan kebutuhan bisnis, kemudian menerjemahkannya ke dalam permasalahan *data mining* dan menyusun rancangan awal untuk mencapai tujuan tersebut. *Business understanding* dalam penelitian ini adalah melakukan klasterisasi pasien terdiagnosis penyakit gagal ginjal kronis untuk membantu para tenaga medis dalam memprediksi penyakit tersebut pada tahap awal secara efisien, bahkan sebelum diagnosis klinis diajukan. Hal ini juga bertujuan untuk meminimalkan biaya dari serangkaian tes yang cukup panjang dalam mendiagnosis penyakit gagal ginjal.

Data understanding dilakukan dengan mengumpulkan data, mengidentifikasi *missing data, noise, outlier,* serta menemukan pola menarik dalam data. Temuan-temuan tersebut nantinya akan diproses lebih lanjut dalam tahap selanjutnya.

Data preparation mencakup proses untuk memperoleh *dataset* yang siap dimodelkan, dengan cara melakukan transformasi dan pembersihan data. *Data preparation* dilakukan dengan menyesuaikan tipe data, memperbaiki nilai yang aneh, serta melakukan imputasi pada *missing data*.

Modelling merupakan tahapan dimana teknik pemodelan diterapkan dan dilakukan kalibrasi pada parameter model sehingga diperoleh nilai yang optimal. Dalam penelitian ini akan dibangun beberapa model *clustering* untuk menangani data campuran, yang selanjutnya akan dibandingkan performanya pada tahap evaluasi. Terdapat lima model yang akan dibangun, yaitu *k-means* dengan mengubah atribut kategorik menjadi numerik, *k-medoids* (PAM) dengan *Gower distance*, DBSCAN dengan *Gower distance*, OPTICS dengan *Gower distance*, serta *k-prototype*.

Evaluation dilakukan untuk menilai apakah model yang dihasilkan telah memenuhi tujuan bisnis. Evaluasi dalam penelitian ini dilakukan dengan mengukur validasi *cluster* yang dihasilkan dari beberapa model *clustering* yang telah dibangun. Metrik evaluasi yang digunakan dalam penelitian ini adalah *silhouette index*.

Deployment merupakan tahapan akhir dimana model yang dihasilkan disajikan kepada pengguna, yang dalam penelitian ini *deployment* dilakukan dengan penyajian beberapa visualisasi hasil *clustering*.

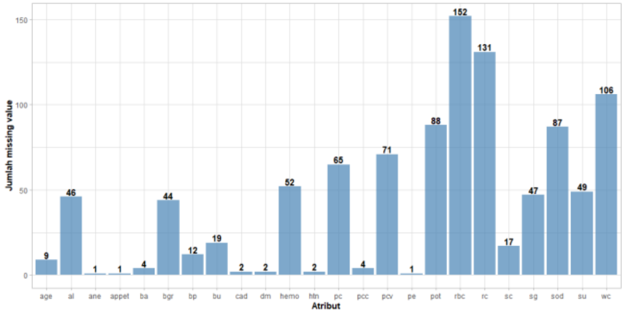
IV. HASIL DAN PEMBAHASAN

A. Data Preparation

Pada tahap *data preparation*, dilakukan beberapa proses untuk memperoleh data yang siap diklasterkan. Tahap ini diawali dengan melakukan pengecekan terhadap struktur dan isian data. Ditemukan terdapat isian data yang aneh pada atribut *dm* dan *cad*, dimana seharusnya kedua atribut tersebut hanya memiliki dua kemungkinan isian, yaitu “yes” dan “no”. Namun, ditemukan isian “ yes”, “\tyes”, dan “\tno” pada atribut *dm*. Sementara pada atribut *cad* ditemukan isian “\tno”. Oleh karena itu, dilakukan perbaikan terhadap isian aneh tersebut, dimana “ yes” dan “\tyes” diubah menjadi “yes” sedangkan “\tno” diubah menjadi “no”.

Selain itu, ditemukan terdapat beberapa atribut dengan tipe data yang belum sesuai, seperti atribut *sg, al, su* yang merupakan atribut kategorik namun memiliki tipe data *numeric*. Kemudian, atribut *pcv, wc, rc* yang merupakan atribut numerik tetapi memiliki tipe data *chr*. Sementara itu, atribut-atribut kategorik lainnya masih bertipe data *chr*. Oleh karena itu, dilakukan perbaikan tipe data, dimana atribut kategorik seluruhnya dikonversi ke dalam tipe data *factor*, sedangkan atribut numerik seluruhnya dibuat bertipe *numeric*.

Selanjutnya, ditemukan isian kosong pada beberapa atribut kategorik, dimana isian kosong tersebut tidak terdeteksi sebagai *missing value*. Oleh karena itu, seluruh isian kosong pada atribut kategorik diisi dengan *NA*, sehingga terdeteksi sebagai *missing value* dan akan dilakukan imputasi. Berikut rincian jumlah data yang *missing* pada setiap atribut dalam *dataset* setelah isian kosong atribut kategorik diisi dengan *NA*.

