Penerapan Fungsi Exponential Pada Pembobotan Fungsi Jarak Euclidean Algoritma *K-Nearest Neighbor*

Muhammad Jauhar Vikri[1], Roihatur Rohmah[2]
[1]Sistem Informasi, Fakultas Sains dan Teknologi,
Universitas Nahdlatul Ulama Sunan Giri
[2]Sistem Komputer, Fakultas Sains dan Teknologi,
Universitas Nahdlatul Ulama Sunan Giri
E-mail: [1]*vikri@unugiri.ac.id*, [2] *roiha.rohmah@unugiri.ac.id*

Abstrak – *k-Nearest Neighbor (k-NN)* merupakan salah satu algoritma klasifikasi yang populer dan banyak digunakan untuk menyelesaikan kasus klasifikasi. Hal ini dikarenakan algoritma *k-NN* memiliki kelebihan seperti sederhana, mudah dijelaskan, dan mudah diterapkan. Namun, Algoritma *k-NN* memiliki kekurangan hasil klasifikasi sangat dipengaruhi oleh skala input data dan jarak *Euclidean* yang memperlakukan atribut data secara merata, tidak sesuai dengan relevansi masing-masing data atribut. Hal ini menyebabkan penurunan hasil klasifikasi. Salah satu cara meningkatkan performa akurasi klasifikasi dari algoritma *k-NN* adalah metode pembobotan pada fiturnya pada saat pengukuran jarak *Euclidean*. Fungsi *Exponential* pada pengukuran jarak *Euclidean* yang telah dioptimasi diaplikasikan ke dalam algoritma algoritma k-NN sebagai metode pengukuran jarak. Peningkatan performa metode *k-NN* dengan fungsi *Exponential* untuk pembobotan fitur pada *k-NN* akan dilakukan dengan eksperimen menggunakan cara *Data Mining*. Kemudian hasil dari performa metode tujuan akan dibandingkan dengan metode k-NN asli dan metode penelitian pembobotan k-NN terdahulu. Sebagai acuan hasil perhitungan jarak terdekat, pengambilan jarak tetangga terdekat pada k-NN akan ditentukan dengan nilai k=5. Setelah eksperimen dilakukan algoritma tujuan dibandingkan dengan algoritma k-NN, Wk-NN, dan DWk-NN. Secara berurutan hasil perbandingan memperoleh nilai rata-rata k-NN 85,87%, Wk-NN 86,98%, DWk-NN 88,19% dan algoritma k-NN yang diberikan pembobotan fungsi Exponential memperoleh nilai 90,17%.

Kata Kunci *— Klasifkasi, k-NN, Pembobotan Atribut, Fungsi Exponential, Jarak Euclidean*

Abstract – *k-Nearest Neighbor (k-NN) is one of the popular classification algorithms and is widely used to solve classification cases. This is because the k-NN algorithm has advantages such as being simple, easy to explain, and easy to implement. However, the k-NN algorithm has a lack of classification results that are strongly influenced by the scale of input data and Euclidean which treats attribute data evenly, not according to the relevance of each data attribute. This causes a decrease in the classification results. One way to improve the classification accuracy performance of the k-NN algorithm is the method of weighting its features when measuring the Euclidean distance. The exponential function of the optimized Euclidean distance measurement is applied to the k-NN algorithm as a distance measurement method. Improving the performance of the k-NN method with the Exponential function for weighting features on k-NN will be carried out by experimentation using the Data Mining method. Then the results of the performance of the objective method will be compared with the original k-NN method and the previous k-NN weighting research method. As a result of the closest distance decision, taking the closest distance to k-NN will be determined with a value of k=5. After the experiment, the goal algorithm was compared with the k-NN, Wk-NN, and DWk-NN algorithms. Overall the comparison results obtained an average value of k-NN 85.87%, Wk-NN 86.98%, DWk-NN 88.19% and the k-NN algorithm given the weighting of the Exponential function obtained a value of 90.17%.*

Keywords *— Classification, k-NN, Attribute Weighting, Exponential, Euclidean Distance*

