Jurnal Nasional Informatika dan Teknologi Jaringan
E-ISSN : 2540-7600
P-ISSN : 2540-7597

# PERBANDINGAN RAPID CENTROID ESTIMATION (RCE) — K NEAREST NEIGHBOR (K-NN) DENGAN K MEANS — K NEAREST NEIGHBOR (K-NN)

Khairul Umam Syaliman bin Lukman[1], M. Zulfahmi[2], Aldi Abdillah Nababan[3]
[1,2,3] *Program Magister (S2) Teknik Informatika, Universitas Sumatera Utara*
*Jl. Dr. Mansyur, Medan (20155) , Indonesia*
[1]*khairul.q14@gmail.com,* [2]*fhm.nst@gmail.com,* [3]*adlyan619@gmail.com*

*Abstrak*— **Teknik Clustering terbukti dapat meningkatkan akurasi dalam melakukan klasifikasi, terutama pada algoritma K-Nearest Neighbor (K-NN). Setiap data dari setiap kelas akan membentuk K cluster yang kemudian nilai centroid akhir dari setiap cluster pada setiap kelas data tersebut akan dijadikan data acuan untuk melakukan proses klasifikasi menggunakan algoritma K-NN. Namun kendala dari banyaknya teknik clustering adalah biaya komputasi yang mahal, Rapid Centroid Estimation (RCE) dan K-Means termasuk kedalam teknik clustering dengan biaya komputasi yang murah. Untuk melihat manakah dari kedua algoritma ini (RCE dan K-Means) yang lebih baik memberikan peningkatan akurasi pada algoritma K-NN maka, pada penelitian ini akan mencoba untuk membandingkan kedua algoritma tersebut. Hasil dari penelitian ini adalah gabungan RCE—K-NN memberikan hasil akurasi yang lebih baik dari K-Means—K-NN pada data set iris dan wine. Namun dalam perubahan nilai akurasi RCE—K-NN lebih stabil hanya pada data set iris. Sedangkan pada data set wine, K-Means—K-NN terlihat mendapati perubahan akurasi yang lebih stabil dibandingkan RCE—K-NN.**

*Keywords*— **Akurasi, Clustering, K-Means, K-Nearest Neighbor (K-NN), Rapid Entimation Centroid (RCE).**

## I. PENDAHULUAN

Clustering digunakan terhadap data yang bersifat unsupervised, dimana data-data tidak terawasi atau tidak diketahui kelasnya. Clustering adalah cara untuk mengelompokkan data yang didasari pada kemiripan antar data, sehingga data dengan kemiripan paling dekat berada dalam satu cluster sedangkan data yang berbeda dalam kelompok lainya [18][13]. Beberapa algoritma clustering adalah K-Means, K-Medoids, DBSCAN, Fuzzy C-Means, dan lain sebagainya [11].

Klasifikasi digunakan untuk kelompok data yang bersifat supervised, dimana data-data terawasi atau sudah diketahui kelasnya. Tujuan klasifikasi adalah untuk mendekrispikan suatu data atau objek baru kedalam kelas tertentu berdasarkan kemiripan karakteristik datanya. Algoritma fungsi klasifikasi terdiri dari Statistical-Based Algorithms, Distance-Based Algorithms, Decision Tree-Based Algorithms, Neural Network-Based Algorithms, dan Rule-Based Algorithms [19].

Distance-Based Algorithms adalah algoritma yang menentukan kemiripan data atau objek berdasarkan pada kedekatan jarak antar data ke suatu kelas atau label atau kelompok data lainnya. Kedekatan jarak itu didapati dari perhitungan nilai karakterisik setiap data, Salah satu algoritma distance-based adalah K-Nearest Neighbor (K-NN).

K-NN diperkenalkan pertama kali pada awal tahun 1950-an [13]. Penentuan kelompok suatu data dalam K-NN ditentukan berdasarkan kelas mayoritas dari K tetangga terdekat. Beberapa hal yang menjadi perhatian lebih dalam algoritma ini adalah pemilihan nilai K yang optimum. Selain itu, pemiliha model jarak juga menjadi salah satu aspek yang harus dipertimbangkan. Meskipun demikian dengan keunikannya, K-NN tetap termasuk salah satu dalam top 10 algorithm [31].

K-NN juga termasuk dalam algoritma lazy learner, dimana algoritma jenis lazy learner ini hanya sedikit melakukan pembelajaran atau tidak sama sekali. K-NN akan meyimpan semua data latih untuk dijadikan pedoman dalam melakukan klasifikasi bagi data baru. Dalam K-NN setiap karakteristik data memiliki bobot yang sama, karena itu pemilihan nilai K untuk dijadikan tetangga terdekat sangat berpengaruh terhadap hasil data baru yang diklasifikasikan.

Akurasi dari algoritma K-NN konvensional masih rendah jika dibandingkan dengan Support Vectore Machine (SVM) [1]. Hal ini tentunya menjadi peluang untuk meningkatkan nilai akurasi tersebut.

Salah satu cara untuk meningkatkan nilai akurasi dan kompleksitas waktu dengan menggunakan teknik clustering [10]. Metode clustering K-Means contohnya, yang digabungkan dengan K-NN [26].

Namun kendala utama dari banyak teknik clustering adalah komputasional yang mahal yang terbatas pada volume dan dimensi data [20]. Ada banyak cara yang telah ditawarkan untuk menetukan titik centroid awal, antara lain dengan cara Biogeography Based Optimization[30], Algoritma Genetika [5], Rapid Centroid Estimation (RCE) [21], dan lain sebagainya.

RCE menyederhanakan aturan Partical Swarm Clustering (PSC)[21], dan sangat mengurangi kompleksitas komputasi dengan meningkatkan efisiensi lintasan partikel. Varian RCE (RCE++, Swarm Clustering) jauh lebih cepat dari PSC dan mPSC[24].

Penelitian ini akan mencoba menganalisa perbandingan dari gabungan algoritma RCE—K-NN dengan K-Means—K-NN. RCE dan K-Means masing-masing akan menentukan nilai centroid untuk setiap kelas data latih [10], kemudian nilai centroid tersebut akan digunakan oleh K-NN untuk klasifikasi terhadap data uji yang baru.

Tujuan penelitian ini adalah untuk melihat pengaruh kinerja dari algoritma clustering (K-Means dan RCE) terhadap proses klasifikasi menggunakan K-Nearest Neighbor (K-NN). Dimana proses klasifikasi pada K-NN didasari pada pusat cluster dari setiap kelas yang didapati melalui proses clustering terlebih dahulu.

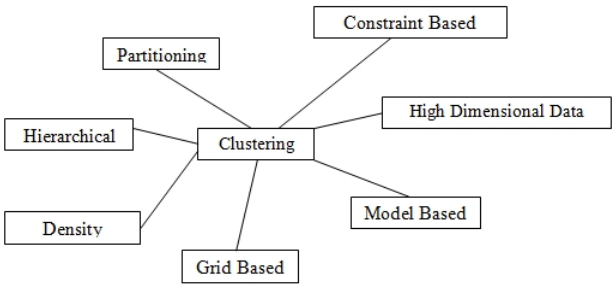
## II. TINJAUAN PUSTAKA

### A. *Clustering*

*Clustering* adalah proses pembentukan kelompok dari satu set objek fisik atau abstrak ke dalam label/kelas yang serupa. Sebuah kelompok yang terdiri dari sekumpulan data-data yang mirip satu sama lain dan berbeda dengan data-data di kelompok lainnya. Sekumpulan data bisa diperlakukan secara kolektif sebagai satu kelompok dan dapat dianggap sebagai bentuk kompresi data [13]. Tujuan clustering adalah untuk memisahkan data yang bersifat *unsupervised*

M. Bramer (2007) menjabarkan bahwa *clustering* adalah pengelompokan bersama data-data yang mirip dan mengelompokkan data yang berbeda dalam kelompok lainya. *Clustering* membagi data kedalam kelompok-kelompok atau cluster tertentu berdasarkan suatu kemiripan atribut-atribut antar data tersebut [19]. Sedangkan menurut F. Gorunescu (2011), *clustering* berarti menemukan kelompok (cluster) dari objek-objek, berdasarkan kesamaan (kemiripan), sehingga masing-masing kelompok memiliki kesamaan yang lebih dekat dibandingkan dengan objek dari kelompok yang lainnya.