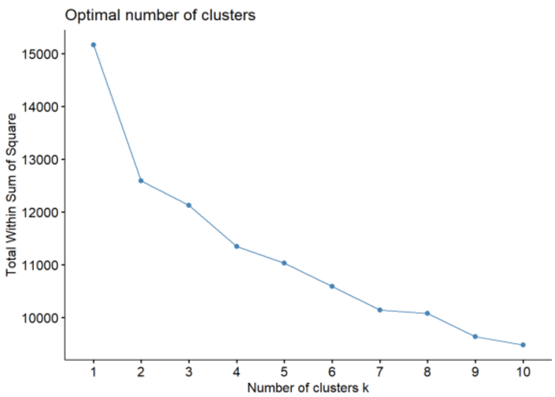


Gambar 1. *Missing value* pada setiap atribut

Untuk menangani *missing value* tersebut, dilakukan imputasi dengan metode *5-NN*. Setelah dilakukan imputasi dan diperoleh data yang lengkap, data tersebut dapat digunakan dalam *clustering*.

B. *K-Means dengan Konversi Atribut (one-hot-encoding)*

Skenario *clustering* yang pertama adalah menggunakan metode *k-means* dengan terlebih dahulu melakukan konversi atribut kategorik menjadi numerik (*biner*). Selanjutnya, dilakukan standardisasi pada data tersebut dan menghitung metrik jarak menggunakan *Euclidean distance*. Penentuan jumlah *cluster* menggunakan *elbow method* memberikan hasil sebagai berikut.



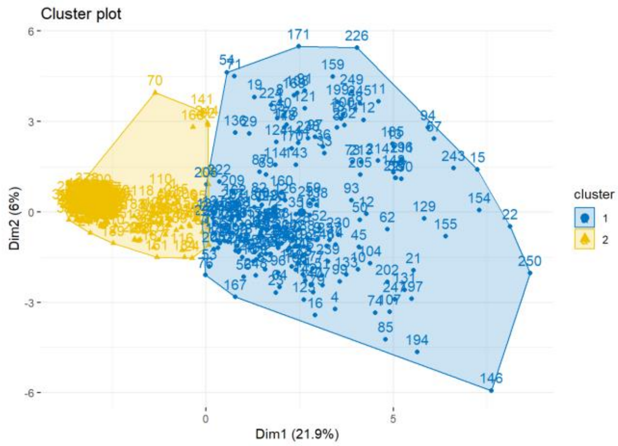
Gambar 2. *Elbow method* pada *k-means*

Berdasarkan Gambar 2, terlihat bahwa grafik mengalami penurunan secara drastis pada k=2. Maka, jumlah *cluster* 2 diperkirakan merupakan jumlah *cluster* optimum. Untuk meminimalkan subjektivitas, dilakukan simulasi *k-means* menggunakan 2 hingga 5 *cluster* untuk selanjutnya dibandingkan pada jumlah *cluster* berapa *silhouette index* memiliki nilai tertinggi. Berikut *silhouette index* yang dihasilkan dari simulasi jumlah *cluster* pada *k-means*.

TABEL II
SILHOUETTE INDEX HASIL SIMULASI K-MEANS

Jumlah *cluster*	***Silhouette index***
2	**0,2238**
3	0,2109
4	0,1972
5	0,2002

Berdasarkan Tabel II, *k-means* dengan jumlah *cluster* k=2 menghasilkan nilai *silhouette index* tertinggi, yaitu sebesar 0,2238. Maka, k=2 dipilih sebagai jumlah *cluster* optimum. Hasil *k-means clustering* dengan k=2 menunjukkan bahwa *cluster* 1 memuat 197 observasi, sedangkan *cluster* 2 memuat 203 observasi. Berikut visualisasi *k-means clustering* dengan k=2.



Gambar 3. Visualisasi *k-means* dengan k=2

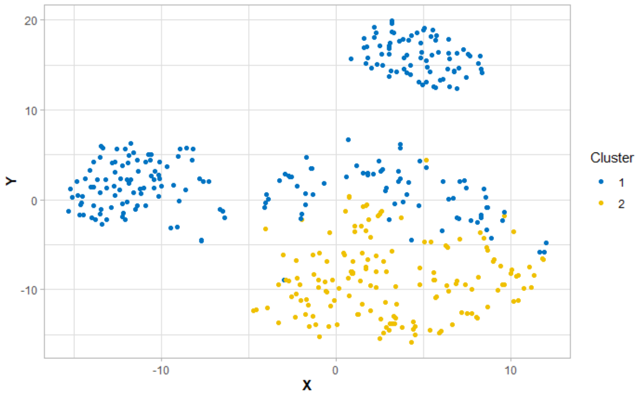
C. *K-Medoids (PAM) dengan Gower Distance*