1. PENDAHULUAN

k-Nearest neighbors (k-NN) diperkenalkan pertama kali pada tahun 1950[1][2], sebagai algoritma klasifikasi pada data mining dan digunakan dalam analisis pada kasus data statistik[3], [4] dengan menggunakan fungsi kedekatan jarak pada objek[1]. Algoritma k-NN yang paling banyak digunakan pada data mining [5], serta menjadi salah satu metode data mining yang banyak digunakan karena sedehana dan efektif dalam menangani teknik klasifikasi [6][7]. Dengan mengunakan asumsi bahwa nilai k yang terdapat pada algoritma k-NN adalah jumlah tetangga terdekat. Sebagai pembelajaran yang sederhana dan mampu menghasilkan akurasi yang baik. Secara skema yang diterapkan k-NN mampu mendeskripsikan atribut atau fitur dari sebuah data[8]. Sebagai algoritma yang digunakan pada kasus data mining, algoritma k-NN dapat diklasifikasi menjadi kategori k-NN terstruktur atau *Structure base k-NN* dan k-NN tidak terstruktur *Non-structure base k-NN*[9], [10]. Algoritma terstruktur tidak banyak digunakan karena dianggap kurang efisien dan membutuhkan waktu yang relatif lebih lama dalam pencarian variabel k pada saat awal proses, khususnya terhadap data dengan atribut banyak, sehingga Teknik *Non-structure base k-NN* dianggap lebih sederhana dan efisien namun berpengaruh pada akurasi terhadap metode penentuan nilai k dan penentuan pengukuran jarak terdekat.

Penerapan fungsi kedekatan jarak dapat menjadi salah penentu akurasi pada k-NN[5][3]. Fungsi pengukuran jarak berbasis *Euclidean* atau dikenal dengan *Euclidean distance* menjadi fungsi penentu jarak terdekat yang paling umum digunakan pada k-NN[11]. Selain pengukuran jarak berbasis gometri Euclid, fungsi jarak yang digunakan pada algoritma k-NN adalah *Minkowski distance* dan *Manhattan distance*[10]. Kinerja algoritma k-NN sangat dipengaruhi dari beberapa faktor, diantaranya adalah penggunaan fungsi jarak, penentuan nilai k dan nilai atribut yang tidak relevan[6][3]. Kinerja pengukuran jarak terdekat pada pengukuran jarak berbasis *Euclidean* dapat di atasi dengan menerapkan pembobotan fitur.

Pada tahun 2019 telah diusulkan pembaruan dari metode berbasis fungsi kedekatan jarak yang diberikan nama *pada publikasi A Generalized Mean Distance k-Nearest Neighbors (GMDKNN)* oleh Gou[3]. Metode *GMDKNN* terinspirasi karena penentuan nilai k pada algoritma k-NN sangat mempengaruhi akurasi k-NN pada keputusan klasifikasi [2], namun *GMDKNN* tidak bereksperimen dalam penentuan k pada k-NN. Metode *GMDKNN* menggunakan nilai k yang sama pada saat eksperimen pada semua data uji dan pada dataset dengan isu sensitif terhadap nilai k. GMDKNN menggunakan 20 dataset UCI dan 8 dataset KEEL dengan 6 algoritma pembanding. Dalam ekperimennya menerapkan validasi split pada scenario pengujian dataset dan dibandingkan dengan algoritma k-NN, WKNN, *Local mean based* k-*nearest neighbor* (LMKNN), *Pseudo nearest neighbor* (PNN), *Local mean based pseudo nearest neighbor* (LMPNN), dan *Multi local mean based k-harmonic nearest neighbor* (MLMKHNN). Pada pengujian eksperimen pertama menggunakan dataset KEEL diantaranya Newth, Tae, Phoneme, Spambase, Band, Dermat, Ring, dan Texture metode GMDKNN membuktikan akurasi meningkat pada 4 dataset KEEL dan akurasi kurang baik pada 2 dataset KELL. Hasil perolehan akurasi rata-rata pada metode GMDKNN adalah 85.71%. Hal ini menunjukkan hasil terbaik dibandingkan dengan metode pembandingnya. Kemudian pada eksperimen kedua, menggunakan dataset UCI akurasi GMDKNN membuktikan akurasi yang baik hampir di semua dataset UCI.