Adapun jenis-jenis metode *clustering* secar umum dapat dilihat pada gambar dibawah ini (Gambar 1) :




Gbr 1 Jenis-Jenis Metode *Clustering*

Pengelompokan data atau *clustering* memiliki dua tujuan. Pertama adalah pengelompokan yang bertujuan untuk pemahaman, dengan cara pembentukan kelompok yang harus mampu menangkap struktur alamiah data, proses dalam pengelompokan ini hanya merupakn proses awal yang kemudian dilanjutkan dengan peringkasan, pelabelan sehingga akhirnya dapat dimanfaatkan sebagai data latih dalam klasifikasi atau sebagainya. Sedangkan pengelompokan yang kedua bertujuan untuk penggunaan, pengelompokan ini memiliki tujuan utama untuk mencari prototipe atau perwakilan yang represntatif terhadap data suatu kelompok, sehingga dapat memberikan sebuah abstraksi dari setiap objek data dalam kelompok yang diwakilkan.

### B. *K-Means*

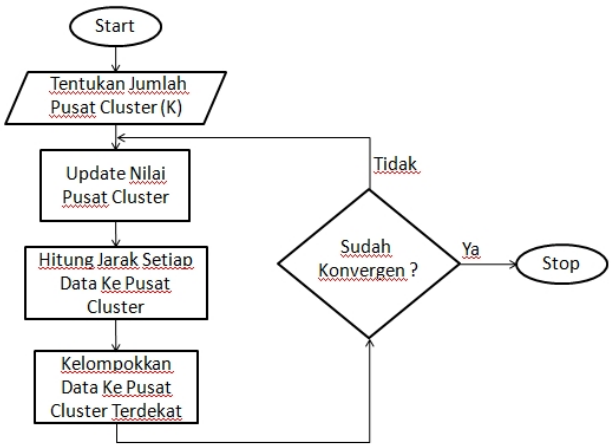
*K-Means* adalah metode yang mempartisi data ke suatu kelompok, dimana data yang berkarakterisitik sama akan terdapat pada satu kelompok, sedangkan data dengan karakteristik berbeda berada dalam kelompok lainnya [12]. *K-Means* termasuk salah satu pengelompokan data nonherarki, dimana jumlah kelompok sudah ditentukan diawal.

Pengelompokan *K-Means* dilakukan pada data sebanyak N yang kemudian akan dikelompokkan sebanyak K. Pengelompokan ini untuk meminimalkan fungsi objektif yang diset dalam proses pengelompokan, *K-Means* berusaha meminimalkan variasi dalam suatu kelompok data, dan memaksimalkan variasi antar kelompok.

Secara umum *K-Means* akan menjalankan langkah-langkah berikut dalam pengelompokkan data :

a. Tentukan Jumlah Kelompok (K).
b. Tentukan Nilai Centroid atau Update Nilai Centroid
   Dalam menentukan nilai centroid awal pada umumnya dilakukan secara acak. Sedangkan nilai centroid pada iterasi selanjutnya digunakan rumus sebagai berikut :
   $$\bar{v}_{\mathrm{k}} = \frac{1}{N_k}\sum_{i=0}^{N_k} X_i \quad (1)$$
   Dimana :
   $\bar{v}_k$ merupakan titik centroid pada *cluster*-k untuk variable-j.
   $N_k$ adalah jumlah data yang menjadi *cluster*-k.
   $X_i$ adalah nilai data ke-i untuk variable-j.
c. Menghitung kemiripan seluruh data terhadap setiap titik centroid.
d. Pengelompokan data yang memiliki kemiripan terdekat.
e. Kembali ketahap b jika data belum konvergen, atau hentikan jika sudah konvergen.

*Flowchart* dari *K-Means clustering* dapat dilihat dari gambar dibawah ini (Gambar 2) :




Gbr 2 *Flowchart K-Means Clustering*

### C. *Particle Swarm Clustering (PSC)*

Ide utama dari algoritma particle swarm clustering (PSC) adalah untuk menciptakan satu set partikel dari data masukan dan memindahkan partikel-partikel ini sehingga menjadikan prototype yang mewakili rangkaian data masukan[27].

PSC dapat dipandang sebagai modifikasi khusus Particle Swarm Optimization (PSO) yang dirancang khusus untuk tugas pengelompokan[27]. Hal ini berbeda dengan implementasi PSO secara umum, dimana masing-masing partikel mewakili solusi kandidat. Pada PSC, masing-masing partikel hanya mewakili sebuah centroid kelompok data[20].

Perbedaan struktural utama antara PSO dan PSC adalah [3] :

- Setiap partikel dalam PSO merupakan solusi potensial untuk masalah ini. Pada PSC, setiap partikel merupakan solusi dalam pengelompokkan data.
- PSO secara eksplisit digunakan untuk mengevaluasi kualitas solusi yang dihasilkan oleh algoritma. PSC digunakan untuk mengukuran ketidaksamaan antara partikel dan objek.
- Dalam algoritma PSC ditambahkan istilah self-organizing dalam persamaan yang memperbarui kecepatan partikel. Istilah ini bertanggung jawab untuk memindahkan partikel ke arah objek input, dengan cara mengatur sendiri.
- Dalam algoritma PSO, semua partikel di swarm diperbarui pada setiap iterasi algoritma. Di PSC, partikel yang akan diperbarui didefinisikan oleh input data (yaitu hanya pemenangnya - yang paling dekat dengan input yang dipertimbangkan - diperbarui sesuai dengan persamaan kecepatan dan posisi).

Algoritma PSC menggunakan kecerdasan kolektif untuk memecahkan masalah dalam teknik clustering [4]. Adapun *pseudocode* dari PSC adalah sebagai berikut [20-24] :

**Algorithm** S = PSC (dataset, max_iteration, $v_{max}$, $n_c$, $\omega$)
Inisialisasikan $n_c$ partikel, random x, dan inisialisasikan v ke nol
Hitung jarak dari p, g, dari setiap partikel pada setiap datum
**While** t < max_iteration
  **For** setiap datum y
    Update matrik jarak dan tentukan partikel terdekat :
    D(t)=d(y_j,x_i ):∀i,j
    I=min (d(y_j,x_i ) ): i □{1,2,3,…,n_c}
    Update personal terbaik dan global terbaik dari partikel

$$p_I(t+1) = \begin{cases} x_I(t) & ji \quad d(y_j, x_I(t)) < d(y_j, p_I(t)) \\ p_I(t) & ji \quad ti \end{cases}$$

$$g_j(t+1) = \begin{cases} x_I(t) & ji \quad d(y_j, x_I(t)) < d(y_j, g_j(t)) \\ g_j(t) & ji \quad ti \end{cases}$$

    Update kecepatan dan posisi

$$v_I(t+1) = \omega(t)v_I(t+1) + \varphi_1 \otimes X_I(t) + \varphi_2 \otimes Y_I(t) + \varphi_2 \otimes Z_I(t)$$

$$v_I \in [-v_m \ , v_m \ ]$$

$$v_I(t+1) = x_I(t) + v_I(t+1)$$

  **End For**
  Tentukan partikel pemenang

$$x_{m \ \_w}(t) = x(t) \in \min\left(d\left(p_i(t) - x_i(t)\right)\right) : \forall i$$

  **For** setiap partikel partikel x
    **If** ($x_i$ tidak mendekati data point manapun)
      Pindahkan $x_i$ ke partikel pemenang

$$v_I(t+1) = \omega(t)v_I(t+1) + \varphi_1 \otimes X_I(t) + \varphi_2 \otimes Y_I(t) + \varphi_2 \otimes Z_I(t)$$

$$v_I \in [-v_m \ , v_m \ ]$$

$$v_I(t+1) = x_I(t) + v_I(t+1)$$

    **End If**
  **End For**

$$w(t+1) = 0.95w(t)$$

$$t = t + 1$$

**End While**

*Pseudocode* 1 Algoritma PSC

### D. *Rapid Centroid Estimation (RCE)*

Penentuan awal *centroid* secara *random* merupakan masalah [30]. Karena awal *centroid* yang ditentukan secara *random* memiliki sensitifitas yang buruk [6] dan tidak menjamin hasil *clustering* yang baik [9].

