Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202184912
Vol. 8, No. 4, Agustus 2021, hlm. 745-752 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

OPTIMASI BOBOT *K-MEANS* CLUSTERING UNTUK MENGATASI MISSING VALUE DENGAN MENGGUNAKAN ALGORITMA GENETIKA

Bain Khusnul Khotimah[*1], Muhammad Syarief[2], Miswanto[3], Herry Suprajitno[4]

[1,2]Jurusan Teknik Informatika, Universitas Trunojoyo Madura, Indonesia
[3,4]Departmen Matematika, Universitas Airlangga Surabaya, Indonesia
[1]bain@trunojoyoac.id; [2]mohammad.syarief@trunojoyo.ac.id; [3]miswanto@ fst.unair.ac.id;
[4]herry-s@fst.unair.ac.id
*Penulis Korespondensi

(Naskah masuk: 09 April 2021, diterima untuk diterbitkan: 19 Juli 2021)

Abstrak

Nilai yang hilang membutuhkan preprosesing dengan teknik imputasi untuk menghasilkan data yang lengkap. Proses imputasi membutuhkan initial bobot yang sesuai, karena data yang dihasilkan adalah data pengganti. Pemilihan nilai bobot yang optimal dan kesesuaian nilai *K* pada metode *K-Means* Imputation (KMI) merupakan masalah besar, sehingga menimbulkan error semakin meningkat. Model gabungan algoritma genetika (GA) dan KMI atau yang dikenal GAKMI digunakan untuk menentukan bobot optimal pada setiap *cluster* data yang mengandung nilai yang hilang. Algoritma genetika digunakan untuk memilih bobot dengan menggunakan pengkodean bilangan riel pada kromosom. Model hybrid GA dan KMI dengan pengelompokan menggunakan jumlah jarak *Euclidian* setiap titik data dari pusat clusternya. Pengukuran kinerja algoritma menggunakan fungsi kebugaran optimal dengan nilai MSE terkecil. Hasil percobaan data hepatitis menunjukkan bahwa GA efisien dalam menemukan nilai bobot awal optimal dari ruang pencarian yang besar. Hasil perhitungan menggunakan nilai MSE =0.044 pada K=3 dan replika ke-5 menunjukkan kinerja GAKMI menghasilkan tingkat kesalahan yang rendah untuk data hepatitis dengan atribut campuran. Hasil penelitian dengan menggunakan pengujian tingkat imputasi menunjukkan algoritma GAKMI menghasilkan nilai $r = 0.526$ lebih tinggi dibandingkan dengan metode lainnya. Penelitian ini menunjukkan GAKMI menghasilkan nilai r yang lebih tinggi dibandingkan metode imputasi lainnya sehingga dianggap paling baik dibandingkan teknik imputasi secara umum.

Kata kunci: *hybrid GA K-Means, optimasi bobot, missing value, MSE, imputasi data, K-Means Imputation*

OPTIMIZATION WEIGHT OF K-MEANS CLUSTERING TO OVERCOME MISSING VALUE USING GENETIC ALGORITHM

Abstract

Missing values require preprocessing techniques as imputation to produce complete data. Complete data imputation results require the appropriate initial weights, because the resulting data is replacement data. The choice of the optimal weighting value and the suitability of the network nodes in the K-Means Imputation (KMI) method are big problems, causing increasing errors. The combined model of Genetic Algorithm (GA) and KMI is used to determine the optimal weights for each data cluster containing missing values. Genetic algorithm is used to select weights by using real number coding on chromosomes. GA is applied to the KMI using clustering calculated using the sum of the Euclidean distances of each data point from the center of the cluster. Performance measurement algorithms using the fitness function optimally with the smallest MSE value. The results of the hepatitis data experiment show that GA is efficient in finding the optimal initial weight value from a large search space. The results of calculations using the MSE value = 0.04 for K = 3 and the 5th replication. So, GAKMI resulted in a low error rate for mixed data. The results of research using imputation level testing performed GAKMI produced r = 0.526 higher than the other methods. Thus, the higher the r value, the best for the imputation technique.

Keywords: *GA, K-Means, weight optimization, missing value, MSE, imputation*

1. PENDAHULUAN

Mekanisme imputasi (Acuna, 2004, Anwar, 2019). Mekanisme imputasi dilakukan dengan memprediksi nilai yang hilang dengan cara tertentu, yang hasilnya digunakan sebagai nilai pengganti (Han, 2010, Enders, 2014). *K-Means* untuk mengatasi

data yang hilang dikenal dengan *K-Means Imputation (KMI)* (Li, 2004, Mahboob, 2018). *KMI* adalah salah satu metode *clustering* untuk mengelompokkan data yang polanya belum diketahui. Proses imputasi KMI sangatlah menarik, ketika berhadapan dengan keragaman dan ketidakpastian data sehingga fitur pembelajaran membutuhkan data lengkap untuk menghasilkan kenerja yang baik (Zeebaree, 2017, Anwar, 2019).

K-Means secara umum memerlukan optimasi untuk menentukan pembaruan bobot baru, yang dapat meningkatkan efisiensi pembelajaran dan menyesuaikan nili *k*. *K-Means* memiliki kelemahan yaitu adanya ketidakkonsistenan initial *k*, bobot awal yang berbeda setiap iterasi, dan penggunaan parameter lainnya. *K-Means* ketika menentukan bobot awal acak seringkali menyebabkan dua kelemahan utama yaitu terjebak dalam minima lokal dan konvergen menjadi lebih lambat (Zeebaree, 2017). *K-Means* membutuhkan algoritma optimasi untuk memperbaiki kualitas parameter yang digunakan. Algoritma ini melibatkan fungsi dinamis untuk satu set parameter yang mengarah pada peningkatan kinerja *clustering* ((Farag, 2015a). Metode pengelompokan baru dipasangkan dengan pendekatan global untuk mengelola stabilitas lokal (Binu, 2015).