Dari uraian state of the art pada analisis algoritma k-NN memiliki kelemahan terhadap nilai atribut sebagai nilai input data, dan kelemahan tersebut dapat diatasi dengan teknik normalisasi data. Namun beberapa studi menyatakan bahwa k-NN merupakan metode yang sangat rentan terhadap data outlier dan noise[12]. Metode k-NN menerapkan teknik klasifikasi berdasarkan kedekatan jarak yang diukur dengan fungsi jarak Euclidean disntance[1][13].

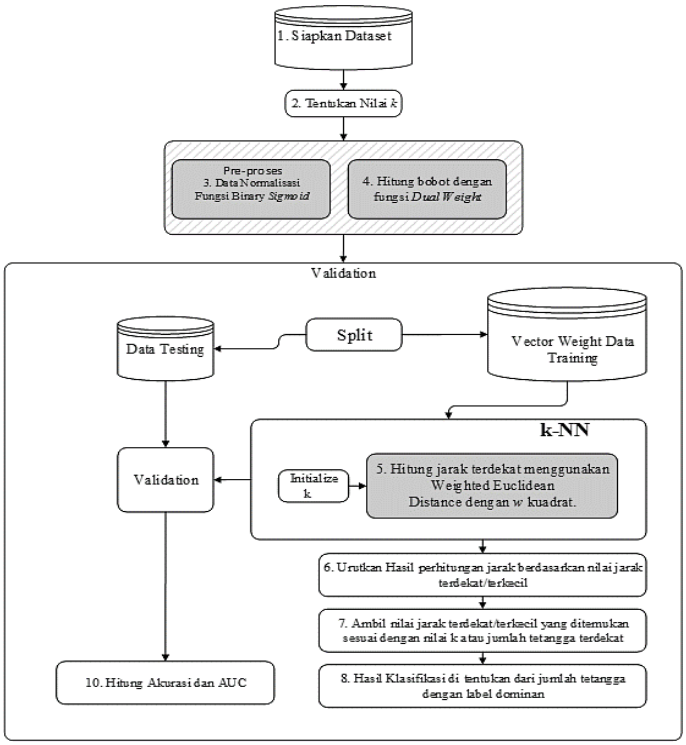
Fungsi pengukuran jarak berbasis fungsi Euclidean sangat rentan terhadap terhadap atribut yang tidak relevan[14][15]. hal ini disebabkan Fungsi pengukuran jarak berbasis fungsi Euclidean menerapkan perlakuan atribut data secara sama tanpa memperhatikan relevansi data, hal ini tentu akan menurunkan akurasi dari k-NN. Kelemahan pada kasus ini dapat diatasi dengan metode pembobotan fitur, dengan menambahkan bobot pada Fungsi pengukuran jarak berbasis fungsi Euclidean. Penentuan bobot terinspirasi dan diperolah dari teknik yang digunakan algoritma DWKNN yaitu Dual Weighted k-NN pada Fungsi pengukuran jarak berbasis fungsi Euclidean, kemudian cara pembobotannya pada Fungsi pengukuran jarak berbasis fungsi Euclidean menerapkan pembobotan kuadrat. Pembobotan kuadrat menjadi salah satu cara optimal yang

digunakan dalam pembobotan pada Fungsi pengukuran jarak berbasis fungsi Euclidean atau Euclidean distance [5].

Meningkatkan akurasi dari algoritma k-NN dengan menerapkan normalisasi berbasis fungsi exponnetial untuk mengatasi masalah relevansi data dengan skala input pada model pembobotan dan diintegrasikan dengan fungsi jarak weighted fungsi Euclideanmerupakan tujuan utama penelitian ini. Untuk menunjukkan akurasi algoritma k-NN pada penelitian ini menggunakan pengukuran akurasi dibandingkan dengan metode terdahulunya yaitu algoritma k-NN, WKNN, DWKNN. Hasil eksperimen menunjukkan perbandingan yang baik terhadap metode k-NN, WKNN, dan DWKNN, metode usulan memperoleh hasil yang baik dengan percobaan 3 dataset dari UCI. Ekperimen ditentukan dengan variasi nilai k=1, k=2, k=3, k=4, dan k=5 dengan memperhatikan mayoritas label pada tetangga terdekat. Secara berurutan hasil perbandingan memperoleh nilai rata-rata k-NN 85,87%, Wk-NN 86,98%, DWk-NN 88,19% dan algoritma k-NN yang diberikan pembobotan fungsi Exponential memperoleh nilai 90,17%.