Yuwono, *et al* (2012) mengusulkan suatu metode dalam penelitiannya yang disebut *Rapid Centroid Estimation* (RCE). RCE secara umum mirip dengan konsep pengambilan keputusan [21]. RCE berbasis pada algoritma *Particle Swarm Clustering* (PSC), namun dikonfigurasikan ulang untuk mengurangi kerumitan komputasional [24].

Algoritma ini mampu mencapai kinerja dengan stabilitas yang lebih tinggi dan kecepatan optimasi yang lebih cepat dari pada PSC [22]. Adapun skema kerja RCE adalah sebagai berikut [20]:

a. Setiap iterasi, lakukan *update* posisi untuk setiap partikel.
b. Matriks jarak dan posisi terbaik diperbarui setelah semua posisi partikel diperbarui.

c. Perhitungan minimum global didefinisikan untuk menyimpan kombinasi posisi terbaik dan menghentikan perulangan pengoptimalan.

Untuk menerapkan skema diatas, maka formulasi untuk melakukan perhitungannya adalah sebagai berikut [20]:

$$X_i(t) = p_i(t) - x_i(t) \quad (2)$$

$$Y_i(t) = \frac{\sum_{\forall j \in x_i(t)} \hat{\varphi}_{i,j} \oplus \left(g_j(t) - x_i(t)\right)}{N_j} \quad (3)$$

$$Z_i(t) = \frac{\sum_{\forall j \in x_i(t)} \hat{\varphi}_{i,j} \oplus \left(y_j(t) - x_i(t)\right)}{N_j} \quad (4)$$

$$M(t) = [x_1^b \quad x_2^b \quad \dots \quad x_n^b \quad] \quad (5)$$

Dimana :

- $\hat{\varphi}_{i,j}$ adalah tingkat subjektivitas terhadap pola masukan, dimodelkan dengan menggunakan bilangan acak seragam $0 \leq \hat{\varphi}_{i,j} \leq 1$.
- $x_i(t)$, $p_i(t)$ menunjukkan posisi dan posisi terbaik partikel i yang terhubung pada pola input j.
- $g_j(t)$ mewakili posisi partikel yang paling dekat dengan pola input j.
- $M(t)$ merupakan kombinasi posisi terbaik yang telah mencapai minimum global sesuai dengan fungsi *fitness* yang diberikan.

Sedangkan, perubahan posisi $x$ didefenisikan sebagai berikut :

$$\Delta x_i(t+1) = w(t) \Delta x_i(t) + \varphi_1 \oplus X_i(t) + Y_i(t) + Z_i(t) \quad (6)$$

$$x_i(t+1) = x_i(t) + \Delta x_i(t+1) \quad (7)$$

Dimana $w(t)$ adalah bobot inersia, dengan nilai awal 0.8. Adapun *pseudocode* algoritma RCE adalah sebagai berikut[22] :

**Algorithm** S = RCE (dataset, max_iterasi, s_max, $\varepsilon$, $n_c$)
Inisialisasikan $n_c$ partikel, random x,
Hitung jarak p, g untuk setiap partikel kesetiap datum
**While** t < max_iterasi && $s_c$ < s_max
Update matrik jarak
$D(t) = d(y_j, x_i) : \forall i, j$
Temukan titik data terdekat dari setiap partikel
$[D^m(t) \quad l_j^m] = \min(D, i)$
Temukan setiap partikel terdekat untuk setiap data
$[D^m(t) \quad l_i^m] = \min(D, y)$
Update $p_i(t)$, $h_j(t)$, M(t)

$$p_i(t+1) : \forall i = \begin{cases} y_{l_i^m}(t), ji & D_i^m(t) < D_i^m(t-1) \\ p_i(t) & ji \quad ti \end{cases}$$

$$g_j(t+1) : \forall j = \begin{cases} y_{l_j^m}(t) ji & D_j^m(t) < D_j^m(t-1) \\ p_j(t) & ji \quad ti \end{cases}$$

$$M(t+1) : \forall i = \begin{cases} x_i(t) : \forall i \, ji & f(x_i(t) : \forall i) < f(M(t)) \\ M(t) & ji \quad ti \end{cases}$$

Increment $s_c$ jika gradient lebih tinggi dari $-\varepsilon$

$$s_c = \begin{cases} s_c + 1 \, ji & f(M(t+1)) - f\left(M(t)\right) > -\varepsilon \\ 0 & ji \quad ti \end{cases}$$

Tentukan partikel pemenang (Partikel yang paling dekat terhadap pola inputan dengan menggunakan rumus Euclidean distance)

$$x_{m\_w}(t) = x(t) \in \min\left(d\left(p_i(t) - x_i(t)\right)\right) : \forall i$$

**For** untuk setiap partikel $x$
Tentukan elemen dari setiap anggota partikel i (*centroid*)
$y_i^{c_i} = \forall y \in x_i(t)$
$N_i = s_i \quad (y_i^{c_i})$
Update posisi dengan persamaan (7) jika $N_i$ lebih besar dari nol, jika tidak langsung menuju kordinat partikel pemenang.

$$x_i(t+1) = \begin{cases} x_i + \Delta x_i(t+1) & ji \quad N_i > 0 \\ x_i(t) + \varphi_5 \otimes \left(x_{m\_t_w}(t) - x_i(t)\right) ji & ti \end{cases}$$

**End For**
$w(t+1) = 0.95 w(t)$
$t = t + 1$
**End While**

*Pseudecode* 2 Algoritma RCE

## *E. Klasifikasi*

Klasifikasi pertama kali diterapkan pada bidang tanaman yang mengklasifikasikan suatu spesies tertentu. Klasifikasi adalah suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu berdasarkan karakteristik yang dimiliki [31].

Klasifikasi biasanya diawali dengan melatih pengklasifikasian pada satu set data berlabel (data latih), kemudian menggunakannya untuk memberi label pada data yang tidak berlabel [32]. Han et al [13] menjabarkan klasifikasi merupakan proses analisis data untuk menemukan model yang mampu memprediksi kelas dari objek diketahui. Model klasifikasi dibangun berdasarkan analisis data training atau objek data yang sudah memiliki kelas terlebih dahulu.

Sebuah sistem yang melakukan proses klasifikasi diharapkan dapat menentukan semua target data input dengan benar, namun tidak dapat dimungkiri bahwa kinerja suatu sistem tidak bisa seratus persen benar, sehingga sebuah sistem klasifikasi juga harus diukur kinerjanya, contohnya dengan menggunakan matriks konfusi (*confusion matrix*) untuk mengukur kinerja klasifikasi.