Skenario *clustering* yang kedua adalah *k-medoids* (PAM) dengan *Gower distance*. Proses *clustering* diawali dengan melakukan standardisasi pada atribut numerik, kemudian dilakukan penghitungan jarak menggunakan *Gower distance*. Selanjutnya, *Gower distance* yang telah diperoleh menjadi input dalam *clustering* PAM. Penentuan jumlah *cluster* optimum pada PAM, juga dilakukan dengan simulasi menggunakan 2 hingga 5 *cluster* untuk selanjutnya dibandingkan pada jumlah *cluster* berapa *silhouette index* memiliki nilai tertinggi. Berikut *silhouette index* yang dihasilkan dari simulasi jumlah *cluster* pada PAM.

TABEL III
SILHOUETTE INDEX HASIL SIMULASI PAM DENGAN GOWER DISTANCE

Jumlah *cluster*	***Silhouette index***
2	**0,3657**
3	0,3446
4	0,1260
5	0,1628

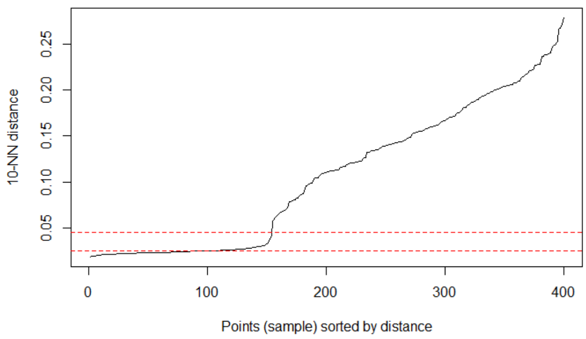
Berdasarkan Tabel III, k=2 dipilih sebagai jumlah *cluster* optimum karena menghasilkan nilai *silhouette index* tertinggi, yaitu sebesar 0,3657. Hasil *clustering* PAM dengan k=2 memberikan hasil bahwa *cluster* 1 memuat 242 observasi, sedangkan *cluster* 2 memuat 158 observasi. Berikut visualisasi *clustering* PAM dengan k=2.



Gambar 4. Visualisasi PAM dengan k=2

D. DBSCAN dengan Gower Distance

Skenario ketiga adalah *clustering* DBSCAN dengan *Gower distance*. Sama seperti skenario kedua, proses *clustering* diawali dengan melakukan standardisasi pada atribut numerik, kemudian dilakukan penghitungan jarak menggunakan *Gower distance*. Selanjutnya, *Gower distance* yang telah diperoleh menjadi input dalam *clustering* DBSCAN. *Clustering* dengan DBSCAN memerlukan parameter *Eps* dan *MinPts*. Dalam penelitian ini ditentukan *MinPts* = 10, sementara *Eps* optimal ditentukan menggunakan teknik *tipping point*. Visualisasi *10-NN distance* dapat dilihat pada gambar berikut.



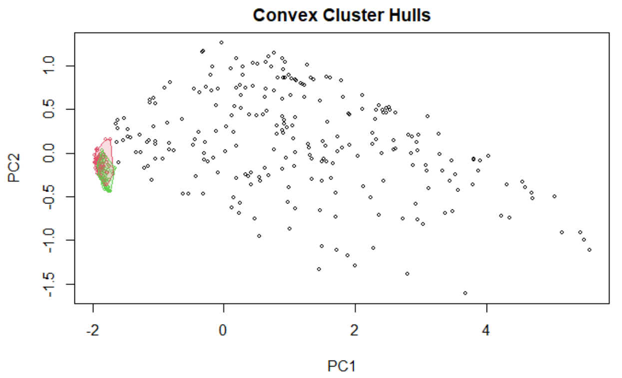
Gambar 5. 10-NN *distance plot*

Berdasarkan Gambar 5, plot mengalami perubahan tajam di sekitar jarak 0,025 hingga 0,045. Oleh karena itu, akan dilakukan simulasi DBSCAN dengan *Eps* 0,025 hingga 0,045 untuk selanjutnya dibandingkan nilai *silhouette index* yang dihasilkan.

TABEL IV
SILHOUETTE INDEX HASIL SIMULASI DBSCAN DENGAN GOWER DISTANCE

MinPts	*Eps*	*Silhouette index*
10	0,025	0,1290
	0,030	0,1374
	0,035	**0,1402**
	0,040	**0,1402**
	0,045	**0,1402**

Tabel IV menunjukkan bahwa nilai *silhouette index* tertinggi, yaitu sebesar 0,1402, dihasilkan pada saat *MinPts*=10 dan *Eps* sebesar 0,035; 0,040; atau 0,045. Dalam penelitian ini akan digunakan *MinPts*=10 dan *Eps*=0,040 sebagai parameter dalam DBSCAN. Hasil *clustering* DBSCAN dengan *parameter* tersebut menghasilkan 2 *cluster*, dimana *cluster* 1 berisi 79 objek, *cluster* 2 berisi 75 objek, dan 246 objek lainnya merupakan *noise*.