K-Means mengklasifikasikan data berdasarkan cluster titik tengah (centroid). Ketika centroid awal diinisialisasi secara acak dan terus diperbarui sampai mendapatkan cluster secara optimal. Proses inisialisasi centroid pada metode *K-Means* sangat mempengaruhi hasil cluster. GA diusulkan untuk mengatasi masalah sensitivitas dalam inisialisasi sentroid (AL Malki, 2016, Khotimah, 2016). Penelitian yang menunjukkan efektifitas GA dalam optimalisasi cluster menunjukkan bahwa algoritma genetika efektif dalam menangani data dengan atribut campuran pada jumlah cluster tertentu (Farag, 2015, Kaczmarowski, 2015, Chehouri, 2017). Sedangkan algoritma genetika dan algoritma *K-Means* and k-Medoids pada data beratribut campuran digunakan untuk memperoleh solusi optimum global sehingga hasil pengclusteran menjadi lebih baik dan akurat (Islam, 2020).

Permasalahan *K-Means Imputation* dalam memilih nilai bobot optimal merupakan masalah besar, karena nilai bobot sangat mempengaruhi hasil estimasi ketika proses imputasi. Penggunaan pola bobot yang konstan pada saat pengisian data, seringkali menjadi kesalahan besar dalam proses meminimalkan *cluster* karena nilai data yang kurang beragam (Maulik, 2000, Rahman, 2014, Anwar, 2019).

Penelitian ini mengembangkan model pencarian bobot optimal KMI pada kasus data yang hilang dengan menggunakan algoritma genetika. GA pada KMI dilatih untuk menggunakan algoritma genetika untuk menyesuaikan bobot dan bias di setiap lapisan. GA yang menggunakan analisis *clustering* untuk mengatur populasi dan memilih orang tua untuk rekombinasi. sehingga hasil akhir kinerja *clustering* dipakai untuk menghitung fungsi kebugaran. *Clustering* pada data yang hilang seringkali terjadi diberbagai bidang yaitu teknik dan disiplin ilmu seperti biologi, kedokteran, pembelajaran mesin, pengenalan pola, analisis citra dan industri.

Algoritma GA-KMI digunakan untuk mencari bobot pusat *cluster* dengan meminimalkan metrik pengelompokan. Pencarian bobot optimal pada KMI melalui algoritma genetika. Analisis *clustering* diterapkan untuk menyesuaikan probabilitas crossover Pc dan mutasi Pm dalam GA. Dengan menerapkan algoritma *K-Means*, populasi dikelompokkan dalam setiap generasi untuk menyesuaikan nilai-nilai operator genetik. Peraturan didasarkan pada mempertimbangkan ukuran relatif antara *cluster* masing-masing memegang kromosom terbaik dan terburuk. Kromosom terbaik adalah kromosom yang dihasilkan untuk menyelesaikan masalah pada berbagai banyak objektif variabel keputusan. Penelitian ini menggunakan studi kasus Hepatitis C terhadap 155 pasien dengan 19 fitur campuran kategori dan kontinu, yang telah diambil dari pembelajaran mesin gudang Universitas California. Masalah tingkat Bilirubin dalam hepatitis merupakan atribut kontinu, atribut yang diambil berdasarkan gejal merupakan atribut kategori. Dataset hepatitis membutuhkan operasi preprocessing dengan menggunakan transformasi *z-score* untuk memudahkan proses perhitungan dan cocok untuk data campuran (Marghny, 2011, Al Kindhi, 2019, Khotimah, 2020). Penelitian ini bertujuan untuk mengevaluasi pengaruh optimasi bobot pada K-Means dengan menggunakan algoritma genetika untuk memperbaiki data yang hilang pada pengelompokan data campuran.

2. METODE PENELITIAN

Proses hyrid imputasi memerlukan optimasi bobot karena data berupa imitasi yaitu luaran berasal dari imputasi, sehingga menghasilkan bias yang tinggi. Kami merangkum langkah-langkah dasar dari algoritma genetika dalam Algoritma 1. Dalam algoritma genetika, setiap individu (atau vektor solusi) dikodekan sebagai string bit biner atau vektor nilai riil, keduanya disebut sebagai kromosom. Representasi standar setiap individu adalah array biner bit, untuk memfasilitasi operasi crossover dan mutasi. Algoritma Optimasi *bobot K-Means* oleh GA jalannya sebagai berikut:

Algoritma : Fungsi f($\rightarrow$ x), $\rightarrow$ x = (x_1 , . . . , x_d) adalah *f* untuk seleksi bobot.

