Analisis Perbandingan Metode *Harmonic Mean* dan *Local Mean Vector* Dalam Penyeleksian Tetangga Pada Algoritma KNN

Muhammad Al Ichsan Nur Rizqi Said[1)], Mohammad Reza Faisal[2)], Dwi Kartini[3)], Irwan Budiman[4)], Triando Hamonangan Saragih[5)]

[1)2)3)4)5)] Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lambung Mangkurat
Banjarbaru Selatan, Kota Banjar Baru
[1)] ichsan.said13@gmail.com
[2)] reza.faisal@ulm.ac.id
[3)] dwikartini@ulm.ac.id
[4)] irwan.budiman@ulm.ac.id
[5)] triando.saragih@ulm.ac.id

Abstrak

Algoritma K Nearest Neighbour (KNN) merupakan salah satu algoritma klasifikasi yang telah digunakan pada banyak penelitian, namun KNN memiliki beberapa kekurangan diantaranya adalah pada pemilihan jumlah tetangga terdekat. Jika jumlah tetangga terdekat terlalu kecil maka akan sensitif terhadap derau (*noise*) dan jika jumlah tetangga terdekat terlalu besar kemungkinan ada tetangga *outlier* dari kelas lain. *Majority Voting* juga merupakan metode yang sederhana dan ini bisa jadi masalah jika jarak bervariasi. Salah satu solusi untuk masalah *outlier* adalah menggunakan *Local Mean Vector* dengan menambahkan *Harmonic Mean* untuk membantunya. Penelitian ini bertujuan untuk mengetahui perbandingan kinerja teknik penyeleksian tetangga terakhir yang didapatkan menggunakan *Local Mean Vector* dan *Harmonic Mean.* Dari Hasil dari penelitian ini menunjukkan bahwa teknik penyeleksian tetanggal berbasis *Local Mean Vector* dan *Harmonic Mean* memberikan akurasi lebih baik yaitu sebesar 0,78 dibandingkan dengan teknik *Majority Voting* dengan akurasi sebesar 0.75.

Kata kunci: *Nearest Neighbour, K Nearest Neighbour, Local Mean Vector, Harmonic Mean, Majority Voting,* Penyeleksian.

Abstract

The K Nearest Neighbour (KNN) algorithm is one of the classification algorithms that has been used in many studies, but KNN has several shortcomings, including the selection of the number of nearest neighbors. If the number of nearest neighbors is too small, it will be sensitive to noise data and if the number of nearest neighbors is too large, there may be outlier neighbors from other classes. Majority Voting is also a simple method and this can be a problem if the distance varies. One solution to the outlier problem is to use Local Mean Vector by adding Harmonic Mean to help it. This study will select the neighbors obtained using the Local Mean Vector and Harmonic Mean so that the last neighbor remains. From the results obtained, namely KNN using Majority Voting gets an accuracy of 0.752998731, and KNN with Neighbor Selection based on Local Mean Vector and Harmonic Mean gets greater accuracy, namely 0.780791833.

Keywords: *Nearest Neighbour, K Nearest Neighbour, Local Mean Vector, Harmonic Mean, Majority Voting, Selection*

1. PENDAHULUAN

Pembelajaran mesin (*Machine Learning*) adalah studi ilmiah tentang algoritma dan model statistik yang digunakan sistem komputer untuk melakukan tugas-tugas tertentu tanpa menggunakan instruksi

DOI: 10.34128/jsi.v9i2.376 *Received: January 3rd, 2022*
Accepted: November 24th, 2023