Gambar 6. Visualisasi DBSCAN dengan MinPts =10 dan Eps = 0,10

Gambar 6 menunjukkan bahwa sebagian besar objek teridentifikasi sebagai *noise*. Hal ini disebabkan *Eps* pada DBSCAN tidak adaptif terhadap kepadatan (densitas) data. Sementara pada plot diatas terlihat bahwa objek yang masuk *cluster* memiliki kepadatan yang sangat tinggi, sedangkan objek-objek yang teridentifikasi sebagai *noise* memiliki kepadatan yang rendah (renggang). Hal ini menjadi salah satu kelemahan dari DBSCAN yang akan dikoreksi oleh OPTICS pada skenario berikutnya.

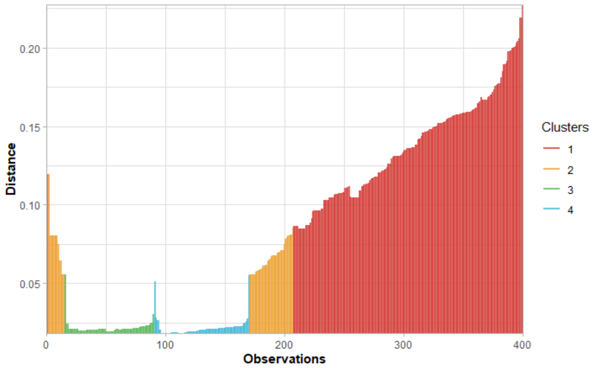
E. OPTICS dengan Gower Distance

Skenario keempat adalah *clustering* OPTICS dengan *Gower distance*. Sama seperti skenario kedua dan ketiga, proses *clustering* diawali dengan melakukan standardisasi pada atribut numerik, kemudian dilakukan penghitungan jarak menggunakan *Gower distance*. Selanjutnya, *Gower distance* yang telah diperoleh menjadi input dalam *clustering* OPTICS. Dalam penelitian ini, ditetapkan *MinPts* = 10 sebagai parameter dalam OPTICS. Sementara itu, ekstraksi *cluster* dilakukan dengan simulasi nilai X_i antara 0,01 hingga 0,05. Berikut perbandingan nilai *silhouette index* hasil simulasi X_i.

TABEL V
SILHOUETTE INDEX HASIL SIMULASI OPTICS DENGAN GOWER DISTANCE

MinPts	X_i	*Silhouette index*
10	0,01	-0,0594
10	0,02	0,0264
10	0,03	0,1152
10	0,04	0,1279
10	**0,05**	**0,1343**

Berdasarkan Tabel V, nilai *silhouette index* tertinggi yaitu sebesar 0,1343 dihasilkan pada saat $X_i = 0{,}05$. Dengan menggunakan $X_i = 0{,}05$ dapat diekstrak sebanyak 4 *cluster* tanpa *noise*, dimana 194 observasi masuk dalam *cluster* 1; 51 observasi masuk dalam *cluster* 2; 75 observasi masuk dalam *cluster* 3; dan 80 observasi masuk dalam *cluster* 4. Berikut *reachability plot* yang dihasilkan.

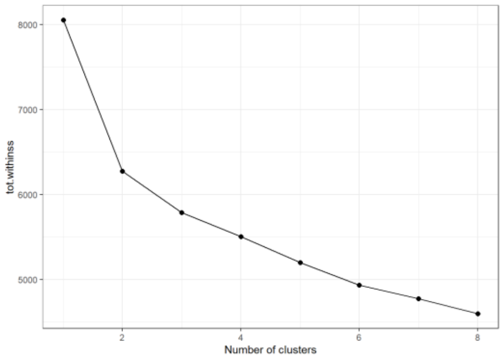


Gambar 7. *Reachability plot* OPTICS dengan *MinPts* = 10 dan X_i= 0,05

Berbeda dengan DBSCAN, *Eps* pada OPTICS bersifat adaptif, yaitu mampu menyesuaikan dengan kepadatan data. Ketika datanya renggang, *Eps* akan melebar. Sebaliknya, ketika datanya padat maka *Eps* akan mengecil. Hal ini terlihat bahwa pada DBSCAN banyak objek yang menjadi *noise*, sedangkan pada OPTICS seluruh objek masuk dalam *cluster*.

F. K-Prototype

Skenario *clustering* kelima adalah *k-prototype*. Penentuan jumlah *cluster* dalam *k-prototype* dapat dilakukan menggunakan metode *elbow* sebagai berikut.



