# Perbandingan Performa antara Imputasi Metode Konvensional dan Imputasi dengan Algoritma *Mutual Nearest Neighbor*

Azwar Rizal Alfarisi, Handayani Tjandrasa, dan Isye Arieshanti
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
*E-mail*: handatj@its.ac.id

***Abstrak*—*Missing value* adalah sebuah permasalahan yang sering terjadi pada *dataset* riil. Kekurangan ini biasanya mempengaruhi akurasi saat dilakukan klasifikasi dengan menggunakan *dataset* tersebut. Salah satu cara menyelesaikan masalah *missing value* tersebut adalah mengisi nilai baru atau dikenal dengan metode imputasi. Algoritma *mutual nearest neighbor* (MNN) adalah sebuah algoritma pengenalan pola yang menggunakan tetangga mutual terdekat suatu *instance*. Dalam studi ini, algoritma MNN digunakan sebagai metode imputasi. Performanya akan dibandingkan dengan metode imputasi konvensional yaitu mengisikan nilai *mean* atau *modus* data atribut ke *missing value*. Berdasarkan hasil uji coba, performa klasifikasi setelah dilakukan imputasi dengan algoritma MNN mengungguli performa klasifikasi dengan metode imputasi konvensional.**

***Kata Kunci*—imputasi, klasifikasi, *missing value*, MNN.**

## I. PENDAHULUAN

SALAH satu hal yang dapat mengurangi akurasi klasifikasi adalah adanya *missing value* dalam *dataset* yang digunakan. Dalam dunia nyata, banyak faktor yang menyebabkan adanya *missing value*. Contohnya, pada diagnosis medis beberapa tes medis tidak bisa dilakukan karena keterbatasan peralatan atau tes medis tidak bisa dilakukan terhadap pasien tertentu.

Ada beberapa pendekatan untuk mengatasi masalah *missing value*. Seperti menghilangkan *instance* yang mempunyai *missing value*, mengabaikan *missing value* saat analisis atau mengganti *missing value* dengan nilai yang didapatkan dari suatu metode atau yang lebih dikenal dengan imputasi [1].

Metode imputasi yang paling umum digunakan adalah mengganti *missing value* dengan nilai kecenderungan pusat atributnya. Yaitu nilai *mean* untuk tipe data atribut kontinu dan nilai *modus* untuk tipe data atribut kategorikal [1]. Metode ini selanjutnya akan disebut metode imputasi konvensional.

Algoritma *mutual nearest neighbor* (MNN) adalah sebuah algoritma klasifikasi hasil pengembangan dari algoritma *k-nearest neighbor* (KNN). Berbeda dengan algoritma KNN yang menggunakan *k* tetangga terdekatnya untuk memprediksi label kelas baru suatu *instance*, algoritma MNN menggunakan tetangga mutual terdekat dari suatu *instance* untuk mendapatkan label kelas *instance* tersebut [2].

Selain untuk klasifikasi, algoritma MNN juga bisa digunakan untuk mengidentifikasi *outlier*. Dalam mengidentifikasi *outlier* algoritma MNN menganggap *instance* yang tidak mempunyai tetangga mutual adalah suatu *outlier* [3].

Tujuan dari studi ini adalah menggunakan algoritma MNN sebagai metode imputasi. Hasilnya akan dibandingkan dengan metode imputasi konvensional. Performa yang dibandingkan adalah akurasi klasifikasi *dataset* yang telah mengalami imputasi dengan kedua metode tersebut.

Artikel ini tersusun sebagai berikut. Latar belakang dan tujuan studi dijelaskan pada Bagian I. Metodologi dijelaskan pada Bagian II. Perancangan data dijelaskan pada Bagian III. Selanjutnya pada Bagian IV dijelaskan mengenai hasil uji coba dan analisisnya. Terakhir, kesimpulan dari studi ini dijelaskan pada Bagian V.