1. Mulai, t = 0.
2. Inisialisasi parameter *K-Means*.
3. Inisialisasi bobot pusat *cluster* baru sesuai jumlah *cluster* 1 …. k

4. Membangkitkan populasi awal kromosom bilangan riel M secara acak fungsi f(x)
5. Mengevaluasi fungsi kebugaran (fitness) pada masing-masing individu dengan menggunakan MSE
6. Memilih individu terbaik dari populasi P0 (t) dengan pemilihan roda rolet.
   a) Menerapkan Reproduksi crossover Pc dan mutasi P_m untuk menghasilkan individu baru.
   b) Ulangi sampai menghasilkan M individu.
   c) Lakukan operasi GA di sub-populasi SPi (t) untuk mendapatkan pi individu terbaik (t +1).
7. Ulangi untuk setiap p_i individu (t) untuk (i = 1, 2, ..., M).
8. Hentikan proses jika kriteria terpenuhi ketika proses berjalan hingga generasi tertentu dengan error<10%.
9. Menghasilkan populasi baru P_0 (t + 1) oleh individu pi (t + 1) dengan i = 1, 2, ..., M
10. Perbarui langkah waktu; t ← t +1. 11. Kembali ke langkah ke (6).
11. Simpan kromosom berisi bobot akhir terbaik
12. Imputasi data yang hilang untuk menghasilkan data yang lengkap
13. Hitung Error Data lengkap
14. Hentikan proses jika kriteria terpenuhi ketika proses berjalan hingga generasi tertentu dengan error<10%.
15. Jika kriteria tidak terpenuhi maka ulangi ke langkah 6
16. Selesai

2.1 Menyiapkan Data Uji

Algoritma GAKMI diawali dengan menginisialisasi parameter *K-Means* dengan menentukan pusat *cluster*, menentukan matriks φ dan matrik t untuk mencari bobot (w). Menginisialisasi GA dengan menentukan populasi untuk kromosom dengan bobot yang sesuai dengan *cluster*, dengan pencarian MSE sampai iterasi *epochs* sesuai dengan Tabel 1.

Tabel 1. Kriteria Terminasi GAKMI

Kriteria	Nilai
Batas Error Kriteria berhenti	10%
Ukuran Populasi	50
Skema Pengkodean	Bilangan riel
Fungsi *Fitness*	*Error K-Means*
Crossover	*Crossover* 2 titik
Peluang *Crossover*	0.5-1
Peluang Mutasi	0.1-0.5
Mekanisme Seleksi	*Roulette Wheel*
Seleksi Survivor	Generational Replacement

Penggunaan parameter untuk pencarian dibatasi dibatasi sampai nilai fitness stabil dengan error tertentu, ukuran populasi awal 50, tingkat probabilitas mutasi Pm= 0.2 dan Probailitas *Crossover* Pc=0.8. Kromosom GA akan digunakan untuk mewakili bobot awal dan *update cluster* pada KMI, w_n mewakili sejumlah fitur dan n mewakili jumlah *cluster*. bit. Penggunaan set data dengan kromosom bilangan real untuk mewakili nilai bobot untuk imputasi data dengan tujuan menghasilkan data lengkap.

2.2 Menyiapkan Data Uji

Uji coba yang dilakukan adalah menggunakan data sintesis penyakit hepatitis dengan mengunduh dari UCI repository pada web site http:archive.ics.uci.edu/ml/datasets/Hepatitis.Verifikasi kinerja GAKMI menggunakan pyton 3. Data hepatitis terdiri jumlah total fitur adalah 19 dengan atribut campuran yang dominan mengandung data kategori yang berjumlah 14 fitur. Sehingga proses skala perlu dilakukan untuk memudahkan proses perhitungan.

Tabel 2. Teknik skala data hepatitis dengan atrbiut kategori

No.	Atribut	Nilai Domain	Nilai Teknik Skala
1	Kelas/label keputusan	Die, Live	Die=1, Live=2
2	Jenis Kelamin	Laki-laki, perempuan	Laki-Laki =1, Perempuan=2
3	Steroid	No, Yes	No=0, Yes =1
4	Antiviral	No, Yes	No=0, Yes =1
5	Fatigue	No, Yes	No=0, Yes =1
6	Malaise	No, Yes	No=0, Yes =1
7	Anorexia	No, Yes	No=0, Yes =1
8	Liver Big	No, Yes	No=0, Yes =1
9	Liver Firm	No, Yes	No=0, Yes =1
10	Spleen Palpable	No, Yes	No=0, Yes =1
11	Spiders	No, Yes	No=0, Yes =1
12	Ascites	No, Yes	No=0, Yes =1
13	Varices	No, Yes	No=0, Yes =1
14	Histology	No, Yes	No=0, Yes =1

Atribut campuran membutuhkan teknik skala dengan mengonversi data kategori yang memiliki nilai 'tidak' dan 'ya' menjadi 0 dan 1. Variabel *survival biner* dinyatakan sebagai Kelas yang berupa resiko kematian 'Die' dan resiko hidup 'Live' telah dikodekan ke kategori dalam bentuk numerik (masing-masing 1 dan 2).

Pada kenyataanya data Hepatitis terdiri beberapa fitur yang mengandung nilai yang hilang, dengan prosentase yang berbeda-beda. Jumlah nilai yang hilang pada masing-masing fitur berbeda-beda sesuai Tabel 3 yaitu SGOT = 4, Alk Phosphate = 20, Albumin = 15, Protime = 40, dan Bilirubin =5. Atribut dengan *missing value* diindikasikan oleh nilai "?".