Gambar 8. *Elbow method* pada k-prototype

Berdasarkan Gambar 8, terlihat bahwa grafik mengalami penurunan secara drastis pada saat jumlah *cluster* k=2. Sementara itu, penurunan grafik tidak lagi signifikan pada saat k>3. Untuk meminimalkan subjektivitas, kembali dilakukan simulasi *k-prototype* menggunakan 2 hingga 5 *cluster* untuk selanjutnya dibandingkan pada jumlah *cluster* berapa *silhouette index* memiliki nilai tertinggi. Berikut *silhouette index* yang dihasilkan dari simulasi jumlah *cluster* pada *k-prototype*.

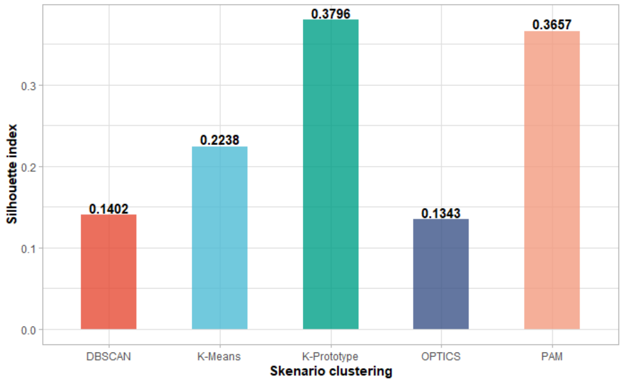
TABEL VI
SILHOUETTE INDEX HASIL SIMULASI K-PROTOTYPE

Jumlah *cluster*	*Silhouette index*
2	**0,3796**
3	0,3520
4	0,2533
5	0,3391

Tabel VI menunjukkan bahwa *k-prototype* dengan jumlah *cluster* 2 menghasilkan nilai *silhouette index* tertinggi, yaitu sebesar 0,3796. Maka, k=2 dipilih sebagai jumlah *cluster* optimum. Hasil menunjukkan bahwa *cluster* 1 memuat 175 observasi, sedangkan *cluster* 2 memuat 225 observasi.

G. Perbandingan Model

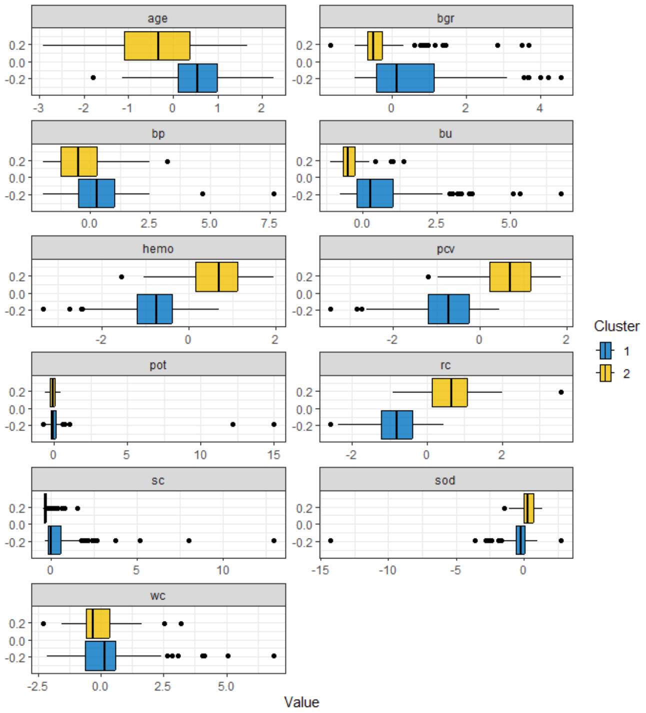
Berdasarkan kelima skenario *clustering* yang telah dilakukan, akan dibandingkan nilai *silhouette index* dari hasil simulasi terbaik pada masing-masing skenario.



Gambar 9. Perbandingan *silhouette index* dari kelima skenario *clustering*

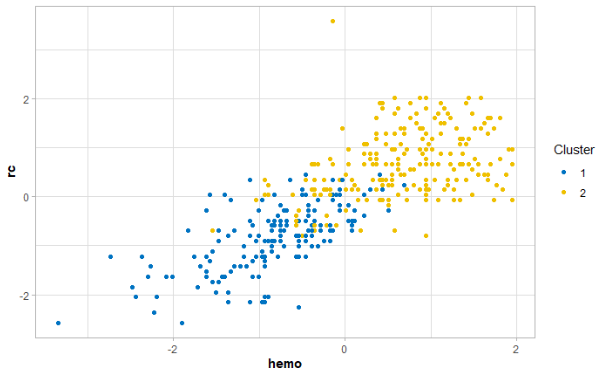
Berdasarkan Gambar 9, diketahui bahwa *k-prototype* menghasilkan nilai *silhouette index* tertinggi dibandingkan empat model lainnya. Maka, *k-prototype* menjadi metode *clustering* yang paling optimal untuk klasterisasi pasien terdiagnosis penyakit gagal ginjal.

Berikut beberapa visualisasi hasil *clustering* menggunakan algoritma *k-prototype*.