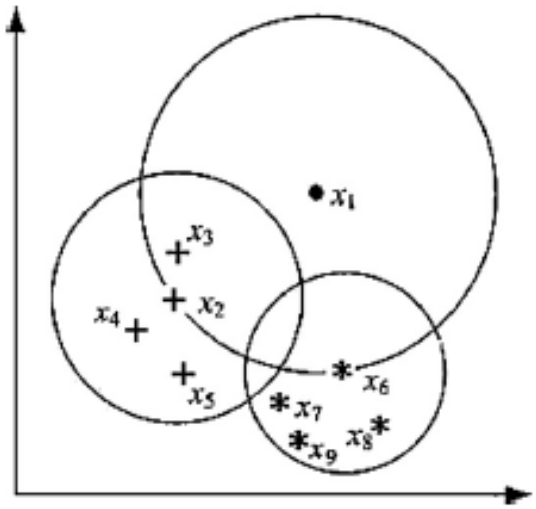
## II. METODOLOGI

### A. *Missing Value*

*Missing value* adalah keadaan dimana beberapa nilai atribut dalam *dataset* kosong (tidak ada nilainya). Ada beberapa teknik untuk menangani masalah ini. Namun dalam artikel ini hanya akan membahas metode imputasi sebagai solusi permasalahan [1].

Imputasi adalah proses yang digunakan untuk menentukan dan menetapkan nilai pengganti untuk *missing value*. Metode imputasi menjadi penting dalam situasi dimana *dataset* lengkap dibutuhkan untuk analisis. Berikut ini adalah beberapa metode imputasi [1]:

- Imputasi secara manual
  Secara umum, metode ini memakan waktu dan hampir tidak mungkin dilakukan apabila *dataset*-nya berukuran besar dengan banyak *missing value*.
- Imputasi dengan konstanta global
  Metode ini mengganti semua *missing value* dengan konstanta tertentu yang sama. Jika menggunakan metode ini, algoritma *mining* akan salah menduga bahwa data-data membentuk suatu konsep yang menarik karena memiliki nilai yang sama. Sehingga walaupun metode ini sederhana, tetapi hasilnya tidak bagus.
- Imputasi dengan metode konvensional
  Metode ini mengganti semua *missing value* dengan nilai ukuran kecenderungan pusat atribut masing-masing *missing value*. Yaitu nilai *mean* untuk tipe data atribut kontinu dan




Gambar. 1. Ilustrasi algoritma *mutual nearest neighbor*.

nilai *modus* untuk tipe data atribut kategorikal.

- Imputasi dengan suatu model prediksi
  Metode ini menggunakan model prediksi untuk mencari nilai pengganti untuk *missing value*. Contohnya adalah model regresi, induksi pohon keputusan dan *bayesian inference*.

### *B. Algoritma Mutual Nearest Neighbor*

Algoritma *mutual nearest neighbor* adalah algoritma klasifikasi yang menggunakan tetangga mutual terdekat dari suatu *instance* untuk memprediksi label kelas *instance* tersebut [2]. Algoritma ini merupakan pengembangan dari algoritma KNN. Ilustrasi algoritma MNN dapat dilihat di Gambar 1. Dalam gambar tersebut, *instance* $x_1$ mempunyai tiga tetangga terdekat yaitu $x_2$, $x_3$, $x_6$. Namun, tiga tetangga terdekat dari *instance-instance* $x_2$, $x_3$, $x_6$ tidak meliputi $x_1$. Sehingga, $x_1$ dianggap tidak mempunyai tetangga mutual dan dianggap sebagai *outlier*.

Selain digunakan untuk pengklasifikasi, algoritma MNN juga bisa digunakan untuk mendeteksi *outlier* untuk dihapus [3] dan imputasi *missing value*.