Tabel 3. Variabel missing value pada data hepatitis

No.	Atribut	Jumlah Missing Value
1.	Kelas/Label Keputusan	0
2.	Umur	0
3.	Jenis Kelamin	0
4.	Steroid	1
5.	Antiviral	0
6.	Fatigue	1
7.	Malaise	1
8.	Anorexia	1
9.	Liver Big	10

No.	Atribut	Jumlah Missing Value
10.	Liver Firm	11
11.	Spleen Palpable	5
12.	Spiders	5
13.	Ascites	5
14.	Varices	5
15.	Bilirubin:	6
16.	Alk Phosphate	29
17.	Sgot	4
18.	Albumin	16
19.	Protime	67
20.	Histology	0

Pengujian yang dilakukan dengan menyiapkan pre-processing terhadap data latih dengan mengubah data kategori menjadi numerik. Algoritma genetika menggunakan bilangan riel untuk menganti formulasi update bobot pada *K-Means*. Data yang digunakan sebanyak 155 data dengan 2 label (kelas). Selanjutnya melakukan pelatihan data dengan menggunakan data keseluruhan (*full train*) untuk algoritma genetika untuk menghasilkan kromosom akhir berupa bobot untuk proses imputasi pada *K-Means*, untuk menghasilkan data lengkap. Proses pengujian menggunakan data testing secara keseluruhan (*full test*) untuk mengelompokkan data sehingga diperoleh kinerja yang diinginkan.

3. KAJIAN PUSTAKA

3.1 *K-Means* Imputation (KMI)

Salah satu teknik heuristik paling populer untuk menyelesaikan *clustering* adalah *K-Means* untuk menyelesaikan masalah didasarkan pada skema iteratif sederhana untuk menemukan solusi minimal local (Binu, 2015). Algoritma *K-Means* diusulkan oleh J.B. MacQueen merupakan algoritma yang tidak diawasi bertujuan untuk meminimalkan indeks kesalahan kinerja *cluster*. Algoritma *K-Means* memiliki kelebihan dalam hal efisiensi dan kecepatan. Namun, algoritma ini sangat bergantung pada bobot awal dan nilai k. algoritma ini berdasarkan pada fungsi target selalu menggunakan metode gradien untuk mendapatkan ekstrem. Simulasi satu set dari *n* titik data dalam ruang dimensi *d*, R_d, dan bilangan integer *k*. Perhitungan varian menggunakan jarak eucledian menentukan satu set titik *k* di R_d disebut pusat, dengan meminimalkan jarak rata-rata kuadrat dari setiap titik data ke pusat terdekatnya (Viloriaa, 2019).

Pelatihan K-Means dilakukan pada setiap input vektor yang akan dipetakan pada *cluster* dengan bobot terdekat. *K-Means* menggunakan bobot pemenang dan semua tetangganya untuk diperbarui secara dinamis oleh fungsi objektif yag membutuhkan teknik optimasi sesuai dengan data input yang disediakan. Setiap perubahan interval waktu selama pembelajaran membutuhkan pembaruan bobot satu per satu secara dengan pengaturan parameter dinamis berulang kali untuk mendapatkan data output yang cocok dengan sistem target (Binu, 2015). *K-Means Imputation* (KMI) menggunakan bobot dari hasil pelatihan K-Means konvensional untuk proses pengisian data (Ahmad, 2007, Anwar, 2019). Sedangkan *KMI* untuk mengatasi kondisi data yang tidak lengkap dengan menerapkan perbaikan data pada tingkat preprosesing. Tahapan algoritma *K-Means* Imputation menunjukkan input vektor inisialisasi sebagai fitur input $x_1, x_2, x_3, \ldots x_n$ dan output untuk mendapatkan klaster $y_1, y_2, y_3, \ldots y_n$ yang diinginkan.

1. Tentukan jumlah *cluster c* dan *centroid* awal (w_0) secara random dari obyek-obyek data komplit yang tersedia sebanyak c *cluster*.
2. Hitung jarak terdekat pada setiap neuron keluaran untuk memasukkan data menggunakan jarak Euclidean:

$$D_{ip} = \sqrt{\sum_{i=1}^{n}(x_{ij} - w_{pj})} \quad (1)$$

dengan
D_{ip}: jarak antara obyek ke-1 dan centroid cluter ke
x_{ij} : data pada obyek ke-i pad fitur ke
w_{pj}: bobot centroid cluter ke pada fitur ke n banyaknya fitur

3. Kelompokkan setiap obyek berdasarkan jarak terdekat antara setiap obyek dengan masing-masing centroid.
4. Lakukan iterasi (t), tentukan posisi centroid pada iterasi ke-t (*c*) dengan rumus sebagai berikut:

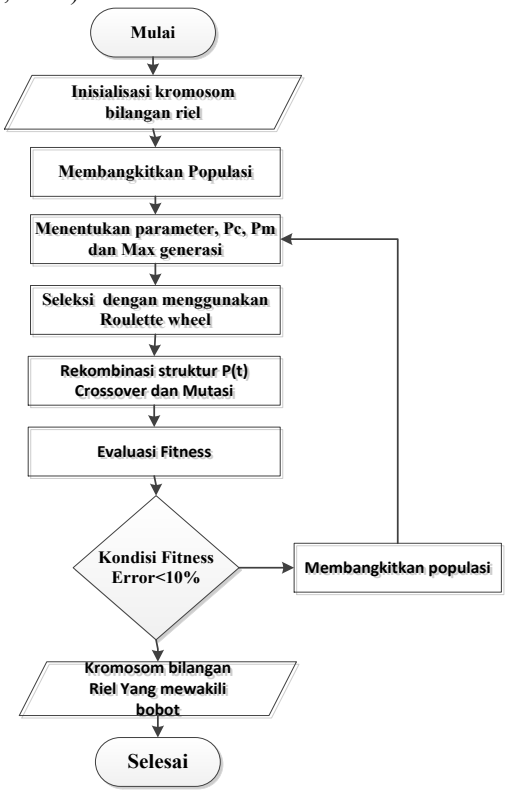
$$U_{ij} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

Dengan:
U_{pj}= centroid *cluster* ke-i pada fitur ke-j
n = banyak/jumlah data yang menjadi anggota *cluster* ke-i
x_{ij} : data pada obyek ke-i pada fiturl ke-j