2. METODE PENELITIAN

Pada penelitian ini akan diusulkan metode dengan pembobotan pada pendekatan pengukuran jarak menggunakan metode pembobotan pada fungsi pengukuran jarak berbasis *Euclidean Distance* untuk menangani permasalahan perlakuan atribut sama pada komputasi pendekatan jarak dan ketidak sesuaian terhadap relevansi masing-masing atribut pada fungsi pengukuran jarak berbasis Euclidean Distance. Metode ini diberi Exponential Weight Distance k-Nearest Neighbor. Data dengan jarak terdekat yang diambil disesuaikan dengan nilai k yang sebelumnya telah ditentukan.



Gambar 1, Alur metode knn dengan fungsi exponential untuk pembobotan fitur menggunakan Euclidean

Alur metode yang diusulkan dapat dilihat pada Gambar 1 adalah sebagai berikut:

a. Menyiapkan dataset, masukan dataset memiliki himpunan (X_{ij}, Y_n) dimana *i* adalah urutan data (1, 2, 3, …, i_n) dimana variabel *i* sampai dengan *i* ke n, dan j adalah urutan atribut (1, 2, 3, …, j_n) dimana nilai variabel *j* sampai dengan atribut ke n. dan Y adalah label dari data ke i;
b. Tentukan nilai *k* pada *k*-NN dimana *k* adalah jumlah tetangga terdekat, pada eksperimen ini menggunakan scenario variasi nilai k. Penentuan nilai *k* dengan cara input secara manual menggunakan nilai *k*=1, *k*=2, , *k*=3, , *k*=4, *k*=5;
c. Normalisasi pelatihan dataset berbasis *Exponential Smoothing Weight Distance k-Nearest Neighbor* mengunakan fungsi, normalisasi ini difungsikan untuk mencari pembobotan; Nilai distribusi fungsi exponnetial menggunakan distribusi antara [0,1] dimana x adalah data, x_{max} dan x_{min} merepresentasikan nilai maksimum dan minimum dari seluruh data [16]; Penulis Yu dan Xu pada tahun 2014 memaparkan bahwa rentang nilai [0.1, 0.9] adalah rentang nilai terbaik yang digunakan selama melakukan eksperimen.
d. Hitung pembobotan kuadrat (w^2) k-NN dengan fungsi fungsi;

$$w_i = f(x) = \begin{cases} \frac{d(x,x_k^{NN}) - d(x,x_i^{NN})}{d(x,x_k^{NN}) - d(x,x_1^{NN})} x \frac{d(x,x_k^{NN}) + d(x,x_1^{NN})}{d(x,x_k^{NN}) + d(x,x_i^{NN})} \\ 1 \end{cases}$$

..... (1)

$$Dimana, \quad \begin{matrix} f(x), if\ d(x,x_k^{NN}) \neq d(x,x_1^{NN}) \\ 1, \quad if\ d(x,x_k^{NN}) = d(x,x_1^{NN}) \end{matrix}$$

e. Hitung jarak terdekat antara data latih dan data uji, dengan pembobotan *w* menggunakan *Exponential Smoothing Weight Distance k-Nearest Neighbor*;
$Euclidean\ D. = \sum_{n=1}^{n=m} \sqrt{W^2(X_{dtraining} - X_{dtesting})^2}$
..... (2)
f. Urutkan hasil perhitungan jarak antara data latih dengan data uji menggunakan fungsi pengukuran jarak berbasis fungsi Euclideanmulai dari jarak terkecil sampai dengan jarak terbesar;
g. Ambil hasil perhitungan jarak yang sudah diurutkan sesuai dengan nilai tetangga terdekat *k*;
h. Hasil klasifikasi ditentukan dari mayoritas label yang dihasilkan atau ditemukan dari jarak terdekat;
i. Klasifikasi data yang baru dengan menggunakan algoritma kNN;
j. Hitung *accuracy dan root measn square error;;*

2.1. Pengumpulan Data

Pada penelitian ini data ekperimen yang digunakan berasal dari data publik yang telah digunakan oleh penelitian terdahulu, berdasarkan penelitian terdahulu, terdapat tiga dataset yang paling sering digunakan pada artikel yang membahas peningkatan performa algoritma k-nearest neighbor sebagai berikut :