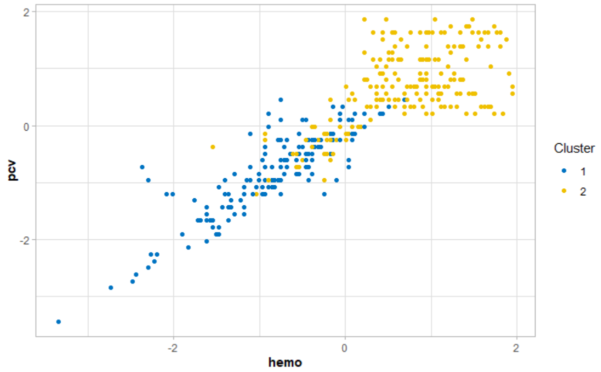


Gambar 10. *Profiling cluster* berdasarkan atribut numerik

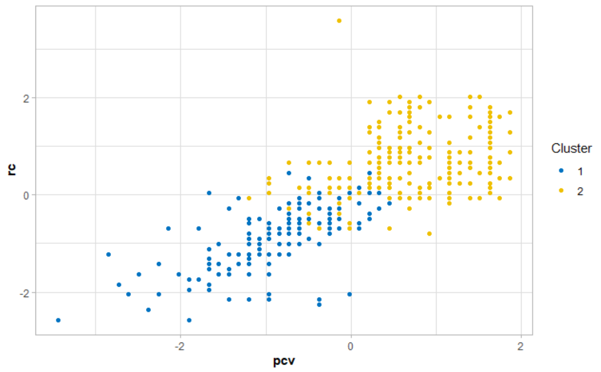
Gambar 10 menunjukkan bahwa *cluster* 1 memiliki rata-rata usia (*age*), kadar gula dalam darah (*bgr*), tekanan darah (*bp*), ureum (*bu*), kreatinin serum (*sc*), dan jumlah sel darah putih (*wc*) yang lebih tinggi dibandingkan *cluster* 2. Sementara untuk atribut hemoglobin (*hemo*), persentase volume sel-sel darah merah dalam satu liter darah (*pcv*), jumlah sel darah merah (*rc*), dan sodium (*sod*), *cluster* 1 memiliki nilai rata-rata yang lebih rendah dibandingkan *cluster* 2. Adapun pada atribut potasium (*pot*), tidak cukup terlihat perbedaan antara *cluster* 1 dan *cluster* 2.



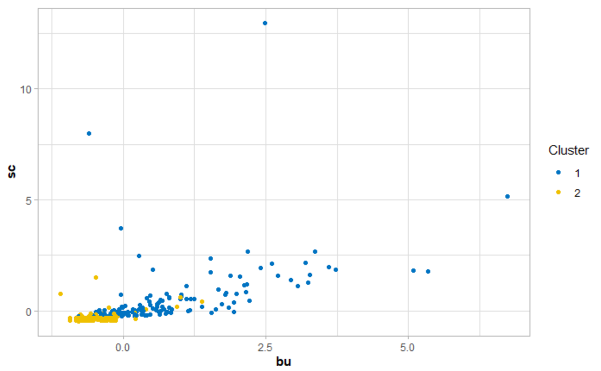
Gambar 11. Sebaran *cluster* berdasarkan atribut *hemo* dan *rc*



Gambar 12. Sebaran *cluster* berdasarkan atribut *hemo* dan *pcv*



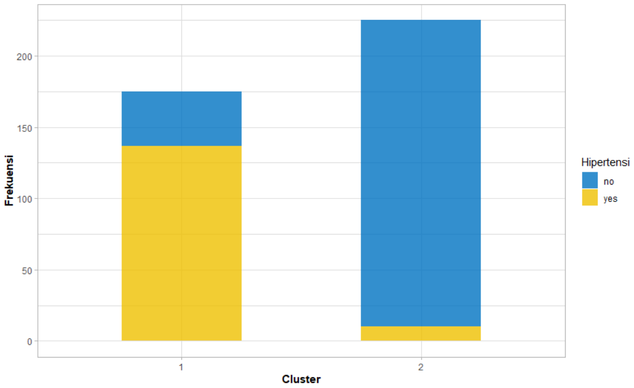
Gambar 13. Sebaran *cluster* berdasarkan atribut *pcv* dan *rc*



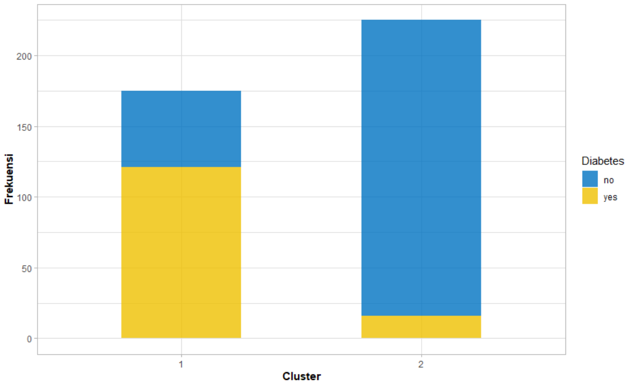
Gambar 14. Sebaran *cluster* berdasarkan atribut *bu* dan *sc*