4. Ulangi langkah 3 jika posisi MSE<10%, Jika "Ya" maka proses perhitungan dihentikan dan dihasilkan kelompok data akhir.
5. Jika "Tidak", periksa langkah sebelumnya hingga tidak ada bobot pembaruan hingga mencapai kondisi berhenti sampai kesalahan terkecil dihasilkan sesuai batas nilai yang diinginkan. Seluruh bobot (Di) adalah jarak dicari terkecil, maka indeks bobot (Di) disebut pemenang. Nilai kesalahan terkecil melibatkan hasil prediksi yang lebih baik, dan sebaliknya.
6. Isi *missing data* dengan *centroid yang* sesuai dengan letak *missing data* berada. Hasil akhir *K-Means* dengan $x_{ip} = w_{ip}$ diperoleh dari hasil proses perhitungan jarak Euclid yang hasilnya berupa inisialisasi bobot (Khotimah, 2020).

3.2 Algoritma Genetika

Golberg (1989) mengembangkan algoritma genetika untuk prosedur pencarian dan optimasi dengan menggunakan pengkodean serangkaian solusi masalah. Algoritma genetika mencari solusi dari kumpulan individu berupa populasi untuk memecahan masalah tersebut. Populasi terdiri sejumlah kromosom yang dipilih secara acak untuk mendapatkan solusi terbaik yang diinginkan (El-Sawy, 2014).



Gambar 1. Ilustrasi kinerja algoritma GA

Generasi baru ini dikenal sebagai keturunannya. Semakin tinggi kesesuaian anggota yang berbeda dalam populasi untuk bereproduksi ke generasi berikutnya sampai proses berlanjut menuju pencapaian yang konvergen (El-Sawy, 2014).

Gambar 1. Ilustrasi GA, generasi baru berasal dari operator reproduksi / seleksi, crossover, dan mutasi. Proses ini dilakukan berulang kali sehingga ditemukan jumlah kromosom yang cukup untuk membentuk generasi baru sebagai representasi dari solusi baru (Maulik, 2000, Ribelito, 2002). Kromosom akan dievaluasi dengan menggunakan pengukuran yang disebut kebugaran. Kromosom menggambarkan bobot yang terbaik berupa kromosom bilangan riel dalam populasi itu.

3.3 Algoritma Genetika Untuk Pengaturan Bobot Dinamis

Aturan pembaruan bobot dimulai dengan memungkinkan vektor bobot untuk mendekati area di sekitar solusi optimal dan *cluster* vektor bobot kemudian pindah ke pusat output *cluster* optimal *K-Means* (Kaczmarowski, 2015, Farag, 2015a, 2015b, Al Malki, 2016, Islam, 2020). Inisialisasi GA secara acak menghasilkan solusi awal yang ditetapkan dalam kromosom populasi yang mewakili bobot untuk setiap fitur ditunjukkan pada Gambar 2. Posisi gen dalam kromosom yang digunakan untuk itu adalah identifikasi fitur dan nilainya digunakan untuk mewakili bobot *K-Means*.

		G1			…	Gn
C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	…	C_{mn}
W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	…	Wmn

Gambar 2. Representasi Kromosom pada Bobot Masing-masing Fitur

Pembobotan pada *K-Means* untuk fitur berdasarkan jumlah *cluster* yang ditetapkan. Setiap kromosom mewakili bobot setiap seri fitur pada seri waktu tertentu. Optimalisasi untuk menemukan vektor bobot terbaik yang terkait dengan w_j untuk mewakili 1xN pada setiap elemen fitur (Zeebaree, 2017, Khotimah, 2020).

$$w_{mn} = (w_{11}, w_{12}, \dots, w_{mn}) \quad (3)$$

$$\sum_{j=1}^{m} w_j = 1, w_j \geq 0, \quad j = 1,2, \dots, m \quad (4)$$

Kromosom GA akan digunakan untuk mewakili bobot awal klaster KMI. Gene berisi satu set bobot pada semua fitur variabel, ketika m mewakili sejumlah fitur dan n mewakili jumlah *cluster*. Kromosom terus berubah menjadi generasi populasi baru melalui proses seleksi, crossover, dan mutasi. Proses tersebut berevolusi menggantikan seluruh kromosom dalam populasi. Kromosom terdiri dari sejumlah gen yang menunjukkan bobot yang akan dialokasikan untuk masing-masing fitur-fitur.

$$fitness = \frac{1}{MSE + h}, 0 \leq h \leq 1 \quad (5)$$

$$MSE = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n} \quad (6)$$

dengan

Y_i = target

$\hat{Y}_i$ = prediksi

GA menggunakan fungsi kebugaran untuk memandu kinerja GAKMI untuk kasus yang harus diselesaikan. MSE digunakan secara langsung sebagai fungsi kebugaran (*fitness*), kesesuaian untuk pemilihan proporsional harus meningkatkan efisiensi ke kondisi yang konvergen. Indeks MSE semakin kecil berarti semakin optimal, dengan iterasi tertentu. kromosom pertama di kombinasikan bagian kedua dari kromosom kedua menjadi keturunan pertama. Sedangkan bagian kedua kromosom pertama di kombinasikan dengan bagian pertama kromosom kedua menjadi keturunan kedua. Hal ini dilakukan pada masing-masing pasangan induk yang akan di *Crossover (*Rahman, 2014).