Tabel 1, Dataset UCI digunakan untuk proses eksperimen

No	Datasets	Databases	Feature	Sample	Classes	Testing
1	Seed	UCI	7	210	3	60
2	Glass	UCI	9	146	2	53
3	Wine	UCI	13	178	3	58

3. HASIL DAN PEMBAHASAN

Eksperimen pada metode tujuan menggunakan dataset Seeds dari UCI sebagai contoh perhitungannya, sebagian dataset Seeds uji dataset ditampilkan pada Tabel 2. dan sebagian dataset Seeds pelatihan ditampilkan pada Tabel 3. sebagai pembeda dari metode pembanding metode tujuan menerapkan proses pencarian bobot pada data pelatihan yang digunakan pada fungsi pengukuran jarak berbasis Euclidean distance. pembobotan pada metode tujuan mengintegrasikan antara fungsi exponential dengan pembobotan dual weighted k-nn. Hasil dari bobot tersebut akan diformulasikan ke dalam fungsi kuadrat pada pengukuran jarak berbasis fungsi Euclideansebagai berikut:

$$w_{ij} = \left\{\frac{d_k^{NN} - d_i^{NN}}{d_k^{NN} - d_1^{NN}} x \frac{d_k^{NN} + d_1^{NN}}{d_k^{NN} + d_i^{NN}}\right\}, dimana\ d_k^{NN} \neq 1, dan\ 1, jika\ d_k^{NN} = d_1^{NN}$$

Kemudian bobot wij dimasukkan ke fungsi pengukuran jarak berbasis gometri Euclid:

$$Euclidean\ D. = \sum_{n=1}^{n=m} \sqrt{W^2(X_{dtraining} - X_{dtesting})^2}$$

Tabel 2. Sample data uji (Seed)

ID	attr 1	attr 2	attr 3	attr 4	attr 5	attr 6	attr 7	Label
(*x'1*)	15.26	14.84	0.871	5.763	3.312	2.221	5.22	kama
(*x'2*)	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	kama
(*x'3*)	14.29	14.09	0.905	5.291	3.337	2.699	4.825	kama
…	…	…	…	…	…	…	…	…
(*x'60*)	11.75	13.52	0.8082	5.444	2.678	4.378	5.31	Canadian
(*x'61*)	11.49	13.22	0.8263	5.304	2.695	5.388	5.31	Canadian
(*x'62*)	12.54	13.67	0.8425	5.451	2.879	3.082	5.491	Canadian

Tabel 3. Sample data training (Seed)

ID	attr 1	attr 2	attr 3	attr 4	attr 5	attr 6	attr 7	Label
(*x'1*)	14.16	14.4	0.8584	5.658	3.129	3.072	5.176	kama
(*x'2*)	14.11	14.26	0.8722	5.52	3.168	2.688	5.219	kama
(*x'3*)	15.88	14.9	0.8988	5.618	3.507	0.7651	5.091	kama
…	…	…	…	…	…	…	…	…
(*x'146*)	13.2	13.66	0.8883	5.236	3.232	8.315	5.056	Canadian
(*x'147*)	13.32	13.94	0.8613	5.541	3.073	7.035	5.44	Canadian
(*x'148*)	13.07	13.92	0.848	5.472	2.994	5.304	5.395	Canadian

Keterangan atribut Tabel:

a. attr 1 = area A
b. attr 2 = *perimeter* P
c. attr 3 = *compactness* C = 4*pi*A/P^2
d. attr 4 = *length of kernel*
e. attr 5 = *width of kernel*
f. attr 6 = *asymmetry coefficient*
g. attr 7 = *length of kernel groove*
h. label = (1)Kama, (2)Rosa and (3)Canadian

Diasumsikan dari data pelatihan telah diketahui nilai Min dan Max dari masing-masing atribut,

Tabel 4. Nilai MIN dan MAX dari data pelatihan

MIN	**10.59**	**12.41**	**0.8099**	**4.899**	**2.63**	**0.7651**	**4.519**
MAX	20.97	17.25	0.9183	6.675	3.991	8.456	6.55

Maka menggunakan fungsi exponential dapat diformulasikan untuk menormalisasi data pelatihan;

$$y = (\frac{x - x_{min}}{x_{max} - x_{min}} x(0.9 - 0.1)) + 0.1 \quad \text{.....(3)}$$