Dengan mengetahui jumlah data yang diklasifikasi secara benar dan salah, kita dapat mengetahui tingkat akurasi serta laju error dari hasil prediksi. Untuk menghitung akurasi dapat menggunakan rumus di bawah ini [13] :

$$\text{Tingkat Akurasi} = \frac{\text{T} + \text{T}}{\text{P+N}} \quad (8)$$

Untuk mengukur laju *error* digunakan formula :

$$L_{e_i} = \frac{F + F}{P+N} \quad (9)$$

Formula untuk mengukur sensitivitas :

$$S_i = \frac{T}{P} \quad (10)$$

Formula untuk mengukur specificity (True Negative) :

$$S_\xi = \frac{T}{N} \quad (11)$$

Formula untuk mengukur precision :

$$P = \frac{T}{T + N} \quad (12)$$

Sedangkan Formula untuk mengukur F-Score :

$$F - S = \frac{2 \times p \quad \times r}{p \quad +r} \qquad (13)$$

Dimana :

TP : True Positive, jumlah data positive yang terprediksi dengan benar

TP : True Negative, jumlah data negative yang terprediksi dengan benar

FP : False Positive, jumlah data positive yang terprediksi salah

FN : False Negative, jumlah data negative yang terprediksi salah

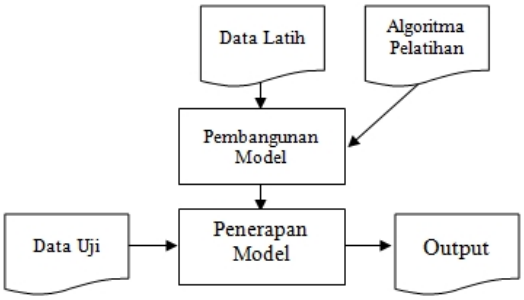
P : Jumlah Seluruh Data Positive

N : Jumlah Seluruh Data Negative

Terdapat dua pekerjaan utama dalam klasifikasi, yaitu pembentukan model sebagai prototipe dan penggunaan model tersebut dalam proses klasifikasi suata objek data.

Semua algoritma klasifikasi mencoba membangun model dengan tingkat akurasi yang tinggi (laju *error* yang rendah). Secara umum, model yang dibangun mampu memprediksi data latih dengan benar, akan tetapi ketika model tersebut dilakukan proses klasifikasi dengan data uji, barulah kinerja dari model sebuah algoritma klasifikasi ditentukan.

Model dalam proses klasifikasi memiliki arti yang sama dengan kotak hitam, dimana model tersebut menerima masukan yang kemudian model tersebut harus mampu melakukan proses klasifikasi dengan tepat terhadap masukkan yang diberikan. Gambaran kerangka kerja klasifikasi dapat dilihat pada gambar dibawah ini (gambar 3) :




Gbr 3 Proses Kerja Klasifikasi

Kerangka kerja klasifikasi meliputi dua langkah proses, yaitu induksi yang merupakan langkah untuk membangun model klasifikasi dari data latih yang diberikan, dan deduksi merupakan proses untuk menerapkan model tersebut pada data uji sehingga kelas yang sesungguhnya dari data uji dapat diketahui atau biasa disebut proses prediksi.

Ada banyak algoritma pelatihan yang sudah dikembangkan oleh para peneliti, namun berdasarkan cara pelatihannya algoritma untuk klasifikasi dapat dibedakan menjadi dua macam, yaitu *eager learner* dan *lazy learner*. *Eager learner* didesain untuk melakukan pembacaan/ pelatihan/ pembelajaran pada data latih untuk mampu memetakan dengan tepat setiap vektor masukan ke target kelas keluarannya, sehingga di akhir proses pelatihannya model diharapkan sudah mampu memetakan seluruh vektor data uji ke label kelas dengan tepat. Selanjutnya, setelah proses pelatihan selesai, model disimpan sebagai pedoman dalam melakukan proses pemetaan. Proses prediksi dilakukan dengan model yang tersimpan, proses ini tidak melibatkan data latih. Cara ini membuat proses prediksi dapat dilakukan dengan cepat, tetapi harus dibayar dengan proses pelatihan yang lama. Algoritma yang masuk dalam kategori ini diantaranya *Artificial Neural Network* (ANN), *Support Vectore Mechine* (SVM), *Decision Tree*, Bayesian, dan lain sebagainya.

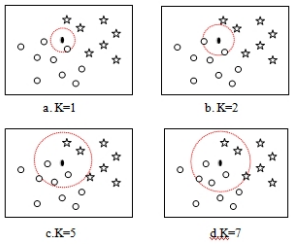
Sementara itu algoritma *lazy learner* merupakan algoritma yang sedikit melakukan pelatihan atau sama sekali tidak melakukan pelatihan. Algoritma jenis ini menyimpan sebagian atau seluruh data latih yang kemudian digunakan untuk proses prediksi, hal ini mengakibatkan proses prediksi menjadi lama. Kelebihan algoritma ini adalah proses pelatihan yang berjalan dengan cepat, algoritma klasifikasi yang termasuk kategori ini diantaranya adalah *K-Nearest Neighboar* (K-NN), *Fuzzy K-Nearest Neighboar* (FK-NN), Regresi Linear, dan sebagainya [17].

### *F. K-Nearest Neighbor (K-NN)*

Algoritma K-*Nearest Neighbor* (K-NN) adalah algoritma yang melakukan klasifikasi berdasarkan kemiripan suatu data dengan data yang lain. Analogi sederhana dari K-NN adalah "Jika suatu hewan berjalan seperti bebek, bersuara kwek-kwek seperti bebek, dan penampilannya seperti bebek, hewan itu mungkin bebek" [7].

Nilai K pada K-NN merupakan jumlah tetangga terdekat, jika K bernilai 1, maka kelas dari satu data latih yang terdekat akan menjadi kelas bagi data uji yang baru, jika K bernilai 2 akan diambil dua data latih yang terdekat menjadi kelas untuk data uji yang baru. Begitu juga jika nilai K adalah 3, 4, 5, dan seterusnya. Pada proses klasifikasi K-NN menggunakan *voting* terbanyak sebagai kelas prediksi dari data yang baru [29].

Untuk lebih jelasnya perhatikan gambar 3, ada dua kelas sebagai sampel yaitu lingkaran dan bintang, dan oval yang berwarna hitam adalah data baru yang akan diklasifikasikan oleh algoritma K-NN.




Gbr 4 K-NN dengan nilai: (a) K=1, (b) K=2, (c) K=5, (d) K=7

Jika K bernilai 1 maka kelas untuk data baru adalah kelas lingkaran (gambar 3 bagian a), jika K bernilai 2 maka kelas masih sama dengan K bernilai 1 yaitu lingkaran (gambar 3 bagian b), jika K bernilai 4 maka kelas untuk data baru juga

lingkaran (gambar 3 bagian c), dan gambar 3 bagian d memiliki hasil prediksi dengan kelas mayoritas lingkaran.

Salah satu masalah yang dihadapi K-NN adalah dalam pemilihan nilai K yang tepat. Pemilihan nilai K yang besar dapat mengakibatkan distorsi data yang besar pula. Hal ini disebabkan setiap tetangga terdekat memiliki nilai bobot yang sama terhadap setiap data uji, sedangkan nilai K yang terlalu kecil dapat menyebabkan algoritma terlalu sensitive terhadap noise.

K-NN merupakan teknik klasifikasi yang sederhana, tetapi mempunyai hasil kerja yang cukup bagus. Beberapa karakter K-NN adalah sebagai berikut :

- K-NN merupakan algoritma yang menggunakan seluruh atau sebagian data latih untuk melakukan proses klasifikasi. Hal ini mengakibatkan proses prediksi yang sangat lama.
- K-NN tidak membedakan setiap fitur (attribut) data denga suatu bobot.
- Hal yang rumit dari K-NN adalah menentukan nilai K yang paling sesuai.
- Prinsip K-NN adalah memilih tetangga terdekat dan melakukan klasifikasi dengan *voting* terbanyak.

Karena K-NN konvensional adalah algoritma yang bersifat *lazy learner,* untuk melakukan klasifikasi K-NN memerlukan seluruh data [26], dan data juga harus sudah disertai dengan kelas atau target, hal ini disebabkan K-NN masuk kedalam kategori terpadu (*supervised*).

K-NN adalah algoritma klasifikasi yang banyak digunakan karena sederhana, mudah diterapkan, dan dapat dieksploitasi di berbagai domain aplikasi [28]. Dengan segala kekurangan dan kelebihannya, K-Nearest Neighbor (K-NN) menjadi salah satu dari *top ten* algoritma *data mining* [31].

### *G. Model Jarak*

Kesamaan kedua objek harus diukur untuk menentukan perbedaan dan kemiripan [16], salah satu cara untuk menentukan kemiripan data adalah dengan menggunakan model pengukuran jarak.