Proses imputai dilakukan pada data x_i membutuhkan nilai kelas c_k dan nilai bobot atribut

w_{ij} Prosedur pengisian nilai yang hilang berupa kumpulan data X_i yang terdiri x_m nilai yang hilang dan x_c tanpa nilai yang hilang. Vektor bobot urutan di setiap *i=1,2,...,j* pada KMI akan dimasukkan kedalam data berdasarkan kelas dan letak kriteria yang sesuai pada data yang hilang $X_{ir} = (c_i, w_j)$ (Khotimah, 2020).

$$1 \leq w_{ij} \leq p, \text{ if } w_{ij} = x_{im}(i,m) \in N_m, \quad (7)$$

4. HASIL DAN PEMBAHASAN

Kromosom GA akan digunakan berupa bobot awal pada KMI. Gen berisi satu set bobot pada semua fitur variabel, ketika *m* mewakili sejumlah fitur dan n mewakili jumlah *cluster*. Skenario ujicoba menggunakan satu set kumpulan data memiliki 19 atribut dan jumlah *cluster* yang diharapkan adalah 2. Gen berisi 19 kromosom dengan jumlah populasi 50. Kromosom berupa bilangan real untuk dengan mudah mewakili nilai bobot imputasi aktual. bobot digunakan untuk uji validasi KMI.

Proses imputasi KMI untuk data missing value dengan imputasi ganda berdasarkan nilai k untuk replika. Contoh data hepatitis pada 4 fitur dan 10 baris pertama terdiri fitur yang mengandung nilai yang hilang dengan prosentase hamper sama kurang lebih 5%. Fitur tersebut adalah Spleen Palpable, Spiders, Ascites, Varices. Hasil akhir learning KMI pada fitness mendekati 0, misalnya pada iterasi 420 menghasilkan bobot *centroid* dan anggota *cluster* 1 yaitu obyek 1, 2, 4, serta *cluster* 2 yaitu obyek 3, 5, 6, 7, 8, 9, 10.

$$w_{420}^1 = (-0.227,; 1.342.; 1.44; 0.642)$$
$$w_{420}^2 = (-0.302,; -0.879.; -0.445; -0.322)$$

Tabel 3. dan Tabel 4. Proses pengisian data misalnya $x_{mis}(x_{13}, w_{31})$ artinya data yang hilang pada fitur ke 3 dan kelas 1. Kemudian mengisi *missing data* dengan *centroid* yang sesuai dengan letak *missing data* berada yaitu

$x_{mis}(x_{11}, w_{31}) = 1.443, x_{mis}(x_{31}, w_{12}) = -0.302; x_{mis}(x_{42}, w_{21}) = 1.342;\ x_{mis}(x_{71}, w_{12}) = -0.302;$ dan $x_{mis}(x_{92}, w_{22}) = -0.879.$

Proses replika dilakukan dengan melakukan pengisisan data secara berulang. Penggunaan replika untuk mengetahui tingkat imputasi keberapa nilai data mendekati aslinya. Untuk mengukur kinerja algoritma pada suatu data setelah proses imputasi, maka dilakukan pengujian dengan menggunakan MSE sesuai Tabel 5.

Tabel 3. Data Preimputasi

No. User	Spleen Palpable	Spiders	Ascites	Varices
1.	-0.272	0.541	NaN	1.129
2.	-0.272	0.360	0.073	1.228
3.	NaN	1.535	-0.073	-0.928
4.	-0.791	NaN	-0.073	-0.281
5.	-0.791	-0.612	-1.622	-0.271
6.	-0.791	-0.287	-1.622	1.427
7.	NaN	0.175	-1.622	1.342
8.	-0.272	-0.867	-0.073	-1.32
9.	-0.272	NaN	-1.622	0.819
10.	-0.791	-1.384	-1.622	0.239

Tabel 4. Data Hasil Pascaimputasi

No. User	Spleen Palpable	Spiders	Ascites	Varices
1.	-0.272	0.541	1.443	1.129
2.	-0.272	0.360	0.073	1.228
3.	-0.302	1.535	-0.073	-0.928
4.	-0.791	1.342	-0.073	-0.281
5.	-0.791	-0.612	-1.622	-0.271
6.	-0.791	-0.287	-1.622	1.427
7.	-0.302	0.175	-1.622	1.342
8.	-0.272	-0.867	-0.073	-1.32
9.	-0.272	-0.879	-1.622	0.819
10.	-0.791	-1.384	-1.622	0.239

Tabel 5. Nilai MSE berdasarkan proses replika

No.	Repl 1	Repl 2	Repl 3	Repl 4	Repl 5
K=2	0,302	0,605	0,532	0,875	0,466
K=2	0,323	0,052	0,932	0,704	0,766
K=3	0,092	0,502	**0,215**	0,855	**0,044**
K=4	0,203	0,456	0,879	0,954	0,659
K=5	0,695	0,367	0,667	0,674	0,586

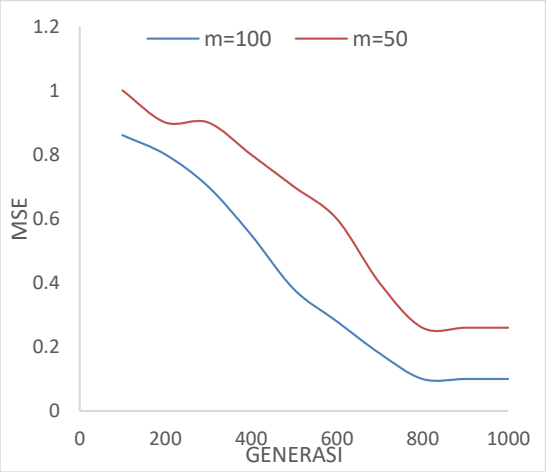
Tabel 5 menunjukkan hasil proses replika pada setiap imputasi pada fitur yang terdapat *missing data*. Hasil imputasi *missing data* menggunakan metode Algoritma GAKMI dengan 2 *cluster* menghasilkan nilai MSE pada replika ke- 1, 2, … 5 bahwa replika ke-3 dan ke-5 pada *K*=3 memberikan nilai MSE yang lebih kecil dibandingkan yang lain. Pada setiap replika menghasilkan MSE yang berbeda-beda, karena nilai bobot untuk mengisi nilai yang hilang bervariasi.