### *C. Algoritma Mutual Nearest Neighbor untuk Imputasi*

Metode ini menggunakan gabungan antara algoritma MNN dan metode imputasi konvensional. Prosedur langkah pendekatan ini dijelaskan sebagai berikut:

1. Masukkan nilai *k*.
2. Jika ditemukan *missing value* pada *instance x*, cari *k* tetangga mutual dari *instance x* dengan cara:
   a) Hitung jarak antara *instance x* dengan *instance* lain.
   b) Urutkan hasilnya dari yang mempunyai jarak paling kecil.
   c) Dari *k* tetangga terdekat yang didapat, misal *y(x)*, cari *k* tetangga terdekat dari masing-masing *instance* dalam *y(x)*. Jika *x* adalah salah satu *k* tetangga terdekat dari *instance* tersebut, masukkan *instance* ke dalam himpunan tetangga mutual *x*, misal *M(x)*.
3. Hitung *mean* nilai data atribut dimana *missing value* berada untuk data kontinu dan *modus* untuk data kategorikal dari *instance-instance* dalam *M(x)*.
4. Masukkan hasil yang didapat dari langkah 2 dan 3 ke nilai *missing value*. Jika *instance x* tidak punya tetangga mutual maka imputasikan hasil dari metode konvensional ke *missing value*.
5. Ulangi langkah 2 sampai langkah 4 sampai tidak ada *missing value*.
6. Untuk menghitung jarak antar *instance* digunakan persamaan jarak *euclidean* untuk *dataset* dengan semua tipe data atribut kontinu, persamaan jarak *jaccard* untuk *dataset* dengan semua data atribut kategorikal dan *heterogeneous euclidean overlap metric* (HEOM) untuk *dataset* yang atributnya mempunyai tipe data heterogen [4]. Perhitungan jarak antar *instance* hanya dilakukan pada nilai atribut yang tidak mengalami missing *value*. Berikut adalah persamaan-persamaan untuk menghitung jarak ketiga metode di atas.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (1)$$

Persamaan (1) merupakan persamaan jarak *euclidean* dimana *p* dan *q* adalah *instance* dan *n* adalah jumlah atribut.

$$J_d(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \quad (2)$$

Persamaan (2) merupakan persamaan jarak *jaccard* dimana *A* dan *B* adalah *instance*, $|A \cup B|$ adalah jumlah atribut dan $|A \cap B|$ adalah jumlah atribut yang nilai datanya sama.

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a^2(x,y)} \quad (3)$$

Persamaan (3) merupakan persamaan jarak *HEOM* dimana *m* adalah jumlah atribut dan fungsi $d_a(x, y)$ mengembalikan nilai jarak *instance x* dan *y* pada atribut ke-*a* dengan ketentuan sebagai berikut.

Keterangan:

$$d_a(x,y) = \begin{cases} |x-y| / \max_a - \min_a & ,\text{jika atribut a kontinu} \\ overlap(x,y) & ,\text{jika atribut a} \end{cases}$$

dimana $overlap(x, y) = 0$ jika $x = y$ dan $overlap(x, y) = 1$ jika sebaliknya.

## III. DATA

*Dataset* yang digunakan dalam artikel ini adalah *dataset wisconsin prognostic breast cancer (wpbc), large soybean database (soybean), 1984 United States Congressional voting records database (voting), credit approval (credit), cleveland heart disease database (cleveland)* dan *hepatitis domain (hepatitis)*. Keenam data tersebut dapat diperoleh di *UCI Machine Learning Repository* [5].

*Wpbc* adalah *dataset* yang terdiri dari 33 atribut dan 198 data. Data dikumpulkan oleh Dr. William H. Wolberg dari University of Wisconsin sejak 1984 melalui pasien kanker payudara yang dirawatnya. Dalam *dataset* ini terdapat empat *missing value* dan semua atributnya bertipe kontinu

*Soybean* adalah *dataset* yang terdiri dari 35 atribut dan 307 data. 6,63% data masukan merupakan *missing value*. Tujuan dari *dataset* ini adalah mengklasifikasi jenis penyakit yang diderita tumbuhan kacang kedelai. Semua atribut *dataset* ini bertipe kategorikal.