Berdasarkan Gambar 11, terlihat bahwa *cluster* 1 memiliki hemoglobin (*hemo*) dan jumlah sel darah merah (*rc*) yang lebih rendah dibandingkan *cluster* 2. Hal serupa juga terlihat pada Gambar 12 dimana observasi dengan *hemo* dan persentase volume sel-sel darah merah dalam satu liter darah (*pcv*) yang rendah dikelompokkan dalam *cluster* 1. *Scatter plot* antara *pcv* dan *rc* pada Gambar 13 menunjukkan bahwa pasien dengan *pcv* dan *rc* rendah cenderung masuk dalam *cluster* 1. Sementara itu, berdasarkan Gambar 14, pasien dengan kadar ureum (*bu*) dan kreatinin serum (*sc*) tinggi masuk dalam *cluster* 1. Berdasarkan beberapa temuan tersebut, dapat diidentifikasi bahwa *cluster* 1 merupakan pasien yang didiagnosis menderita penyakit gagal ginjal kronis. Hal ini karena pasien gagal ginjal kronis memiliki kadar eritrosit (sel darah merah) yang rendah atau mengalami anemia. Ketika sel darah merah (*rc*) menurun, maka hemoglobin (*hemo*) yang terkandung di dalamnya juga akan menurun, persentase volume sel-sel darah merah dalam satu liter darah (*pcv*) juga ikut menurun [18]. Sementara itu, kreatinin serum (*sc*) dan ureum (*bu*) kadarnya akan meningkat seiring dengan menurunnya fungsi ginjal [19].

Adapun visualisasi hasil *clustering* *k-prototype* berdasarkan atrubut kategorik *htn* dan *dm* dapat dilihat pada gambar berikut.



Gambar 15. Sebaran *cluster* berdasarkan atribut *htn*



Gambar 16. Sebaran *cluster* berdasarkan atribut *dm*

Berdasarkan Gambar 15 dan 16, terlihat bahwa pasien pada *cluster* 1 didominasi oleh pasien dengan hipertensi (*htn*) dan *diabetes mellitus* (*dm*). Hal ini juga mendukung bahwa *cluster* 1 merupakan pasien terdiagnosis penyakit gagal ginjal kronis, dimana *diabetes mellitus* dan hipertensi merupakan faktor risiko terjadinya gangguan fungsi ginjal. Tekanan darah dan kadar gula dalam darah yang tinggi dapat mengganggu kemampuan ginjal dalam menyaring darah [20].

TABEL VII
CROSS-TAB HASIL K-PROTOTYPE

Cluster	*CKD*	*NOT CKD*
1	175	0
2	75	150

Apabila mengaitkan hasil *clustering* dengan label yang tersedia pada *dataset*, diperoleh bahwa 175 pasien yang masuk dalam *cluster* 1 seluruhnya merupakan penderita gagal ginjal. Sementara pada *cluster* 2, sebanyak 150 pasien bukan penderita gagal ginjal, sedangkan 75 pasien sisanya menderita gagal ginjal. Hal tersebut menunjukkan bahwa hasil *clustering* menggunakan *k-prototype* memberikan nilai akurasi sebesar 81,25 persen.

V. KESIMPULAN

Penelitian ini menerapkan lima skenario untuk mengklasterkan *dataset Chronic Kidney Disease* (CKD) dengan atribut bertipe data campuran numerik dan kategorik. Lima skenario yang dimaksud meliputi *k-means* dengan konversi variabel kategorik menjadi numerik, PAM dengan *Gower distance*, DBSCAN dengan *Gower distance*, OPTICS dengan *Gower distance*, dan *k-prototype*. Berdasarkan simulasi yang telah dilakukan pada masing-masing skenario *clustering*, diperoleh bahwa *k-prototype* dengan k=2 mampu menghasilkan *silhouette index* tertinggi dibandingkan empat metode lainnya, yaitu sebesar 0,3796. Hasil ini dapat dijadikan pertimbangan untuk membantu tenaga kesehatan dalam memberikan diagnosis penyakit gagal ginjal dengan lebih cepat. Dengan menerapkan salah satu metode *unsupervised learning* ini, memungkinkan pasien untuk menerima diagnosis dini, sehingga pasien yang terdiagnosis penyakit gagal ginjal dapat segera memperoleh penanganan.

DAFTAR PUSTAKA

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, 3rd Edition*. 2011.

[2] P. A. Popoola, J. R. Tapamo, and A. G. Assounga, “Cluster Analysis of Mixed and Missing Chronic Kidney Disease Data in KwaZulu-Natal Province, South Africa,” *IEEE Access*, vol. 9, pp. 52125–52143, 2021, doi: 10.1109/ACCESS.2021.3069684.