eksplisit, mengandalkan pola dan inferensi sebagai gantinya. Ini dikenal sebagai bagian dari kecerdasan buatan [1]. Salah satu fungsi kecerdasan buatan adalah klasifikasi, yang dimana klasifikasi adalah sebuah proses yang dapat menentukan dan membedakan sebuah objek yang label nya belum diketahui [2]. Algoritma K Nearest Neighbour (KNN) adalah salah satu metode yang digunakan untuk analisis klasifikasi, namun beberapa dekade terakhir metode KNN juga digunakan untuk prediksi masa studi mahasiswa [3], klasifikasi sentimen komentar pada helpdesk [4], klasifikasi ham dan spam email [5], dan klasifikasi citra kanker serviks [6]. KNN mencari sebanyak k tetangga berdasarkan jarak terdekat antara data uji dengan data latih, lalu diambil keputusan dengan mengambil jumlah kelas terbanyak untuk bisa melakukan klasifikasi terhadap data yang diuji [7]. Meski metode KNN memiliki kelebihan, masih ada beberapa masalah yang harus diselesaikan. Pada aturan KNN, pemilihan jumlah tetangga terdekat atau bisa disebut sebagai k memiliki dua isu yaitu jika k terlalu kecil, hasil dari klasifikasi akan sensitif terhadap derau (*noise* data), dan label yang ambigu karena kesalahan pemberian label. Sebaliknya, jika k terlalu besar, kemungkinan ada beberapa tetangga yang bersifat *outlier* dari tetangga yang lain. Lebih jelasnya performa dari klasifikasi KNN sangat sensitif terhadap k yang dipilih, umumnya angka yang dipilih sebagai nilai k adalah angka ganjil [8] dengan nilai minimal di lebih besari dari jumlah class, namun tidak menutup kemungkinan menggunakan nilai dari dimulai dari 1, 2, 3 dan seterusnya [9]. Terlebih lagi, *Majority Voting* adalah metode yang paling sederhana untuk pemilihan label pada KNN, ini bisa jadi masalah jika tetangga terdekat memiliki jarak yang bervariasi [10]. Serta ada kemungkinan dua kelas dengan jumlah terbanyak yang sama [11]. Salah satu hal menghadapi data *outlier* salah satunya adalah mengeksplorasi *Local Mean Vector* [12].

Terdapat penelitian yang sudah mengadaptasi *Local Mean Vector* terhadap algoritma klasifikasi KNN. Salah satunya pada penelitian memodifikasi metode *Local Mean Based k-Nearest Centroid Neighbour* dengan tambahan *Harmonic Mean distance* karena lebih handal pada *Local Mean Vector* dengan rentang error di antara 0.4 [9]. Metode yang akan diajukan akan mengadaptasi penyeleksian tetangga. Penyeleksian sendiri pernah dilakukan namun dengan istilah lain yaitu reduction pada metode yang sudah pernah dibuat bernama *Prototype Selection for k-Nearest Neighbour using Geometric Median*, namun memiliki tujuan untuk meningkatkan runtime dan menyeleksi data train dengan hasil akurasi yang lebih rendah [13].

Berdasarkan penjelasan KNN telah banyak digunakan pada banyak kasus klasifikasi namun masih memiliki kekurangan dalam penyeleksian tetangga yang dapat meningkatkan kinerja klasifikasi. Karena hal tersebut maka diusulkan metode untuk memperbaiki kekurangan KNN yang telah ada dengan menerapkan penyeleksian tetangga berdasarkan *Local Mean Vector* agar menyelesaikan masalah pada ketetanggaan yang kemungkinan ada beberapa point *outlier* dan menggunakan *Harmonic Mean*. Diharapkan dengan menggabungkan *Local Mean Vector* dan *Harmonic Mean* dapat menyisakan tetangga yang terbaik untuk pengambilan keputusan terakhir. Kinerja akurasi dari metode yang sudah ada yaitu KNN melakukan penentuan class dengan Majority Voting dibandingkan dengan metode yang kami usulkan untuk mengetahui apakah metode yang kami usulkan dapat meningkatkan kinerja klasifikasi.