*Voting* adalah *dataset* yang meliputi 16 suara anggota dewan Amerika Serikat pada tahun 1984. *Dataset* ini mempunyai 435 data dengan *item* yang berupa *missing value* berjumlah 392. Semua atributnya bertipe biner.

*Credit approval* adalah *dataset* yang berisi aplikasi kartu kredit. *Dataset* ini dianggap menarik karena atributnya heterogen dan ada *missing value*. *Dataset* ini memiliki 690 data, 15 atribut dan 67 *item* merupakan *missing value*.

*Cleveland* adalah *dataset* yang dikumpulkan oleh Robert Detrano, M.D., Ph.D. dari V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. *Dataset* ini berisi data pasien yang menderita penyakit jantung. *Dataset* ini memiliki 303 data, 13 atribut dan enam *item* merupakan *missing value*.

*Hepatitis dataset* terdiri dari 155 pasien yang dideskripsikan dengan 19 atribut. Diantara pasien-pasien tersebut, 32 pasien meninggal karena hepatitis dengan sisanya bertahan hidup. Data masukan yang berupa *missing value* mencapai 5,67%.

## IV. UJI COBA

Untuk mendapatkan performa metode imputasi, *dataset* hasil imputasi diklasifikasikan dengan algoritma KNN, MNN, RIPPER, pohon keputusan C4.5, *naive bayes classifier* (NBC) dan *support vector machine* (SVM). Algoritma KNN dan MNN diuji coba dengan nilai *k = 3*, *5* dan *7*. Uji coba dilakukan dengan *k-fold cross validation* dengan nilai *k = 10*.

### A. Evaluasi Hasil Uji Coba

Fungsi yang digunakan untuk evaluasi hasil uji coba pada studi ini adalah fungsi akurasi. Akurasi didapatkan dengan membandingkan jumlah data yang terprediksi benar dengan jumlah semua data.

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

Persamaan (4) merupakan persamaan untuk mencari nilai akurasi evaluasi dimana *TP+TN* adalah jumlah data yang terprediksi benar dan *TP+FP+TN+FN* adalah jumlah semua data.

### B. Skenario Uji Coba

Skenario uji coba pada studi ini adalah membandingkan hasil akurasi antar *dataset* yang telah dilakukan imputasi dengan kedua metode. Uji coba dilakukan sebanyak 10 kali dan diambil nilai rata-rata akurasinya. Untuk imputasi dengan algoritma MNN nilai *k* yang digunakan untuk uji coba adalah *3*, *5* dan *7*. Dan hasil akurasi yang paling tinggi yang dibandingkan.

Tabel 1.
Hasil uji coba terhadap *dataset wpbc*

| Algoritma Klasifikasi | Algoritma Imputasi | |
|---|---|---|
| | Konvensional | Algoritma MNN |
| KNN *k = 3* | 74,14 | 74,14 |
| KNN *k = 5* | 77,53 | 77,53 |
| KNN *k = 7* | 78,33 | **78,38** |
| MNN *k = 3* | **73,96** | 73,86 |
| MNN *k = 5* | 74,51 | **74,77** |
| MNN *k = 7* | 76,01 | **76,18** |
| RIPPER | **74,24** | 74,04 |
| C4.5 | 71,92 | **72,42** |
| NBC | 66,57 | **66,57** |
| SVM | 76,16 | **76,26** |

Tabel 2.
Hasil uji coba terhadap *dataset soybean*

| Algoritma Klasifikasi | Algoritma Imputasi | |
|---|---|---|
| | Konvensional | Algoritma MNN |
| KNN *k = 3* | 88,14 | 88,14 |
| KNN *k = 5* | 86,58 | 86,58 |
| KNN *k = 7* | 84,72 | 84,72 |
| MNN *k = 3* | **93,10** | 93,09 |
| MNN *k = 5* | 91,73 | **91,76** |
| MNN *k = 7* | 89,62 | 89,62 |
| RIPPER | 86,16 | 86,16 |
| C4.5 | 88,40 | 88,40 |
| NBC | 91,79 | 91,79 |
| SVM | 91,47 | 91,47 |

