Penentuan *Centroid* Awal Pada Algoritma K-Means Dengan *Dynamic Artificial Chromosomes Genetic Algorithm* Untuk *Tuberculosis Dataset*

Pre-Centroid Determination in K-Means Algorithm using Dynamic Artificial Chromosomes Genetic Algorithm for Tuberculosis Dataset

Mursalim[1], Purwanto[2], M Arief Soeleman[3]
[1,2,3] Fakultas Ilmu Komputer, Program Magister Teknik Informatika, Universitas Dian Nuswantoro
Email: [1]mursalim.dsc@gmail.com, [2]purwanto@dsn.dinus.ac.id, [3]m.arief.soeleman@dsn.dinus.ac.id

Abstrak

Data mining merupakan disiplin ilmu yang mempelajari metode untuk mengekstraksi data menghasilkan informasi. Salah satunya adalah klastering yang berfungsi untuk mengelompokkan data berdasarkan tingkat kemiripan dan jarak minimum. Algoritma K-Means sangat populer dan banyak digunakan diberbagai bidang seperti bidang pendidikan, kesehatan, sosial, biologi, ilmu komputer. Metode K-Means sering dikombinasikan dengan metode optimasi seperti algoritma genetika atau GA untuk mengatasi permasalah pada K-Means yaitu sensitif dalam penentuan *centroid* awal. Penelitian ini mengkombinasikan *Dynamic Artificial Cromosomes Genetic Algorithm* atau DAC GA dengan K-Means dalam menentukan nilai centroid awal. Hasil eksperimen menunjukkan bahwa metode DAC GA + K-Means lebih unggul dibandingkan dengan K-Means dan GA + K-Means. 2 dataset yang diuji dengan optimal nilai klaster sebanyak 2 dan 1 dataset sebanyak 3 klaster. Metode tersebut perolehan nilai DBI sebesar 0.138, 0.279 serta 0.382, nilai *Sum Square Error* sebesar 92.56, 332,39 dan 1280.68 serta nilai fitness yang tebentuk adalah 7.12, 3.57 dan 2.13.

Kata kunci: *Initial centroid, k-means, Sum Square Error, Davies Bouldin Index, DAC-GA, Tuberculosis*

Abstract

Data mining is a scientific discipline that studies methods for extracting data to produce information. One of them is clustering which functions to classify data based on the level of similarity and minimum distance. The K-Means algorithm is very popular and widely used in various fields such as education, health, social, biology, computer science. The K-Means method is often combined with optimization methods such as genetic algorithms or GA to solve the problem of K-Means, which is sensitive in determining the initial centroid. This study combines Dynamic Artificial Chromosomes Genetic Algorithm or DAC GA with K-Means in determining the initial centroid value. The experimental results show that the GA + K-Means DAC method is superior to the K-Means and GA + K-Means methods. 2 datasets were tested with optimal cluster values of 2 and 1 dataset of 3 clusters. This method obtains DBI values of 0.138, 0.279 and 0.382, the Sum Square Error values of 92.56, 332.39 and 1280.68 and the formed fitness values are 7.12, 3.57 and 2.13.

Keywords: *Initial centroid, k-means, Sum Square Error, Davies Bouldin Index, DAC-GA, Tuberculosis*

1. PENDAHULUAN

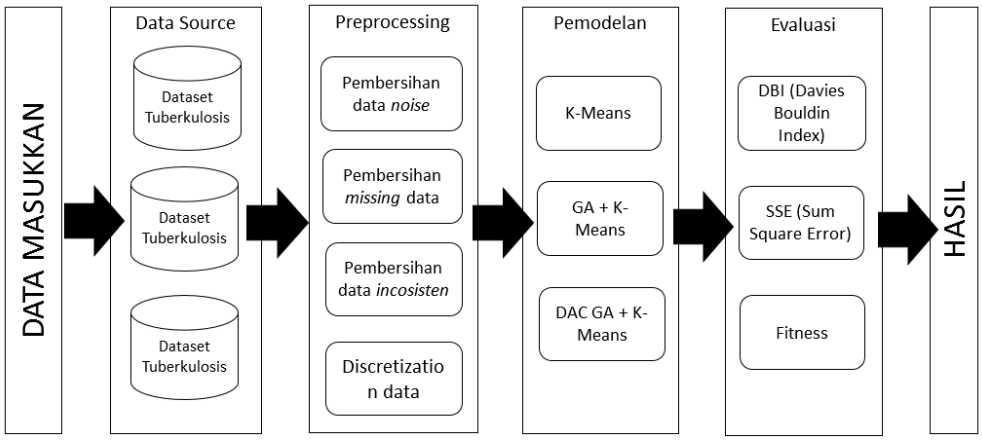
Klastering atau data segmentasi merupakan metode *Unsupervised learning* yang dapat mengelompokkan data ke dalam beberapa *cluster* atau kelompok berdasarkan tingkat kemiripan[1,2]. Klastering memiliki 6 syarat yaitu: skalabilitas, kemampuan analisa beragam bentuk data, menemukan klaster dengan bentuk yang tidak terduga, kemampuan untuk menangani *noise*, sensitifitas terhadap perubahan *input*, mampu melakukan klastering untuk data dimensi tinggi dan interprestasi dan kegunaan [3]. K-Means merupakan metode klastering yang paling populer dan banyak diterapkan berbagai bidang yang membutuhkan klastering seperti kedokteran, biologi, kesehatan, matematika, teknik dan ilmu komputer[4].