Terdapat banyak model pengukuran jarak, antara lain manhattan, euclidean, minkowsky, chebyshev, harmonic, dan lain sebagainya. Berikut ini adalah beberapa persamaan dari model jarak tersebut :

Pengukuran jarak *manhattan* menggunakan formula :

$$D(x,y) = ||x-y||_1 = \sum_{j=1}^{N} |x-y| \quad (13)$$

Pengukuran jarak *euclidean* menggunakan formula :

$$D(x,y) = ||x-y||_2 = \sqrt{\sum_{j=1}^{N} |x-y|^2} \quad (14)$$

Pengukuran jarak *chebyshev* menggunakan formula :

$$D(x,y) = ||x-y||_\lambda = \lim_{\lambda\to\infty} \sqrt[\lambda]{\sum_{j=1}^{N} |x-y|^\lambda} \quad (15)$$

Pengukuran jarak *minkowsky* menggunakan formula :

$$D(x,y) = ||x-y||_\lambda = \sqrt[\lambda]{\sum_{j=1}^{N} |x-y|^\lambda} \quad (16)$$

Dimana :

D adalah jarak antara data x dan y.
N adalah jumlah fitur (dimensi) data.
$\lambda$ adalah parameter jarak Minkowsky.

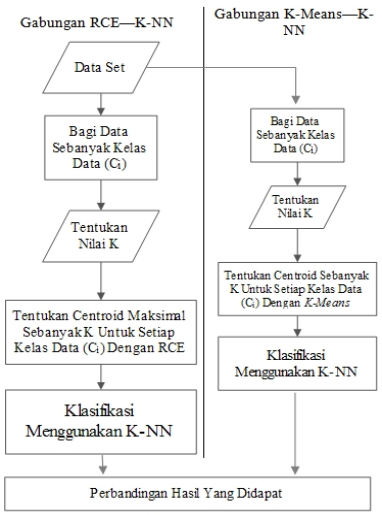
Secara umum *minkowsky* adalah generalisasi dari jarak yang ada seperti *euclidean* dan *manhattan* [15]. Lamda ( $\lambda$ ) merupakan parameter penentu dan bernilai bilangan positif dari 1 sampai dengan tak terhingga (∞), jika nilai $\lambda = 1$ maka ruang jarak minkowsky sama dengan *manhattan* [2], dan jika $\lambda = 2$ ruang jaraknya sama dengan *euclidean* [14], dan jika $\lambda = \infty$ sama dengan ruang jarak *chebyshev* [25].

Setiap model pengukuran jarak memiliki kelebihan tersendiri, *euclidean* cocok dalam menentukan jarak terdekat antara data, sedangkan *manhattan* cocok untuk mendeteksi *outlier* data.

## III. METODOLOGI PENELITIAN

Diagram penelitian dibawah ini merupakan gambaran proses kerja dari awal dilakukannya penelitian hingga akhir dari penelitian ini. Pada tahap awal penelitian akan dilakukan pengumpulan bahan penelitian dari berbagai sumber, seperti buku, jurnal, prosiding, artikel dan sumber lainnya yang relevan dalam penelitian ini.

Adapun diagram penelitian yang diusulkan dapat dilihat pada gambar 4 berikut :




Gbr 5 Diagram Proses Penelitian

Berdasarkan gambar tersebut, dapat dijelaskan tahapan pada penelitian ini adalah sebagai berikut :

- Data set. Data yang digunakan dalam penelitian adalah data subset yang diambil dari UCI Machine Learning Repository, antara lain iris, dan wine dimana 80% dari data set dijadikan data latih dan 20% dari data set digunakan pada pengujian.
  - Alur proses dalam penggabungan RCE dan K-NN pada penelitian ini adalah sebagai berikut :

1. Pembagian data set sejumlah kelas data (Ci). Data set yang memiliki kelas data sebanyak Ci akan dibagi kedalam kelompok data sebanyak Ci berdasarkan kelas data untuk kemudian akan dilakukan clustering.
2. Tentukan nilai K, dimana nilai K menunjukan jumlah maksimum cluster untuk setiap kelas data dan sebagai jumlah tetangga terdekat dalam klasifikasi dengan menggunakan K-NN.
3. Penentuan centroid menggunakan Rapid Centroid Estimation (RCE) untuk setiap kelas data ($C_i$). Pada setiap kelas data, akan ditentukan centroid dimana jumlah centroid yang dibentuk maksimal sebanyak K dari setiap kelas. Sehingga jumlah data yang nantinya akan digunakan untuk klasifikasi dalam K-NN paling banyak adalah $C_i$ x K.
4. Klasifikasi menggunakan K-NN. Klasifikasi dilakukan dengan menghitung kemiripan data yang baru keseluruh centroid dengan menggunakan rumus jarak euclaudien terhadap setiap kelas data dengan vote majority.

- Alur proses dalam penggabungan K-Means dan K-NN adalah sebagai berikut :
  1. Pembagian data set sejumlah kelas data ($C_i$). Data set yang memiliki kelas data sebanyak $C_i$ akan dibagi kedalam kelompok data sebanyak $C_i$ berdasarkan kelas data untuk kemudian akan dilakukan *clustering*.
  2. Tentukan nilai K, dimana nilai K menunjukan jumlah cluster untuk setiap kelas data dan sebagai jumlah tetangga terdekat dalam klasifikasi dengan menggunakan K-NN.
  3. Pengelompokan data dengan *K-Means* untuk setiap kelas data ($C_i$). Pada setiap kelas data, akan dilakukan *clustering* dimana jumlah *cluster* yang dibentuk sebanyak K. Sehingga jumlah data yang nantinya akan digunakan untuk klasifikasi dalam K-NN adalah sebanyak $C_i$ x K. Dimana data yang digunakan merupakan pusat *cluster* akhir dari setiap kelas data.
  4. Klasifikasi menggunakan K-NN. Klasifikasi dilakukan dengan menghitung kemiripan data yang baru keseluruh pusat *cluster* dari setiap kelas data dengan menggunakan *vote majority*.

• Hasil yang didapat akan dilakukan perbandingan untuk melihat teknik clustering manakah yang mampu meningkatkan kinerja pada algoritma K-NN.

## IV. HASIL DAN PEMBAHASAN

Untuk melihat kinerja dari gabungan RCE dan K-NN maka akan dilakukan perbandingan akurasi dengan K-NN konvensional dengan menggunkan data set iris dan wine yang berasal dari UCI Machine Learning Repository. Untuk mempermudah dalam perhitungan kinerja dari algoritma ini maka dilakukan pengimplementasian dengan menggunakan MATLAB dengan fungsi RCE yang telah dipublikasi pada situs Mathworks MATLAB[23].

Pengujian pertama dilakukan dengan menggunakan data iris, dimana setiap kelas data akan ditentukan nilai centroid maksimal sebanyak K=10. Pada algoritma RCE, didapati jumlah centroid pada kelas setosa sebanyak 10 centroid, pada kelas virginica sebanyak 9 centroid dan pada kelas versicolor sebanyak 7 centroid. Adapun nilai centorid tersebut dapat dilhat pada tabel dibawah ini :

TABEL I
NILAI CENTROID PADA DATA SET IRIS MENGGUNAKAN RCE

| Kelas | Nilai Attribut | | | |
|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
| Setosa | 5.24 | 3.77 | 1.60 | 0.30 |
| | 5.22 | 3.44 | 1.55 | 0.27 |
| | 4.87 | 3.45 | 1.24 | 0.20 |
| | 4.85 | 3.04 | 1.48 | 0.20 |
| | 5.42 | 3.83 | 1.58 | 0.32 |
| | 4.50 | 2.31 | 1.30 | 0.30 |
| | 4.46 | 3.08 | 1.33 | 0.18 |
| | 5.67 | 4.20 | 1.37 | 0.27 |
| | 4.92 | 3.49 | 1.25 | 0.21 |
| | 4.94 | 3.43 | 1.69 | 0.35 |
| Virginica | 7.02 | 3.13 | 5.81 | 2.05 |
| | 7.55 | 3.37 | 6.34 | 2.14 |
| | 6.38 | 3.32 | 5.60 | 2.40 |
| | 6.89 | 3.11 | 5.45 | 2.17 |
| | 6.44 | 2.93 | 5.51 | 2.00 |
| | 4.91 | 2.49 | 4.51 | 1.70 |
| | 5.85 | 2.83 | 5.02 | 1.93 |
| | 6.17 | 2.66 | 5.03 | 1.64 |
| | 7.68 | 2.76 | 6.75 | 2.15 |
| Versicolor | 6.45 | 3.02 | 4.51 | 1.42 |
| | 6.23 | 2.43 | 4.62 | 1.39 |
| | 6.82 | 3.04 | 4.82 | 1.50 |
| | 5.51 | 2.54 | 3.80 | 1.19 |
| | 6.07 | 2.89 | 4.17 | 1.35 |
| | 5.65 | 2.83 | 4.26 | 1.27 |
| | 5.00 | 2.30 | 3.28 | 1.02 |