### C. Hasil Uji Coba

Berikut adalah hasil pengujian sistem terhadap enam *dataset* berdasarkan skenario uji coba di atas. Tabel 1 menunjukkan hasil uji coba terhadap *dataset wpbc*, Tabel 2 menunjukkan hasil uji coba terhadap *dataset soybean*, Tabel 3 menunjukkan hasil uji coba terhadap *dataset voting*, Tabel 4 menunjukkan hasil uji coba terhadap *dataset credit*, Tabel 5 menunjukkan hasil uji coba terhadap *dataset cleveland* dan Tabel 6 menunjukkan hasil uji coba terhadap *dataset hepatitis.* Data tabel yang dicetak tebal berarti mempunyai akurasi yang lebih tinggi daripada pembandingnya.

Dari Tabel 1 diketahui bahwa nilai akurasi imputasi dengan algoritma MNN lebih besar daripada metode imputasi konvensional pada enam jenis algoritma klasifikasi dan pada dua algoritma klasifikasi yang lainnya mempunyai nilai yang lebih kecil.

Dari Tabel 2 diketahui bahwa nilai akurasi imputasi dengan algoritma MNN rata-rata sama dengan metode imputasi konvensional. Perbedaan hanya terjadi pada algoritma klasifikasi MNN. Sebenarnya hasil antara nilai 93,10 dengan 93,09 dan 91,73 dengan 91,76 tidak berbeda. Pada intinya, untuk *dataset soybean* tidak ada perbedaan hasil antara kedua metode imputasi.

Tabel 3.
Hasil uji coba terhadap *dataset voting*

| Algoritma Klasifikasi | Algoritma Imputasi | |
|---|---|---|
| | Konvensional | Algoritma MNN |
| KNN *k = 3* | 92,99 | **93,17** |
| KNN *k = 5* | 93,38 | **93,66** |
| KNN *k = 7* | 93,36 | **93,54** |
| MNN *k = 3* | 93,54 | **94,00** |
| MNN *k = 5* | 93,41 | **93,78** |
| MNN *k = 7* | 94,22 | **94,25** |
| RIPPER | **95,63** | 95,59 |
| C4.5 | 96,28 | 96,28 |
| NBC | **90,21** | 90,18 |
| SVM | 95,79 | 95,79 |

Tabel 4.
Hasil uji coba terhadap *dataset credit*

| Algoritma Klasifikasi | Algoritma Imputasi | |
|---|---|---|
| | Konvensional | Algoritma MNN |
| KNN *k = 3* | **85,70** | 85,65 |
| KNN *k = 5* | 86,19 | **86,32** |
| KNN *k = 7* | 86,43 | **86,48** |
| MNN *k = 3* | 83,29 | **83,33** |
| MNN *k = 5* | 85,55 | **85,63** |
| MNN *k = 7* | 85,94 | **85,96** |
| RIPPER | 85,39 | **85,55** |
| C4.5 | **85,57** | 85,43 |
| NBC | 77,80 | **77,84** |
| SVM | 84,88 | 84,88 |

Tabel 5.
Hasil uji coba terhadap *dataset cleveland*

| Algoritma Klasifikasi | Algoritma Imputasi | |
|---|---|---|
| | Konvensional | Algoritma MNN |
| KNN *k = 3* | 55,58 | **55,61** |
| KNN *k = 5* | 56,37 | **56,63** |
| KNN *k = 7* | 56,73 | **56,93** |
| MNN *k = 3* | **53,82** | 53,79 |
| MNN *k = 5* | 55,13 | **55,21** |
| MNN *k = 7* | 56,07 | **56,13** |
| RIPPER | **54,79** | 54,62 |
| C4.5 | 52,48 | **52,61** |
| NBC | 56,20 | 56,20 |
| SVM | 58,91 | **58,98** |

Dari Tabel 3 diketahui bahwa nilai akurasi imputasi dengan algoritma MNN lebih besar daripada metode imputasi konvensional pada enam jenis algoritma klasifikasi dan pada dua algoritma klasifikasi yang lainnya mempunyai nilai yang lebih kecil.