2. TINJAUAN PUSTAKA

2.1 Preprocessing

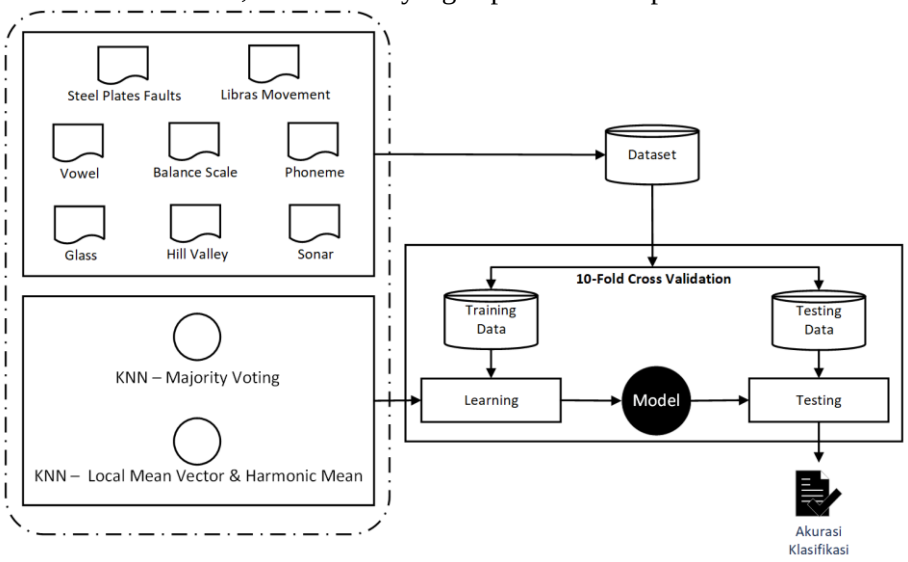
Preprocessing data merupakan hal yang perlu dilakukan dalam proses data *mining*. Hal ini dikarenakan tidak semua data atau atribut data pada data tersebut digunakan dalam proses data *mining*. Proses ini digunakan agar data yang digunakan siap digunakan pada proses klasifikasi [14], [15]. Proses yang dilakukan adalah normalisasi data yang bertujuan untuk membuat skala nilai atribut data sehingga berada dalam rentang tertentu [16], [17]. Teknik yang digunakan pada tahap ini adalah *min max scaler* dan *one hot encoding*. *Min-max normalization* atau *min-max scaler* merupakan metode normalisasi yang melakukan transformasi linier pada data asli agar menghasilkan keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses [18]. Sedangkan *one hot encoding* adalah skema pengkodean untuk membandingkan setiap tingkat variabel kategori ke tingkat referensi tetap. *One hot encoding* mengubah satu variable dengan *n* observasi dan *d* nilai berbeda, menjadi *d* variabel biner dengan *n* observasi masing-masing [19].

2.2 K-Fold Cross Validation

K-Fold Cross Validation juga mencakup teknik validasi untuk membagi data menjadi k bagian dan kemudian setiap bagian dimasukkan dalam proses klasifikasi. Dimana dengan menggunakan *K-Fold Cross Validation* akan diuji sebanyak k. Setiap pengujian menggunakan satu data uji dan k-1 bagian akan menjadi data latih, kemudian data uji tersebut akan ditukar dengan satu data latih sehingga untuk setiap pengujian akan diperoleh data uji yang berbeda [7].

3. METODE PENELITIAN

Alur dari penelitian untuk Penyeleksian Tetangga Berdasarkan *Harmonic Mean* dan *Local Mean Vector* pada algoritma KNN atau disingkat LMVHM secara sistematis terdiri dari pengumpulan data, *preprocessing, k fold cross validation*, proses klasifikasi KNN *Majority Voting* atau bisa disebut KNN MV dan KNN dengan Penyeleksian Tetangga Berdasarkan *Local Mean Vector* dan *Harmonic Mean* atau bisa disebut KNN LMVHM, dan evaluasi yang di presentasikan pada Gambar 1.



Gambar 1. Metode penelitian.

Pada gambar dapat dilihat terdapat delapan dataset yang digunakan pada penelitian ini. Setiap dataset akan dibagi menjadi data training dan testing dengan cara 10-fold cross validation. Data training dari sebuah dataset yang dipilih tersebut akan digunakan untuk membuat model dengan algoritma KNN – Majority Voting. Kemudian model diuji dengan menggunakan data testing. Proses ini dilakukan sebanyak 10 kali karena menggunakan 10-fold cross validation. Hasil prediksi yang dikumpulkan selanjutnya digunakan untuk menghitung akurasi klasifikasi. selanjutnya dataset yang sama digunakan untuk membuat model dengan algoritma KNN – Local Mean Vector dan Harmonic Mean dan akhirnya akan dihitung akurasi klasifikasinya. Proses ini diulang sampai seluruh dataset yang ada.

3.1 Pengumpulan Data

Data yang digunakan pada riset ini didapat dari UCI *Machine Learning Repositories*. Dipilih dataset-dataset dengan berbagai kasus yaitu data tidak seimbang, data seimbang, data dengan *multi class*, dan data *binary class* seperti yang dapat dilihat pada Tabel 1.

Tabel 1.Dataset.