Karena RCE merupakan pengembangan dari Particle Swarm Clustering (PSC), membuat RCE menentukan jumlah cluster yang optimum secara otomatis. Berbeda dengan K-Means jumlah cluster akan dibentuk sesuai dengan jumlah cluster yang diinginkan (dalam penilitian ini nilai K=10), tidak perduli apakah jumlah cluster itu optimum atau tidak.

Adapun nilai centroid akhir dari peng-cluster-an K-Means adalah sebagai berikut :

TABEL II NILAI CENTROID PADA DATA SET IRIS MENGGUNAKAN K-MEANS

| Kelas | Nilai Attribut | | | |
|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
| Setosa | 5.03 | 3.40 | 1.63 | 0.50 |
| | 4.37 | 3.03 | 1.27 | 0.17 |
| | 5.06 | 3.59 | 1.46 | 0.23 |
| | 5.43 | 3.65 | 1.60 | 0.28 |
| | 5.60 | 4.13 | 1.35 | 0.30 |
| | 4.60 | 3.60 | 1.00 | 0.20 |
| | 4.50 | 2.30 | 1.30 | 0.30 |
| | 5.00 | 3.25 | 1.30 | 0.20 |
| | 4.95 | 3.60 | 1.90 | 0.30 |
| | 4.80 | 3.06 | 1.48 | 0.20 |
| Virginica | 6.30 | 3.33 | 5.58 | 2.38 |
| | 7.68 | 2.85 | 6.58 | 2.18 |
| | 4.90 | 2.50 | 4.50 | 1.70 |
| | 6.86 | 3.32 | 5.82 | 2.34 |
| | 5.95 | 2.75 | 4.96 | 1.88 |
| | 6.43 | 2.87 | 5.67 | 2.17 |
| | 6.33 | 2.85 | 5.43 | 1.70 |
| | 7.20 | 3.03 | 6.00 | 1.83 |
| | 6.76 | 3.04 | 5.28 | 2.16 |
| | 7.80 | 3.80 | 6.55 | 2.10 |
| Versicolor | 5.47 | 2.47 | 3.88 | 1.20 |
| | 5.68 | 2.85 | 4.25 | 1.29 |
| | 6.63 | 2.83 | 4.67 | 1.40 |
| | 5.65 | 2.75 | 3.55 | 1.15 |
| | 5.00 | 2.30 | 3.28 | 1.03 |
| | 6.50 | 3.15 | 4.50 | 1.48 |
| | 6.27 | 2.33 | 4.60 | 1.43 |
| | 6.87 | 3.13 | 4.77 | 1.47 |
| | 6.18 | 2.88 | 4.38 | 1.30 |
| | 6.70 | 3.00 | 5.00 | 1.70 |

Kemudian nilai centroid ini akan dijadikan pedoman dalam melakukan klasifikasi menggunakan K-NN. Adapun hasil perbandingan akurasi dengan menggunakan nilai *centroid* dari RCE dan K-Means pada K-NN dapat dilihat pada tabel dibawah ini :

TABEL III
PERBANDINGAN AKURSAI PADA K-NN DENGAN NILAI CENTROID RCE DAN NILAI CENTROID K-MEANS (DATA SET IRIS)

| Nilai K | Akurasi RCE—K-NN | Akurasi K-Means—K-NN |
|---|---|---|
| 1 | 93.33% | 93.33% |
| 2 | 93.33% | 93.33% |
| 3 | 93.33% | 96.67% |
| 4 | 93.33% | 96.67% |
| 5 | 96.67% | 96.67% |
| 6 | 96.67% | 100.00% |
| 7 | 96.67% | 90.00% |
| 8 | 96.67% | 93.33% |
| 9 | 96.67% | 86.67% |
| 10 | 93.33% | 86.67% |
| Rata-rata | 95.00% | 93.33% |

Pada pengujian kedua dilakukan dengan menggunkan data set wine, proses yang sama pada data set iris juga dilakukan pada data set wine, dimana setiap kelas data akan ditentukan nilai centroid maksimal sebanyak K dengan menggunakan algoritma RCE, dimana jumlah *centroid* pada kelas 1 sebanyak 10 centroid, pada kelas 2 dan 3 juga mendapatkan 10 centroid. Adapun nilai centorid tersebut dapat dilhat pada tabel dibawah ini :

TABEL IV
NILAI CENTROID PADA DATA SET WINE MENGGUNAKAN RCE

| Kelas | Nilai Attribut | | | |
|---|---|---|---|---|
| | $X_1$ | $X_2$ | … | $X_{13}$ |
| 1 | 13.81 | 1.67 | … | 1292.48 |
| | 13.46 | 2.02 | … | 878.29 |
| | 13.71 | 1.71 | … | 1259.77 |
| | 13.80 | 1.88 | … | 1386.04 |
| | 13.48 | 1.70 | … | 1020.01 |
| | 13.87 | 1.72 | … | 1488.12 |
| | 13.81 | 2.10 | … | 1092.27 |
| | 13.44 | 1.74 | … | 1183.67 |
| | 13.48 | 3.12 | … | 746.58 |
| | 14.15 | 1.60 | … | 1668.01 |
| 2 | 12.25 | 2.04 | … | 451.90 |
| | 12.66 | 1.87 | … | 333.23 |
| | 12.23 | 1.90 | … | 298.75 |
| | 11.79 | 1.76 | … | 418.02 |
| | 12.45 | 2.85 | … | 382.40 |
| | 12.47 | 1.48 | … | 937.52 |
| | 12.02 | 3.20 | … | 566.65 |
| | 12.22 | 1.91 | … | 620.64 |
| | 12.34 | 1.58 | … | 686.21 |
| | 12.38 | 1.72 | … | 506.16 |
| 3 | 13.54 | 3.86 | … | 750.95 |
| | 13.41 | 3.88 | … | 556.63 |
| | 12.81 | 3.30 | … | 847.08 |
| | 13.05 | 3.31 | … | 486.88 |
| | 13.14 | 3.47 | … | 521.19 |
| | 13.25 | 2.39 | … | 641.49 |
| | 13.19 | 3.65 | … | 625.75 |
| | 13.32 | 3.44 | … | 681.11 |
| | 12.45 | 3.03 | … | 880.00 |
| | 13.38 | 2.62 | … | 779.88 |

Adapun nilai centroid akhir dari peng-cluster-an K-Means adalah sebagai berikut :