Gambar 3. Penggunaan GAKMI dengan perbedaam ukuran populasi menunjukkan nilai MSE yang sangat fluktuatif. Semakin tinggi populasinya, semakin kecil nilai kesalahan yang dihasilkan. Semakin tinggi populasi mengakibatkan jumlah generasi meningkat dan tingkat kesalahan yang dihasilkan semakin kecil.

Tabel 6. Perbandingan populasi terhadap waktu eksekusi

Parameter	Perubahan Jumlah Populasi	*Fitness*	Iterasi fitness akhir	Waktu Eksekusi (*second*)
Pc = 0,5	50	0,928	603	834
Pm = 0,5	100	0,823	721	1174
Pc = 0,6	50	0,864	425	762
Pm = 0,4	100	0,706	676	1478
Pc = 0.7	50	0,880	506	706
Pm = 0,3	100	0,914	639	1059

Parameter	Perubahan Jumlah Populasi	*Fitness*	Iterasi fitness akhir	Waktu Eksekusi (*second*)
Pc = 0,8	50	0,671	576	322
Pm = 0,2	100	0,891	821	1421
Pc = 0,9	50	0,915	853	301
Pm = 0,1	100	0,588	186	811



Gambar 3. Perbandingan pengaruh populasi terhadap fitness

Tabel 6 menunjukkan hasil pengujian berdasarkan penggunaan parameter dengan perbedaan populasi, parameter Pc dan Pm. Perubahan nilai probabilitas crossover dan mutasi menghasilkan nilai fitness yang bervariasi. Nilai probabilitas crossover terbaik antara 0.5 sampai 1. Semakin besar peluang crossover maka perubahan data semakin tinggi. Sehingga data berpeluang besar mendekati aslinya. Sedangkan nilai probabilitas mutasi berkisar 0.5-0.1, karena semakin kecil mutasi maka perubahan kecil. Algoritma GAKMI berkerja baik dalam proses estimasi bobot dengan nilai MSE yang kecil. GA mengeksplorasi nilai hak untuk mengatur bobot untuk memeriksa setiap generasi, solusinya akan meningkat. Peningkatan kualitas kromosom di semua kovariat sangat signifikan untuk menangani data yang mempunyai fitur mayoritas kategori.

Tabel 7. Uji korelasi r dan uji t berpasangan data Hepatitis

No	KMI		GAKMI		*Mean* Imp	
k	t	r	t	r	t	r
2	0,564	0,202	0,392	0,526	0,046	-0,235
3	-0,298	0.523	0,002	0,574	0,044	-0,014
4	0,195	-0,268	0,291	0,129	0,658	-0,087
5	0,86	-0,574	0,063	0,064	0,564	-0,096

Signifikan pada α=0.05

Uji korelasi digunakan untuk menguji tingkat seberapa jauh nilai pengganti imputasi mendekati dengan nilai aslinya. Tabel 7 menunjukkan imputasi yang baik adalah ketika nilai *r* positif, yang berarti metode imputasi cocok untuk mengganti nilai data yang hilang. Nilai *r* yang mendekati 1 yang paling baik untuk teknik imputasi. Tabel 7 menunjukkan tingkat signifikan (lebih besar dari α=0.05) sehingga H_0 diterima. Algoritma GAKMI selalu memberikan nilai r lebih besar dibanding metode *Mean,* dan KMI itu sendiri yang tanpa optimasi. Algoritma GAKMI dengan nilai *r* dominan positif karena dapat memilih nilai replika yang mendekati nilai aslinya dengan MSE terkecil, sehingga GAKMI menunjukkan hasil yang lebih baik dibanding metode metode imputasi lainnya.

5. KESIMPULAN

GA memilih bobot yang tepat akan memengaruhi pengelompokan *K-Means*, yang menghitung jarak terkecil antara data dengan pusat klaster (centroid). Pengaruh nilai *k* pada *K-Means* menunjukkan bahwa MSE semakin tinggi. Hasil penelitian GAKMI menunjukkan perbedaan signifikan dalam nilai mean square error (MSE) pada setiap generasi dan jumlah populasi GA, nilai MSE terbesar adalah 0,928 dan yang terkecil adalah 0,588. Sedangkan pada pengujian kecocokan imputasi nilai korelasi (r) pada GAKMI cenderung positif, artinya GAKMI cenderung lebih unggul dibandingkan dengan KMI dan Means yang cenderung nilai *r* nya negatif. Karena GAKMI menggunakan pemilihan bobot yang sesuai dan mendekati nilai sebenarnya..

DAFTAR PUSTAKA

AL KINDHI, B., SARDJONO, T. A., PURNOMO, M. H., VERKERKE, G. J., 2019. Hybrid K-means, Fuzzy C-Means, And Hierarchical Clustering for DNA Hepatitis C Virus Trend Mutation Analysis. Expert Systems with Applications, Issue 121, 1 May , pp. 373-381.