Dataset	Atribut	Jumlah Data	Tipe Data	Jenis Data
Glass	11	214	*Real*	*Multi class*, tidak seimbang
Hill Valley	101	1212	*Real*	*Binary Class*, seimbang
Sonar	60	208	*Real*	*Binary Class*, seimbang
Vowel	10	528	*Real*	*Multi class*, seimbang
Balance Scale	4	625	*Categorical*	*Multi class*, tidak seimbang
Phoneme	5	5404	*Real*	*Binary Class*, tidak seimbang

Steel Plates Faults	27	1941	*Real*	*Multi class*, tidak seimbang
Libras Movement	91	360	*Real*	*Multi class*, seimbang

3.2 Preprocessing

Setelah data dikumpulkan, setiap data akan dilakukan preprocessing. Tahapan ini menggunakan metode Min Max Scaler untuk fitur berupa angka, sedangkan One Hot Encoding untuk fitur berupa kategori.

3.3 K Fold Cross Validation

Setelah data dilakukan *preprocessing*, selanjutnya data akan dibagi menggunakan *K-Fold Cross Validation.*Untuk *K-Fold Cross Validation* akan menggunakan 10-Fold, sehingga data akan terbagi jadi 10% data uji dan 90% data latih untuk setiap fold nya.

3.4 Proses Klasifikasi

Pada tahap ini, data yang sudah di lakukan *preprocessing* akan dilakukan klasifikasi dengan algoritma KNN, pada tahapan ini memiliki dua skenario yang akan dilakukan. Pada skenario pertama dilakukan klasifikasi KNN konvensional yaitu dengan pengambilan keputusan *Majority Voting*. Kemudian pada skenario kedua akan dilakukan klasifikasi KNN dengan seleksi tetangga berdasarkan *Local Mean Vector* dan *Harmonic Mean*. Proses pelatihan KNN menghasilkan *k* yang memberikan akurasi tertinggi dalam mengeneralisasi data yang akan datang. Masalahnya, sampai saat ini *k* tidak dapat ditentukan secara matematis. Jadi, proses pelatihan pada dasarnya adalah mengamati sejumlah *k* sampai dihasilkan *k* yang paling optimal [20].

3.5 Evaluasi

Tahapan selanjutnya melakukan evaluasi dengan menghitung akurasi, presisi, recall dan f1 score menggunakan *Confusion Matrix* [21].

3.6 Perbandingan Hasil

Tahap akhir adalah membandingkan hasil berupa rata rata akurasi, presisi, *recall* dan *f1 score* dari dua skenario yang sudah dilakukan agar bisa melihat perbandingan kinerja antara metode KNN MV dan KNN LMVHM, Untuk dapat mendapatkan hasil akhir, digunakan rata rata dari kinerja klasifikasi dari semua dataset [22].

4. PEMBAHASAN

4.1 *Preprocessing*

Proses *One Hot Encoding* dilakukan pada dataset *Balance Scale*, kemudian *One Hot Encoding* pada data akan terjadi perubahan jumlah fitur tergantung jumlah fitur sebelumnya dan data tersebut. Seperti misalkan di salah satu fitur memiliki data dengan jumlah kategori 5, maka fitur tersebut akan terbagi lagi menjadi 5. Perubahan tersebut bisa dilihat pada Tabel 2.

Tabel 2. Tiga sampel dari dataset *Balance Scale* sebelum *OneHot*

Left-W	*Left*-D	*Right*-W	*Right*-D
1	2	3	4
5	1	3	4
5	2	4	1

Yang awalnya fitur *Left*-W akan terbagi jadi 5 fitur lagi, dikarenakan fitur *Left*-W memiliki 5 kategori maka akan dibagi lagi menjadi 5 fitur dengan nama sebagai contoh LW1 sampai LW5. Setelah itu, jika pada *record* disebut kalau *Left*-W bernilai kategori 1 maka LW1 akan bernilai 1 dan LW lainnya akan bernilai 0, atau bisa disebut jika fitur Y pada *record* bernilai kategori X maka YX akan bernilai 1 dan fitur Y lainnya akan bernilai 0 bisa dilihat pada Tabel 3.

Tabel 3. Tiga contoh dari dataset *Balance Scale* setelah *OneHot*

LW1	..	LW5	..	LD1	LD2	..	LD5	RW1	..	RW3	RW4	RW5	RD1	..	RD4	RD5
1	..	0	..	0	1	..	0	0	..	1	0	0	0	..	1	0
0	..	1	..	1	0	..	0	0	..	1	0	0	0	..	1	0
0	..	1	..	0	1	..	0	0	..	0	1	0	1	..	0	0

Sedangkan untuk tahapan *Min Max Scaler* akan mengubah semua dataset kecuali *Min Max Scaler*, setiap fitur dari dataset akan diubah rentang nilainya menjadi 0 sampai 1 agar setiap fitur dianggap sama. Salah satu contoh yaitu pada dataset *Glass* memiliki nilai yang negatif, namun setelah dilakukan *Min Max Scaler* tidak lagi bernilai negatif dikarenakan seluruh datanya akan diubah rentang menjadi 0 sampai 1.