TABEL V
NILAI CENTROID PADA DATA SET WINE MENGGUNAKAN K-MEANS

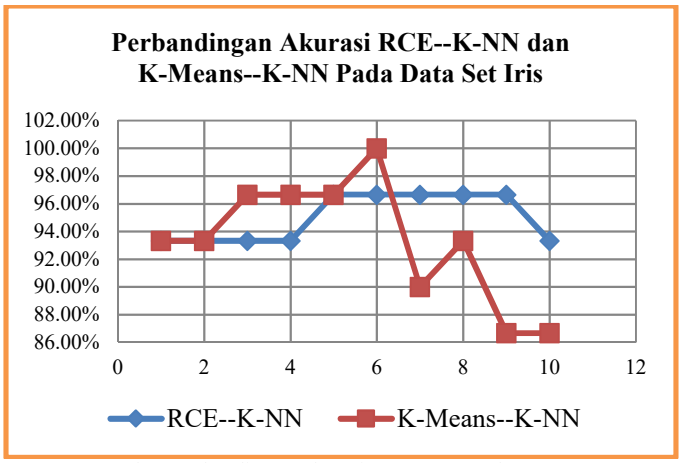
| Kelas | Nilai Attribut | | | |
|---|---|---|---|---|
| | $X_1$ | $X_2$ | … | $X_{13}$ |
| 1 | 13.89 | 1.71 | … | 1482.50 |
| | 13.37 | 1.67 | … | 1126.25 |
| | 13.47 | 1.69 | … | 1019.00 |
| | 13.54 | 3.20 | … | 745.00 |
| | 13.47 | 2.01 | … | 881.67 |
| | 14.19 | 1.59 | … | 1680.00 |
| | 13.79 | 1.71 | … | 1280.71 |
| | 13.99 | 2.29 | … | 1079.00 |
| | 13.77 | 1.90 | … | 1375.00 |
| | 13.66 | 1.68 | … | 1206.67 |
| 2 | 11.97 | 3.10 | … | 575.00 |
| | 12.35 | 1.59 | … | 493.25 |
| | 11.77 | 1.71 | … | 418.00 |
| | 12.23 | 1.91 | … | 319.00 |
| | 12.28 | 2.14 | … | 455.33 |
| | 12.65 | 2.80 | … | 512.50 |
| | 12.11 | 0.92 | … | 520.00 |
| | 12.47 | 2.79 | … | 377.00 |
| | 12.34 | 1.65 | … | 660.11 |
| | 12.47 | 1.52 | … | 937.00 |
| 3 | 13.36 | 2.56 | … | 780.00 |
| | 13.56 | 4.05 | … | 746.67 |
| | 13.21 | 3.47 | … | 526.67 |
| | 13.74 | 2.88 | … | 655.00 |
| | 13.78 | 2.76 | … | 615.00 |
| | 12.95 | 2.41 | … | 633.33 |
| | 12.86 | 3.35 | … | 841.67 |
| | 13.21 | 3.58 | … | 685.00 |
| | 12.87 | 4.61 | … | 625.00 |
| | 12.45 | 3.03 | … | 880.00 |

Adapun hasil perbandingan akurasi dengan menggunakan nilai *centroid* dari RCE dan K-Means pada K-NN dapat dilihat pada tabel dibawah ini :

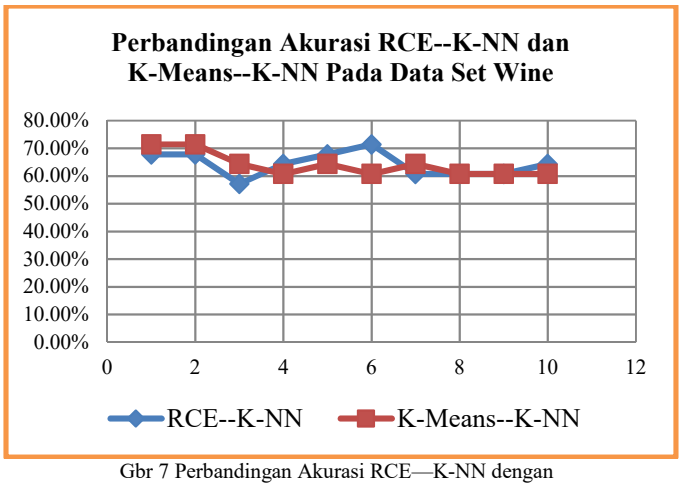
TABEL VI
PERBANDINGAN AKURSAI PADA K-NN DENGAN NILAI CENTROID RCE DAN NILAI CENTROID K-MEANS (DATA SET WINE)

| Nilai K | Akurasi RCE—K-NN | Akurasi K-Means—K-NN |
|---|---|---|
| 1 | 67.85% | 71.43% |
| 2 | 67.85% | 71.43% |
| 3 | 57.14% | 64.29% |
| 4 | 64.28% | 60.71% |
| 5 | 67.85% | 64.29% |
| 6 | 71.43% | 60.71% |
| 7 | 60.71% | 64.29% |
| 8 | 60.71% | 60.71% |
| 9 | 60.71% | 60.71% |
| 10 | 64.28% | 60.71% |
| Rata-rata | 64.29% | 63.93% |

Grafik nilai akurasi dari perbandingan RCE—K-NN dan K-Means—K-NN adalah sebagai berikut :




Gbr 6 Perbandingan Akurasi RCE—K-NN dengan K-Means—K-NN Pada Data Set Iris




Gbr 7 Perbandingan Akurasi RCE—K-NN dengan K-Means—K-NN Pada Data Set Wine

Pada data iris RCE—K-NN mendapati nilai akurasi paling tinggi yaitu sebesar 96.67% pada saat K bernilai 5, 6, 7, 8, dan 9. Dari grafik (gambar 5) dapat dilihat bahwa saat K bernilai 5 sampai dengan K bernilai 9 sudah mencapai hasil akurasi yang stabil, meskipun pada saat K bernilai 10, hasil akurasi mengalami penurunan kembali menjadi 93.33%.
Sedangkan K-Means—K-NN mendapati nilai akurasi paling tinggi yaitu sebesar 100% pada saat K bernilai 6. Namun saat K bernilai 7 nilai akurasi dari gabungan algoritma ini mengalami penurunan akurasi sebesar 10%, yang membuat algoritma ini mendapatkan nilai akurasi sebesar 90%, pada saat K bernilai 8 gabungan dari algoritma ini mengalami

peningkatan akurasi kembali yaitu sebesar 3.33% (akurasi saat K=8 adalah sebesar 93.33%), namun pada saat K bernilai 9 dan 10 kembali algoritma ini mengalami penurunan akurasi menjadi 86.67%.

Rata-rata akurasi RCE—K-NN yang dicapai dari data set iris ini adalah sebesar 95.00%, sedangkan K-Means—K-NN mampu mencapai rata-rata akurasi sebesar 93.33%. Pada grafik (Gambar 5) terlihat bahwa akurasi yang dihasilkan RCE—K-NN cendrung lebih stabil dari K-Means—K-NN.

Sedangkan pada data wine RCE—K-NN mendapati nilai akurasi paling tinggi sebesar 71.43% pada saat K bernilai 6 dan akurasi paling rendah sebesar 57.14% saat K bernilai 3. Dari grafik (gambar 6) dapat dilihat bahwa saat K bernilai 7 sampai dengan K bernilai 10 sudah mencapai hasil akurasi yang stabil.

Gabungan K-Means—K-NN pada data set wine mendapati nilai akurasi paling tinggi sebesar 71.43% pada saat K bernilai 1 dan 2. Sedangkan nilai akurasi paling rendah yang didapati dari K-Means—K-NN adalah sebesar 60.71% saat K bernilai 4, 6, 8, 9 dan 10.

Rata-rata akurasi RCE—K-NN yang dicapai dari data set iris ini adalah sebesar 64.29%, sedangkan K-Means—K-NN mampu mencapai rata-rata akurasi sebesar 63.93%. Pada grafik (Gambar 6) terlihat bahwa akurasi yang dihasilkan K-Means—K-NN cendrung lebih stabil dari RCE—K-NN.

## V. KESIMPULAN

Pada data iris, selisih rata-rata akurasi antara RCE—K-NN dan K-Means—K-NN adalah sebesar 1.67%. Dengan rata-rata hasil akurasi tertinggi dicapai oleh RCE—K-NN yaitu sebesar 95.00%. Pada data wine, selisih rata-rata akurasi antara RCE—K-NN dan K-Means—K-NN adalah sebesar 0.36%. Dengan rata-rata hasil akurasi tertinggi dicapai oleh RCE—K-NN yaitu sebesar 64.29%.