[3] R. Wijayati and D. R. S. Saputro, “Clustering Data Campuran Numerik dan Kategorik Menggunakan Algoritme K-Prototype,” *PRISMA: Prosiding Seminar Nasional Matematika* 6, pp. 702–706, 2023, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/prisma/

[4] A. Ahmad and L. Dey, “A k-mean clustering algorithm for mixed numeric and categorical data,” *Data Knowl Eng*, vol. 63, no. 2, pp. 503–527, Nov. 2007, doi: 10.1016/j.datak.2007.03.016.

[5] Z. Huang, “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values,” 1998.

[6] M. Ganmanah and A. Kudus, “Penerapan Algoritme K-Prototypes untuk Pengelompokkan Desa-Desa di Provinsi Jawa Barat Berdasarkan Indikator Indeks Desa Membangun Tahun 2020,” *Prosiding Statistika*, vol. 7, no. 2, pp. 543–548, 2021, doi: 10.29313/.v0i0.28974.

[7] J. C. Gower, “A General Coefficient of Similarity and Some of Its Properties,” 1971.

[8] S. Revathy, B. Bharathi, P. Jeyanthi, and M. Ramesh, “Chronic kidney disease prediction using machine learning models,” *Int J Eng Adv Technol*, vol. 9, no. 1, pp. 6364–6367, Oct. 2019, doi: 10.35940/ijeat.A2213.109119.

[9] L. Antony *et al.*, “A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction,” *IEEE Access*, vol. 9, pp. 126481–126501, 2021, doi: 10.1109/ACCESS.2021.3109168.

[10] S. Gopika and M. Vanitha, “Machine Learning Approach of Chronic Kidney Disease Prediction Using Clustering Technique,” *Int J Innov Res Sci Eng Technol*, vol. 6, no. 7, pp. 14488–14496, 2017, doi: 10.15680/IJIRSET.2017.0607267.

[11] A. F. Sallaby and A. Azlan, "Analysis of Missing Value Imputation Application with K-Nearest Neighbor (K-NN) Algorithm in Dataset," *The IJICS (International Journal of Informatics and Computer Science)*, vol. 5, no. 2, p. 141, Aug. 2021, doi: 10.30865/ijics.v5i2.3185.

[12] J. Gandhi, R. Goyal, J. Guha, K. Pithawala, and S. Joshi, "Comparative Study on Hierarchical and Density based Methods of Clustering using Data Analysis," in *International Conference on IoT based Control Networks and Intelligent Systems (ICICNIS 2020)* , 2020, pp. 62–68. [Online]. Available: https://ssrn.com/abstract=3768295

[13] S. Wang, J. G. Yabes, and C.-C. H. Chang, "Hybrid Density- and Partition-Based Clustering Algorithm for Data With Mixed-Type Variables," *Journal of Data Science*, pp. 15–36, 2021, doi: 10.6339/21-jds996.

[14] M. N. T. Elbatta, "An improvement for DBSCAN algorithm for best result in varied densities." Islamic University of Gaza, Gaza, 2012.

[15] M. Refaldy, S. Annas, and Z. Rais, "K-Prototype Algorithm in Grouping Regency/City in South Sulawesi Province Based on 2020 People's Welfare," *ARRUS Journal of Mathematics and Applied Science*, vol. 3, no. 1, pp. 11–19, May 2023, doi: 10.35877/mathscience1763.

[16] M. R. Irianto, A. Maududie, and F. N. Arifin, "Implementation of K-Means Clustering Method for Trend Analysis of Thesis Topics (Case Study: Faculty of Computer Science, University of Jember)," *BERKALA SAINSTEK*, vol. 10, no. 4, p. 210, Dec. 2022, doi: 10.19184/bst.v10i4.29524.

[17] L. Rubini, P. Soundarapandian, and P. Eswaran, "Chronic_Kidney_Disease." UCI Machine Learning Repository, 2015.

[18] F. Muchtar, "Gambaran Hematologi pada Pasien Gagal Ginjal Kronik yang Menjalani Hemodialisa," 2013.

[19] D. G. A. Suryawan, I. A. M. S. Arjani, and I. G. Sudarmanto, "Gambaran Kadar Ureum dan Kreatinin Serum pada Pasien Gagal Ginjal Kronis yang Menjalani Terapi Hemodialisis di RSUD Sanjiwani Gianyar," *Meditory*, vol. 4, no. 2, pp. 145–153, 2016.

[20] E. Sulistiowati and S. Idaiani, "Faktor Risiko Penyakit Ginjal Kronik Berdasarkan Analisis Cross-sectional Data Awal Studi Kohort Penyakit Tidak Menular Penduduk Usia 25-65 Tahun di Kelurahan Kebon Kalapa, Kota Bogor," *Buletin Penelitian Kesehatan*, vol. 43, no. 3, pp. 163–172, 2015.