4.2 K Fold Cross Validation

Data yang telah di *preprocessing* akan dilakukan *K Fold Cross Validation* dengan 10 Fold. Salah satu contoh dari *K Fold Cross Validation* pada dataset Glass dengan jumlah data sebesar 214 dengan pembagian data uji setiap fold nya antara 192 atau 193, dengan data latih setiap fold nya antara 22 atau 21. Proses *K Fold Cross Validation* sendiri akan membagi dataset menjadi beberapa bagian yang ditentukan dari berapa *Fold* yang digunakan.

4.3 Perbandingan Hasil

Pada penelitian ini akan dilakukan yaitu perbandingan pengambilan keputusan pada algoritma klasifikasi KNN. Pengambilan keputusan yang dibandingkan yaitu *Majority Voting* dengan penyeleksian tetangga berdasarkan *Local Mean Vector* dan *Harmonic Mean*.

Tabel 4. Akurasi dari tiap dataset yang dihasilkan

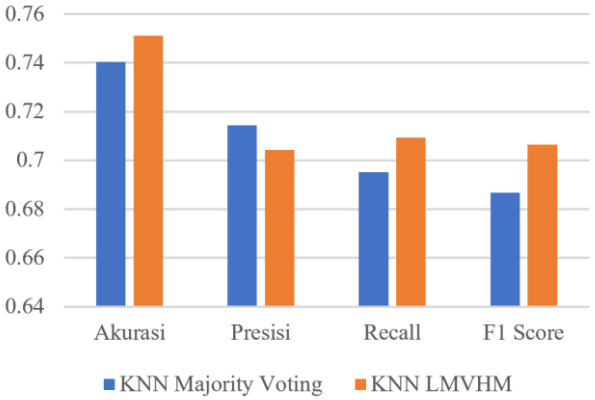
Dataset	KNN MV	KNN LMVHM	Jenis Dataset
Glass	0,897196262	0,892523364	*Multi class*, tidak seimbang
Hill Valley	0,588283828	0,648514851	*Binary Class*, seimbang
Sonar	0,6875	0,6875	*Binary Class*, seimbang
Vowel	0,850378788	0,90719697	*Multi class*, seimbang
Balance Scale	0,616	0,64	*Multi class*, tidak seimbang
Phoneme	0,889712697	0,90231696	*Binary Class*, tidak seimbang
Libras Movement	0,727272727	0,793939394	*Multi class*, tidak seimbang
Steel Plates Fault	0,767645544	0,774343122	*Multi class*, seimbang

Terlihat bahwa akurasi yang didapatkan dengan jenis yang sama memiliki akurasi yang kurang lebih besarnya, seperti pada jenis data *Binary Class* dan Seimbang yaitu dataset Sonar mendapatkan 0.687 untuk KNN MV dan KNN LMVHM, dataset Hill Valley mendapatkan 0,588 untuk KNN MV dan 0,648 untuk KNN LMVHM. Namun ada satu jenis dengan memiliki kejanggalan yaitu data *Multi class* dan tidak seimbang dengan dataset yang terdiri dari Glass, Balance Scale, Phoneme dan Libras Movement. Kejanggalan disini terjadi pada dataset Balance Scale dikarenakan mendapatkan akurasi yang terbilang rendah dengan akurasi dibawah 0,7 atau lebih tepatnya 0,616 untuk KNN MV dan 0,64 untuk KNN LMVHM. Sedangkan pada dataset lain yang sejenis mendapatkan akurasi yang cukup tinggi dengan akurasi diatas 0,7. Balance Scale diambil dari situs UCI *Machine Learning Repository* dengan ditulis bahwa dataset Balance Scale ini bersifat kategori, namun setelah dilihat kembali fitur dari dataset tersebut bernama *Left*-Weight, *Left*-Distance, *Right*-Weight, dan *Right*-Distance. Sehingga balance scale ini bisa dibilang memiliki 2 sifat yaitu bisa menjadi kategori atau bisa jadi sebagai numerik, dikarenakan fitur fitur dataset tersebut adalah angka meskipun memiliki rentan yang pasti dari 1 sampai 5. Hal ini bisa menjadi alasan kenapa Balance Scale memiliki akurasi yang lebih rendah dari data lain yang sejenis, dikarenakan pada penelitian ini melakukan klasifikasi pada data Balance Scale sebagai data kategori. Kemudian berdasarkan hasil yang didapatkan, akan dibagi lagi hasilnya menjadi beberapa kategori.