Hasil data pelatihan hasil normalisasi menggunakan fungsi exponnetial pada Tabel berikut:

Tabel 5. Hasil normalisasi dengan fungsi exponnetial

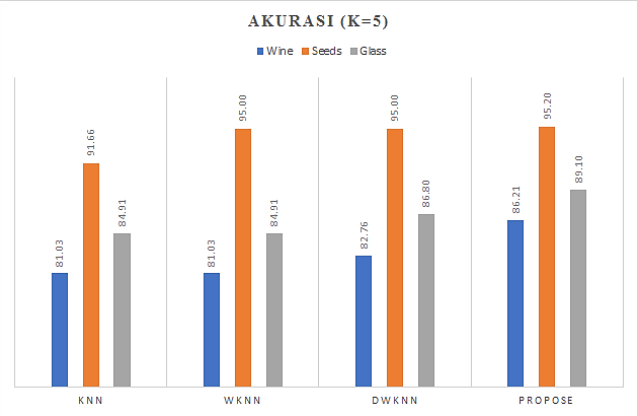
ID	attr 1	attr 2	attr 3	attr 4	attr 5	attr 6	attr 7	L
(*x'1*)	0.1034	0.10367	0.09062	0.09524	0.09281	0.0927	0.094	1
(*x'2*)	0.1033	0.10353	0.09063	0.09511	0.09285	0.09238	0.0948	1
(*x'3*)	0.1050	0.10415	0.09066	0.09521	0.09317	0.09053	0.0947	1
…	…	…	…	…	…	…	…	…
(*x'146*)	0.1025	0.10295	0.0906	0.09484	0.09291	0.09780	0.0946	3
(*x'147*)	0.1026	0.10322	0.09062	0.09513	0.09275	0.09657	0.0950	3
(*x'148*)	0.1023	0.10320	0.09061	0.09506	0.09268	0.09490	0.0949	3

Pada Tabel 7 ditampilkan hasil pengukuran kinerja metode k-NN, rata-rata akurasi 85.87%. Hasil pengukuran kinerja metode Wk-NN, rata-rata akurasi 86.98%. Hasil pengukuran kinerja metode DWKNN, rata-rata akurasi 88.19%. Hasil pengukuran kinerja metode tujuan dengan parameter k=5, rata-rata akurasi 90.17%.

Tabel 6. Hasil Akurasi dengan ekperimen k=5

Dataset	AKURASI % (k=5)			
	kNN	WkNN	DWkNN	Propose
Wine	81.03	81.03	82.76	86.21
Seeds	91.66	95.00	95.00	95.20
Glass	84.91	84.91	86.80	89.10
Rata-rata	85.87	86.98	88.19	90.17

Pada Tabel 7. Menunjukkan metode usulan lebih unggul dari metode pembanding. Dengan nilai rata-rata akurasi sebesar 90.17%. Dataset Wine diketahui memiliki 3 kelas klasifikasi, dataset Seeds diketahui memiliki 3 kelas klasifikasi dan dataset Glass memiliki 2 kelas klasifikasi.



Gambar 2, Perbandingan hasil eksperimen datamining data UCI terhadap beberapa algoritma k-NN

Pada Gambar 2. dapat disimpulkan secara visual bahwa hasil pengukuran akurasi pada metode usulan kalah terhadap metode pembanding dengan urutan nilai mulai akurasi tertinggi adalah Metode tujuan, DWKNN, WKNN dan KNN.

4. SIMPULAN

Metode *Exponential Weight Distance k-NN* (EWKNN) dibandingkan dengan k-NN, Wk-NN dan DWKNN. Evaluasi kinerja metode yang digunakan adalah akurasi. Hasil eksperimen pada tiga dataset publik dari UCI Machine Learning Repository diantaranya: wine, glass dan seeds metode EWKNN terbukti dapat meningkatkan kinerja algoritma kNN. Metode EWKNN memiliki akurasi rata-rata tertinggi sebesar 90.17% mengungguli k-NN, Wk-NN dan DWKNN. Hasil penelitian menunjukkan bahwa metode EWKNN dengan menerapkan Exponential weight dan Dual Weighting diintegrasikan dengan fungsi jarak pembobotan fitur pada fungsi pengukuran jarak berbasis fungsi Euclideanuntuk mengatasi fungsi pengukuran jarak berbasis fungsi Euclidean yang memperlakukan atribut data secara sama, tidak sesuai dengan relevansi masingmasing atribut data dapat meningkatkan kinerja algoritma k-NN. Dengan demikian masalah penelitian ini telah terjawab dan tujuan penelitian telah tercapai