Dari tabel (tabel 2 dan tabel 4 ) dan grafik (gambar 5 dan gambar 6) dapat disimpulkan bahwa centroid yang didapatkan melalui perhitungan menggunakan RCE lebih baik dalam meningkatkan hasil akurasi pada K-NN. Namun dari grafik perbandingan pada data set wine, K-Means—K-NN memiliki perubahan akurasi yang lebih stabil dari RCE—K-NN.

## REFERENSI

[1] Amri Danades, A, et. al. "Comparison of Accuracy Level K-Nearest Neighbor Algorithm and Support Vector Machine Algorithm in Classification Water Quality Status". *International Conference on System Engineering and Technology*, October 2016, Pages 137-141. https://doi.org/10.1109/ICSEngT.2016.7849638.

[2] Ause Labellapansa, et. al. "Lambda Value Analysis on Weighted Minkowski Distance Model in CBR of Schizophrenia Type Diagnosis". *Fourth International Conference on Information and Communication Technologies (ICoICT)*, Volume , 2016, Pages 1-4. https://doi.org/10.1109/ICoICT.2016.7571898

[3] A. Szabo, et. al. "The Proposal of a Fuzzy Clustering Algorithm Based on Particle Swarm". *Third World Congress on Nature and Biologically Inspired Computing*. Desember 2011, Pages 469-465 , https://doi.org/10.1109/NaBIC.2011.6089630

[4] A. Szabo, et. al. "The Behavior of Particles in the Particle Swarm Clustering Algorithm". *IEEE International Conference on Fuzzy Systems (FUZZ)*. September 2010. https://doi.org/10.1109/FUZZY.2010.5584118

[5] Bain Khusnul Khotim, et. al. "A Genetic Algorithm For Optimized Initial Centers K-Means Clustering In SMEs". *Journal of Theoretical and Applied Information Technology*, Volume 90, Agustus 2016, Pages 23 – 30.

[6] Caiquan Xiong, et. al. "An Improved K-means text clustering algorithm By Optimizing initial cluster centers". *International Conference on Cloud Computing and Big Data*, Volume , 2016, Pages 265 - 268. https://doi.org/10.1109/CCBD.2016.059

[7] Eko Prasetyo. *Data Mining : Konsep dan Aplikasi Menggunakan MATLAB*, Yogyakarta : Andi Offset, 2012.

[8] Florin Gorunescu, et al. *Data Mining Concept, Model and Techniques*. Berlin : Springer-Verlag . 2011.

[9] Fuyuan Cao, et. al. "An Initialization Method For The K -Means Algorithm Using Neighborhood Model". *Computers and Mathematics with Applications*, Volume 58 Agustus 2009, Pages 474 – 483. https://doi.org/10.1016/j.camwa.2009.04.017

[10] Hosein Alizadeh, et. al. "A New Method for Improving the Performance of K Nearest Neighbor using Clustering Technique". *Journal of Convergence Information Technology*, Volume 4, Januari 2009, Pages 84 – 92, 10.4156/jcit.vol4.issue2.alizadeh

[11] Josè Valente de Oliveira, et. al (Editor). "Advance in Fuzzy Clustering and It's Applications". The Atrium, Southern Gate, Chichester. British Library Cataloguing in Publication Data. Jhon Willey and Son, Ltd. England, 2007.

[12] Jiawei Han, et. al. *Data Mining : Concepts and Techniques*. 2rd Edition. Amsterdam : Morgan Kaufmann. 2006.

[13] Jiawei Han, et. al. *Data Mining : Concepts and Techniques*. 3rd Edition. Amsterdam : Morgan Kaufmann. 2011.

[14] Josè M. Merigó, et. al. "The Induced Minkowski Ordered Weighted Averaging Distance Operator". *ESTYLF08, Cuencas Mineras (Mieres-Langreo), Congreso Espanol sobre Tecnologiasy Logica Fuzzy*, September 2008, Pages 35-41.

[15] Josè M. Merigó, et. al. "A New Minkowski Distance Based on Induced Aggregation Operators". *International Journal of Computational Intelligence Systems*, Volume 4, April 2011. 10.1080/18756891.2011.9727769

[16] K.M.N Mahyuddin, et. al.. "New Similarity". *IOP Conference Series: Materials Science and Engineering*, Volume 180, 2017. https://doi.org/10.1088/1757-899X/180/1/012297

[17] K.U.*Syaliman bin Lukman, et al." Analisa Nilai Lamda Model Jarak Minkowsky Untuk Penentuan Jurusan SMA (Studi Kasus di SMA Negeri 2 Tualang)". Jurnal Teknik Informatika dan Sistem Informasi. Jurnal Teknik Informatika dan Sistem Informasi. Volume 1, 2 Agustus 2015, e-ISSN: 2443-2229.*

[18] Max Bramer. *Principles of Data Mining*. London : Springer-Verlag, 2007, pp, 8.

[19] Margaret H Dunham, et. al. *Data Mining- Introductory and Advanced Topics*. Prentice Hall: USA. 2006.

[20] Mitchell Yuwono, et. al. "Fast Unsupervised Learning Method For Rapid Estimation Of Cluster Centroids". *IEEE Congress on Evolutionary Computation*, Juni 2012. https://doi.org/10.1109/CEC.2012.6256453

[21] Mitchell Yuwono, et. al. "Method For Increasing The Computation Speed Of An Unsupervised Learning Approach For Data Clustering". *IEEE World Congress on Computational Intelligence*, Juni 2012. https://doi.org/10.1109/CEC.2012.6252927

[22] Mitchell Yuwono, et. al. "Optimization Strategies for Rapid Centroid Estimation" . *Annual International Conference of the IEEE, Engineering in Medicine and Biology Society (EMBC)*, November 2012. https://doi.org/10.1109/EMBC.2012.6347413

[23] Mitchell Yuwono, et. al. (2012, Sep.).Rapid Centroid Estimation: An Efficient Particle Swarm Approach for Rapid Optimization of Cluster Centroids [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/38107-swarm-rapid-centroid-estimation--a-particle-swarm-clustering-algorithm [20 Agustus 2017]

[24]Mitchell Yuwono, et. al. "Data Clustering Using Variants of Rapid Centroid Estimation". *IEEE Transactions on Evolutionary Computation,* Volume 18, Juni 2014. https://doi.org/10.1109/TEVC.2013.2281545

[25]M. Koteswara Rao, et. al. "Face Recognition Using Different Local Feature with Different Distance Techniques", *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, Vol.2, No.1, Februari 2012, Pages 67-74, https://doi.org/10.5121/ijcseit.2012.2107

[26]Putu Wira Buana, et. al. "Combination of K-Nearest Neighbor and K-Means based on Term Re-weighting for Classify Indonesian News". *International Journal of Computer Applications*, Volume 50, Juli 2012, Pages 37 - 42. https://doi.org/10.5120/7817-1105.

[27]S.C.M. Cohen, et. al. "Data Clustering with Particle Swarms," in Proc 2006 IEEE Congress on Evolutionary Computations, 2006, Pages 1792-1798.

[28]Stefanos Ougiaroglou, et. al."Fast and Accuratek-Nearest Neighbor Classification using Prototype Selection by Clustering". *Panhellenic Conference on Informatics (PCI)*, 2012, Pages 168 - 173. 10.1109/PCi.2012.69

[29]Syahfitri Kartika Lidya, et. al. "Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (SVM) Dan K-Nearest Neighbor (K-NN)". *Seminar Nasional Teknologi Informasi dan Komunikasi,* 2015. ISSN: 2089-9815.

[30]Vijay Kumar, et. al. "Initializing Cluster Center for K-Means Using Biogeography Based Optimization". *Advances in Computing, Communication and Control*. Volume 123, 2011. Pages 448-456. https://doi.org/10.1007/978-3-642-18440-6_57

[31]Xindong Wu, et. al. *The Top Ten Algorithms in Data Mining*, New York : Taylor & Francis Group , 2009.

[32]Jiuyong Li (Ed). *AI 2010 : Advance in Artificial Intelligence*. Springer Verlag : Berlin. 2010.

[33]Oded Mainmon, et al. *Data Mining And Knowledge Discovery Handbook*. Springer : New York, 2010.