Kategori pertama adalah kategori data tidak seimbang, yang dimana diambil rata rata hasil dari setiap data yang tidak seimbang dengan hasil yang didapatkan seperti pada Gambar 2.

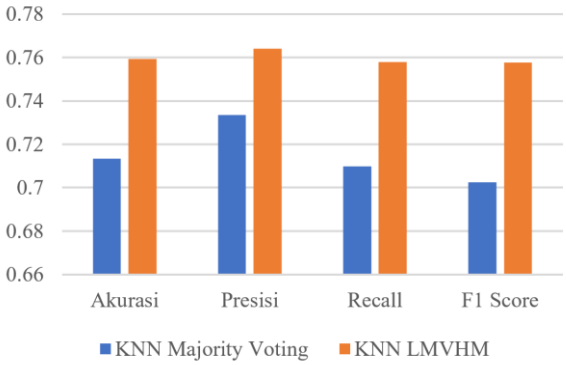
p-ISSN: 2460-173X
e-ISSN: 2598-5841

Berdasarkan hasil yang didapatkan KNN LMVHM bekerja lebih baik dibanding dengan KNN MV untuk data tidak seimbang kecuali pada presisi.



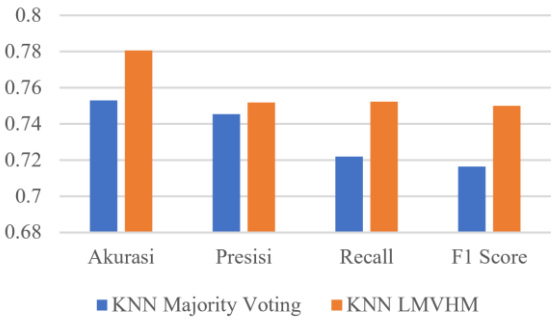
Gambar 2. Hasil yang didapatkan dari data tidak seimbang

Kategori kedua adalah kategori data seimbang, yang dimana diambil rata rata hasil dari setiap data yang seimbang yang bisa dilihat pada Gambar 3. Berdasarkan hasil yang didapatkan KNN LMVHM bekerja lebih baik dibanding dengan KNN MV untuk data seimbang di semua aspek dari segi akurasi, presisi, recall dan f1 score.



Gambar 3. Hasil yang didapatkan dari data seimbang

Kategori terakhir adalah kategori secara keseluruhan, akan diambil rata rata hasil dari keseluruhan data yang telah diuji yang bisa dilihat pada Gambar 4. Berdasarkan hasil yang didapatkan KNN LMVHM bekerja lebih baik dibanding KNN MV pada keseluruhan di semua aspek dari segi akurasi, presisi, recall dan f1 score.



Gambar 4. Hasil yang didapatkan dari data keseluruhan

5. KESIMPULAN

Penyeleksian tetangga menggunakan *Local Mean Vector* dan *Harmonic Mean* pada algoritma klasifikasi KNN (KNN LMVHM) menghasil kinerja akurasi yang lebih bagus secara keseluruhan dibanding KNN menggunakan *Majority Voting*, hal ini dikarenakan pada KNN LMVHM menyisakan satu tetangga yang mayoritas adalah tetangga yang sama dengan k = 1 pada algoritma KNN, hal ini adalah salah satu kelebihan dan kekurangan. Kelebihannya adalah akurasi yang didapatkan lebih baik dikarenakan banyak tetangga yang tersisa adalah tetangga yang sama saat k = 1 pada KNN dengan persentase tetangga nya sekitar 100% hingga 40%, hal ini juga bisa dipengaruhi oleh jumlah dari k yang dipilih, sedangkan kelemahannya sendiri adalah metode yang diusulkan masih kurang bisa mendapatkan alternatif yang lebih baik. Hal ini kemungkinan terjadi dikarenakan metode yang diusulkan mengabaikan data di sekitar tetangga terluar, karena informasi mengenai apakah tetangga tersebut adalah termasuk data anomali yang didapatkan dari algoritma KNN. Sehingga untuk penelitian selanjutnya dilakukan eksplorasi agar bisa memperluas informasi apakah tetangga tersebut adalah data anomali atau tidak, bisa dengan cara menambahkan beberapa data sekitar tetangga terluar atau tepi untuk termasuk dalam perhitungan *Local Mean Vector* dan *Harmonic Mean* dengan harapan tetangga yang tersisa adalah benar benar alternatif yang lebih baik dibanding tetangga pada saat k = 1. Selain itu dilakukan juga untuk menggunakan metode algoritma klasifikasi KNN yang lain dengan membentuk kumpulan data yang berbeda dengan KNN seperti *Radius Nearest Neighbour* dan AdaBoost *Classifier*.