Dari Tabel 4 diketahui bahwa nilai akurasi imputasi dengan algoritma MNN lebih besar daripada metode imputasi konvensional pada tujuh jenis algoritma klasifikasi dan pada dua algoritma klasifikasi yang lainnya mempunyai nilai yang lebih kecil.

Dari Tabel 5 diketahui bahwa nilai akurasi imputasi dengan algoritma MNN lebih besar daripada metode imputasi konvensional pada hampir semua algoritma klasifikasi, kecuali algoritma RIPPER dan MNN dengan *k = 3*.

Tabel 6.
Hasil uji coba terhadap *dataset hepatitis*

| Algoritma Klasifikasi | Algoritma Imputasi | |
|---|---|---|
| | Konvensional | Algoritma MNN |
| KNN *k = 3* | **82,39** | 82,06 |
| KNN *k = 5* | 83,48 | 83,48 |
| KNN *k = 7* | 83,81 | **84,65** |
| MNN *k = 3* | 83,19 | **83,56** |
| MNN *k = 5* | 83,22 | **83,37** |
| MNN *k = 7* | 82,12 | **83,08** |
| RIPPER | 78,19 | **81,87** |
| C4.5 | 77,42 | **78,84** |
| NBC | 84,00 | **84,58** |
| SVM | 85,61 | **86,00** |

Dari Tabel 6 diketahui bahwa nilai akurasi imputasi dengan algoritma MNN lebih besar daripada metode imputasi konvensional pada hampir semua jenis algoritma dan mempunyai nilai lebih kecil hanya pada algoritma KNN dengan *k = 3*.

## V. KESIMPULAN

Imputasi dengan algoritma *mutual nearest neighbor* mempunyai performa yang lebih baik daripada imputasi dengan metode konvensional. Pada hasil uji coba terlihat rata-rata metode imputasi algoritma MNN mempunyai nilai akurasi yang lebih besar daripada metode imputasi konvensional. Hal ini terjadi pada uji coba terhadap *dataset wpbc*, *voting*, *credit*, *cleveland* dan *hepatitis*. Sementara itu, pada uji coba terhadap *dataset soybean* nilai rata-rata akurasi cenderung berimbang.

Imputasi algoritma MNN menghasilkan nilai akurasi yang lebih baik karena algoritma ini hanya menggunakan tetangga mutual *instance* yang mengalami *missing value* saja untuk mendapatkan nilai imputasi. Berbeda dengan metode konvensional yang menggunakan seluruh data yang kemungkinan berisi *instance* yang dianggap *outlier* oleh *instance* yang mengalami *missing value* tersebut.

## DAFTAR PUSTAKA

[1] P.N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining (4th ed.)*. Boston: Pearson Addison Wesley (2006).

[2] H. Liu, S. Zhang, J. Zhao, X. Zhao, and Y. Mo, "A New Classification Algorithm Using Mutual Nearest Neighbors," *Ninth International Conference on Grid and Cloud Computing* (2010) 52-57.

[3] H. Liu and S. Zhang, "Noisy Data Elimination Using Mutual K-Nearest Neighbor for Classification Mining", *The Journal of Systems and Software 85* (2012) 1067-1074.

[4] D.R. Wilson and T. Martinez, "Improved Heterogeneous Distance Functions", *Journal of Artifical Intelligence Research 6* (1997) 1-34.

[5] A. Frank and A. Asuncion. (2010). "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, Tersedia : http://www.archive.ics.uci.edu/ml.