5. SARAN

Pada pengembangan penelitian lebih lanjut, peningkatan performa algoritma k-NN pada kasus metode data mining klasifikasi menggunakan analisis perhitungan jarak pada k-NN masih sangat relevan. Hal ini ditunjukkan sebagaian besar cara eksperimen peningkatan performa algoritma k-NN banyak digunakan pada riset publikasi ilmiah. Pada penelitian ini kasus yang ditemukan adalah bahwa algoritma k-NN sangat dipengaruhi oleh skala input data dan fungsi jarak

Euclidean yang memperlakukan atribut data secara merata, tidak sesuai dengan relevansi masing-masing data atribut.

DAFTAR PUSTAKA

[1] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.

[2] M. Bicego, "Weighted K-Nearest Neighbor Revisited," pp. 1643–1648, 2016.

[3] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, "A generalized mean distance-based k -nearest neighbor classifier," *Expert Syst. Appl.*, vol. 115, pp. 356–372, 2019, doi: 10.1016/j.eswa.2018.08.021.

[4] J. Gou, "A New Distance-weighted k -nearest Neighbor Classifier," no. November 2011, 2014.

[5] P. Cao *et al.*, "Nonlinearity-aware based dimensionality reduction and over-sampling for AD/MCI classification from MRI measures," *Comput. Biol. Med.*, vol. 91, pp. 21–37, Dec. 2017, doi: https://doi.org/10.1016/j.compbiomed.2017.10.002.

[6] I. Witten, E. Frank, and M. Hall, *Data Mining Practical Machine Learning Tools and Techniques Third Edition*, vol. 277, no. Tentang Data Mining. 2011.

[7] D. Devi, S. kr. Biswas, and B. Purkayastha, "Redundancy-driven modified Tomek-link based undersampling: A solution to class imbalance," *Pattern Recognit. Lett.*, vol. 93, pp. 3–12, Jul. 2017, doi: https://doi.org/10.1016/j.patrec.2016.10.006.

[8] X. Zheng, Z. Lin, H. Xu, C. Chen, and T. Ye, "Efficient learning ensemble SuperParent-one-dependence estimator by maximizing conditional log likelihood," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7732–7745, Nov. 2015, doi: https://doi.org/10.1016/j.eswa.2015.05.051.

[9] U. R. Yelipe, S. Porika, and M. Golla, "An efficient approach for imputation and classification of medical data values using class-based clustering of medical records," *Comput. Electr. Eng.*, vol. 0, pp. 1–18, 2017, doi: 10.1016/j.compeleceng.2017.11.030.

[10] P. Rani, "A Review of various KNN Techniques," vol. 5, no. Viii, pp. 1174–1179, 2017.

[11] X. Wu *et al.*, *Top 10 algorithms in data mining*. 2008.

[12] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Third Edition*. Elsevier, 2011.

[13] S. Hosseini, B. Turhan, and M. Mäntylä, "A benchmark study on the effectiveness of search-based data selection and feature selection for cross project defect prediction," *Inf. Softw. Technol.*, vol. 95, pp. 296–312, Mar. 2018, doi: https://doi.org/10.1016/j.infsof.2017.06.004.

[14] J. Xia *et al.*, "Adjusted weight voting algorithm for random forests in handling missing values," *Pattern Recognit.*, vol. 69, pp. 52–60, 2017, doi: 10.1016/j.patcog.2017.04.005.

[15] J. P. Kandhasamy and S. Balamurali, "Performance Analysis of Classifier Models to Predict Diabetes Mellitus," *Procedia - Procedia Comput. Sci.*, vol. 47, pp. 45–51, 2015, doi: 10.1016/j.procs.2015.03.182.

[16] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Appl. Energy*, vol. 134, pp. 102–113, 2014, doi: 10.1016/j.apenergy.2014.07.104.