DAFTAR PUSTAKA

[1] R. R. Pratama, "Analisis Model Machine Learning Terhadap Pengenalan Aktifitas Manusia," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 2, pp. 302–311, 2020, doi: 10.30812/matrik.v19i2.688.

[2] I. Budiman, D. T. Nugrahadi, and R. A. Nugroho, "Implementasi Algoritma K-Nearest Neighbour untuk Prediksi Waktu Kelulusan Mahasiswa," *Pros. SNRT (Seminar Nas. Ris. Ter.)*, vol. 5662, pp. 9–10, 2016.

[3] W. Hardianti, F. Indriani, and R. A. Nugroho, "Analisis Perbandingan Algoritma Distance-Weighted KNN dan Algoritma KNN pada Prediksi Masa Studi Mahasiswa," *Semin. Nas. Ilmu Komput.*, vol. 1, pp. 108–117, 2017.

[4] R. A. Supono and M. A. Suprayogi, "Perbandingan Metode TF-ABS dan TF-IDF Pada Klasifikasi Teks Helpdesk Menggunakan K-Nearest Neighbor," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 5, pp. 911–918, 2021, doi: 10.29207/resti.v5i5.3403.

[5] E. Laksono, A. Basuki, and F. Bachtiar, "Optimization of K Value in KNN Algorithm for Spam and Ham Email Classification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 2, pp. 377–383, 2020, doi: 10.29207/resti.v4i2.1845.

[6] R. S. D. Wijaya, Adiwijaya, Andriyan B Suksmono, and Tati LR Mengko, "Segmentasi Citra Kanker Serviks Menggunakan Markov Random Field dan Algoritma K-Means," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 139–147, 2021, doi: 10.29207/resti.v5i1.2816.

[7] A. Bode, "K-Nearest Neighbor Dengan Feature Selection Menggunakan Backward Elimination Untuk Prediksi Harga Komoditi Kopi Arabika," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 188–195, 2017, doi: 10.33096/ilkom.v9i2.139.188-195.

[8] M. Rivki and A. M. Bachtiar, "Implementasi algoritma K-Nearest Neighbor dalam pengklasifikasian follower twitter yang menggunakan Bahasa Indonesia," *J. Sist. Inf.*, vol. 13, no. 1, pp. 31–37, 2017.

[9] S. Mehta, X. Shen, J. Gou, and D. Niu, "A new nearest centroid neighbor classifier based on k local means using harmonic mean distance," *Inf.*, vol. 9, no. 9, 2018, doi: 10.3390/info9090234.

[10] J. Gou, T. Xiong, and Y. Kuang, "A novel weighted voting for K-nearest neighbor rule," *J. Comput.*, vol. 6, no. 5, pp. 833–840, 2011, doi: 10.4304/jcp.6.5.833-840.