A ACUNA, E, AND RODRIGUES C., 2004. The Treatment of Missing Values and its Effect is the Classifier Accuracy. Proceedings of the Meeting of the International Federation of Classification Sociaties (IFCS), 15 Juli 2004.

AL MALKI, A., MOHAMED M. RIZK1, M. M., EL-SHORBAGY, M. A., MOUSA, A. A., 2016. Hybrid Genetic Algorithm with K-Means for Clustering Problems, Open Journal of Optimization, issue 5, pp. 71-83.

ANWAR, T., SISWANTINING, T., SARWINDA, D., SOEMARTOJO, S. M., BUSTAMAM, A., A study on missing values imputation using K-Harmonic means algorithm: Mixed datasets. AIP Conference Proceedings, issue 2202, 020038 (2019), pp.1-10

BINU, D. 2015. *Cluster* Analysis Using Optimization Algorithms with Newly Designed Objective Functions. Expert Systems with Applications, 42(14), pp.5848- 5859.

CHEHOURI, A., R. YOUNES, R., KHODER, J., PERRON, J., ILINCA, A., 2017. A Selection Process for Genetic Algorithm Using

Clustering Analysis. Algorithms, 10(123), pp. 1-15

DAVEY, A, And SAVLA J., 2010. Statistical Power Analysis with Missing Data. New York: Taylor and Francis Group.

ENDERS, C. K., 2014. Applied Missing Data Analysis [monograph online]. New York: The Guilford Press.

EL-SAWY, A. A., HUSSEIN, M. A., ZAKI, E.M. AND MOUSA, A. A., 2014. An Introduction to Genetic Algorithms: A Survey A Practical Issues. International Journal of Scientific & Engineering Research, 5(1), pp.252.

FARAG, M.A., EL-SHORBAGY, M.A., EL-DESOKY, I.M., EL-SAWY, A.A. MOUSA, A.A., 2015. Genetic Algorithm Based on *K-Means-Clustering* Technique for Multi-Objective Resource Allocation Problems. British Journal of Applied Science & Technology, 8(1), pp. 80-96. http://dx.doi.org/10.9734/BJAST/2015/16570

FARAG, M. A., EL-SHORBAGY, M.A., EL-DESOKY, I. M., EL-SAWY, A.A., MOUSA, A. A., 2015. Binary-Real Coded Genetic Algorithm Based K-Means Clustering for Unit Commitment Problem. Applied Mathematics, 6(11), pp.1873-1890. http://dx.doi.org/10.4236/am.2015.611165

HAN, J. AND KAMBER, M. 2000. Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco.

IZZAH, A., AND HAYATIN, N., 2013. Imputasi Missing Data Menggunakan Algoritma Pengelompokan Data K-Harmonic Means. Seminar Nasional Matematika dan Aplikasinya (SNMA), 21 September.

ISLAM, M. T. P., BASAK, K., BHOWMIK, P., KHAN, M., 2020. Data Clustering Using Hybrid Genetic Algorithm with k-Means and k-Medoids Algorithms. 2019 23rd International Computer Science and Engineering Conference (ICSEC), IEEE *Xplore*. Phuket, Thailand, 30 January 2020.

KACZMAROWSKI, A., YANG, S., SZLUFARSKA, I. AND MORGAN, D. 2015. Genetic Algorithm Optimization of Defect *Cluster*s in Crystalline Materials. Computational Materials Science, 98(1), pp. 234-244.

KHOTIMAH, B. K., MISWANTO, SUPRAJITNO, H., 2020. Optimization of Feature Selection Using Genetic Algorithm in Naïve Bayes Classification for Incomplete Data, International. Journal of Intelligent Engineering and Systems (IJIES), Issue.13, 2020, pp.334-343.

KHOTIMAH, B. K., IRHAMNI, F., SUNDARWATI, T., 2016. A Genetic Algorithm for Optimized Initial Centers K-Means Clustering in SMEs. Journal of Theoretical and Applied Information Technology, 15 August, 90(1), pp. 23-30

LI D., DEOGUN J., SPAULDING W., SHUART B. Toward Missing Data Imputation: Study of Fuzzy K-Means *Clustering* Method. Proceedings 4 th International Conference. 2004 jun 1.

MAULIK, U. AND BANDYOPADHYAY, S. 2000. Genetic Algorithm Based *Clustering* Technique. *Pattern Recognition,* 33 (9), pp 1455 -1465.

MARGHNY, M. H., EL-AZIZ, R. M. A., TALOBA, R. M. A., 2011. An Effective Evolutionary Clustering Algorithm: Hepatitis C Case Study. International Journal of Computer Applications, 34(6), pp. 1-6.

RAHMAN, M.A. AND M.Z. ISLAM, 2014. A Hybrid *Clustering* Technique Combining a Novel Genetic Algorithm with *K-Means*. Knowledge-Based Systems, 71, p. 345-365.

T. Mahboob, A. Ijaz, A. Shahzad, and M. Kalsoom, Handling Missing Values in Chronic Kidney Disease Datasets Using KNN, K-Means and K-Medoids Algorithms. International Conference on Open Source Systems and Technologies (2018).

VILORIAA A., LEZAMA, O. B. P., 2019. Improvements for Determining the Number of Clusters in k-Means for Innovation Datab ases in SMEs. Procedia Computer Science, Issue 151, pp.1201–1206.

ZEEBAREE, D. Q., HARON, H., ABDULAZEEZ, A. M., AND SUBHI R. M., 2017. Combination of *K-Means Clustering* with Genetic Algorithm: A Review. International Journal of Applied Engineering Research, 12 (24), pp. 14238- 14245.