[11] K. U. Syaliman, E. B. Nababan, and O. S. Sitompul, "Improving the accuracy of k-nearest neighbor using local mean based and distance weight," *J. Phys. Conf. Ser.*, vol. 978, no. 1, 2018, doi: 10.1088/1742-6596/978/1/012047.
[12] Y. Mitani and Y. Hamamoto, "A local mean-based nonparametric classifier," *Pattern Recognit. Lett.*, vol. 27, no. 10, pp. 1151–1159, 2006, doi: 10.1016/j.patrec.2005.12.016.
[13] C. Kasemtaweechok and W. Suwannik, "Prototype selection for k-nearest neighbors classification using geometric median," *ACM Int. Conf. Proceeding Ser.*, pp. 140–144, 2016, doi: 10.1145/3033288.3033301.
[14] R. R. Rerung, "Penerapan Data Mining dengan Memanfaatkan Metode Association Rule untuk Promosi Produk," *J. Teknol. Rekayasa*, vol. 3, no. 1, p. 89, 2018, doi: 10.31544/jtera.v3.i1.2018.89-98.
[15] M. K. Delimayanti, R. Sari, M. Laya, M. R. Faisal, Pahrul, and R. F. Naryanto, "The effect of pre-processing on the classification of twitter's flood disaster messages using support vector machine algorithm," *Proc. ICAE 2020 - 3rd Int. Conf. Appl. Eng.*, no. February 2021, 2020, doi: 10.1109/ICAE50557.2020.9350387.
[16] Wahyudi, M. R. Faisal, D. Kartini, I. Budiman, and A. Farmadi, "Effect of Normalization of Genre Music Data on Classification Performance with Random Forest," *J. Data Sci. Softw. Eng.*, vol. 02, no. 01, pp. 56–63, 2021.
[17] W. A. Ningsih, F. Indriani, and A. Farmadi, "Klasifikasi Detak Jantung Janin dengan Learning Vector Quantization (LVQ)," in *Seminar Nasional Ilmu Komputer (SOLITER)*, 2019, pp. 130–135.
[18] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, pp. 78-82, 2019, doi: 10.24114/cess.v4i1.11458.
[19] K. Potdar, T. S. Pardawala, and C. D. Pai, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, pp. 7– 9, 2017, doi: 10.5120/ijca2017915495.
[20] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: INFORMATIKA, 2018.
[21] M. N. Nasir, I. Budiman, and A. Farmadi, "Perbandingan Pengaruh Nilai Centroid Awal pada Algoritma K-Means dan K-Means++ terhadap Hasil Cluster Menggunakan Metode Confusion Matrix," in *Seminar Nasional Ilmu Komputer (SOLITER)*, 2017, pp. 118–127.
[22] J. Li, S. Fong, Y. Sung, K. Cho, R. Wong, and K. K. L. Wong, "Adaptive swarm cluster-based dynamic multi-objective synthetic minority oversampling technique algorithm for tackling binary imbalanced datasets in biomedical data classification," *BioData Min.*, vol. 9, no. 1, pp. 1–15, 2016, doi: 10.1186/s13040-016-0117-1.

Biodata Penulis

Muhammad Al Ichsan Nur Rizqi Said, lahir di Kota Banjarbaru pada tahun 2000. Sejak kecil penulis pertama menempuh pendidikan di sekolah negeri Kota Banjarbaru. Pada tahun 2021, penulis pertama memperoleh gelar S. Kom. di Universitas Lambung Mangkurat dengan Program Studi Ilmu Komputer. Saat ini penulis pertama aktif mendalami bidang data science dan software engineer.

Mohammad Reza Faisal, lahir di Banjarmasin pada tahun 1976, Kalimantan Selatan. Penulis kedua merupakan alumni Teknik Informatika dan Fisika di Institut Teknologi Bandung dengan pendidikan terakhir saat ini S3 Bioinformatika di Kanazawa University. Saat ini penulis kedua aktif menjadi dosen di Fakultas MIPA Universitas Lambung Mangkurat.

Dwi Kartini, lahir di Jakarta dan menyelesaikan pendidikan S1 di Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yaitu UPI "YPTK" Padang. Lulus S2 Ilmu Komputer tahun 2011 kemudian kembali ke Banjarbaru dan tahun 2013 menjadi dosen di Ilmu Komputer Universitas Lambung Mangkurat sampai sekarang.

Irwan Budiman, lahir di Banjarmasin, Kalimantan Selatan. Penulis keempat merupakan alumni Teknik Informatika dan Fisika di Institut Teknologi Bandung dengan pendidikan terakhir saat ini S2 Sistem Informasi di Universitas Diponegoro Semarang. Saat ini penulis keempat aktif menjadi koordinator dan sekaligus dosen di Program Studi S1 Ilmu Komputer FMIPA Universitas Lambung Mangkurat, serta menjadi asesor di BNSP.

Triando Hamonangan Saragih, lahir di Banjarmasin, menyelesaikan pendidikan S1 dan S2 di Universitas Brawijaya Malang, saat ini sedang mendalami riset di bidang *Machine Learning* dan menjadi dosen di Ilmu Komputer Universitas Lambung Mangkurat.