Menurut[2] K-Means memiliki banyak keunggulan diantaranya adalah mudah diimplementasikan dan dijalankan, waktu proses relatif cepat, lebih mudah untuk diadaptasikan dengan metode lain dan sangat umum digunakan dikalangan para peneliti. Namun, algoritma K-Means juga memiliki kekurangan yaitu inisialisasi nilai *centroid* awal yang bersifat *random* sangat berpengaruh pada hasil pengelompokkan sehingga menjadi kurang optimal[2], penentuan nilai k yang masih menggunakan uji coba *trial* and *error, outlier* data [5]. Salah satu kekurangan Algoritma K-Means adalah inisialisasi nilai *centroid* awal yang bersifat *random*. Hal tersebut sangat sensitif terhadap hasil akhir pengklasteran. Permasalahan ini menjadi sudah menjadi pembahasan utama oleh para peneliti sebelumnya[6]–[14]. Algoritma K-Means sering dikombinasikan dengan metode lain salah satunya adalah algoritma genetika (GA) [6, 8, 11], [15]–[17].

Peneliti[6] mengkombinasikan antara K-Means dengan GGA untuk meningkatkan klaster optimal, hasilnya menunjukkan bahwa metode kombinasi tersebut memiliki kinerja yang baik dengan pendekatan statistik menggunakan *rand index* sebesar 0,873 dengan 3 klaster. Penelitian lain[8] mengatasi random pada penetuan klaster dengan mengkombinasikan K-Means dengan GA, hasilnya menunjukkan inisialisasi klaster dengan GA lebih tepat dibandingkan *randomly* (acak). Penelitian[14] menggunakan metode K-Means dengan *dimention reduction* dan selecting *cluster probability*, dimana hasil eksperimen menunjukkan hasil yang baik dibandingkan dengan metode K-Means biasa. Penelitian[16] menggunakan metode K-Means dengan GA paralel untuk mengatasi cluster yang sensitif dikarenakan penentuan centroid yang masih random, hasil penelitian menunjukkan bahwa metode kombinasi K-Means dengan GA paralel lebih baik dibandingkan dengan metode K-Means biasa. Penelitian[17]kombinasi metode *GenClust-H* dengan K-Means untuk menangani klaster data yang bersifat sensitif dalam penentuan klaster secara *random*, hasil menunjukkan bahwa metode yang diusulkan lebih baik dibandingkan dengan klaster dengan metode random pada K-Means biasa. Penelitian [18] GA dikembangkan dengan menerapkan *dynamic diversity control* yang diukur menggunakan rumus *linear scale measure* dan penelitan ini mengusulkan kombinasi *Dynamic Artifical Cromosoms Genetic Algorithm* (DAC) yang terbukti keluar dari optimum lokal atau konvergensi prematur [18].

Dari beberapa penelitian yang disajikan tersebut kombinasi metode K-Means dengan GA dapat digunakan untuk mengatasi permasalahan pada penentuan klaster maupun *centroid* di metode K-Means, merujuk pada penelitian [6, 8, 18] , metode GA memiliki kelemahan pada konvergensi prematur, hal ini tentunya dapat mempengaruhi hasil kombinasi metode K-Means dan GA itu sendiri, sehingga pada penelitian ini mencoba mengkombinasikan metode K-Means dengan DAC GA. Metode DAC GA tersebut terbukti keluar dari optimum lokal atau konvergensi prematur[18] dan kombinasi metode tersebut masih belum banyak dilakukan oleh peneliti. Sehingga hasil penelitian ini akan dibandingkan dengan metode K-Means biasa dan K-Means yang dikombinasikan dengan GA pada data *tuberculosis dataset.*

2. METODE PENELITIAN



Gambar 1 Desain penelitian penentuan *centroid* awal pada algoritma K-Means dengan *dynamic artificial chromosomes genetic algorithm*

2.1 Dataset

Dataset yang digunakan adalah 3 jenis dataset yang diperoleh dari website resmi *World Health Organization* (WHO) yaitu: dataset beban kasus tuberculosis MDR (*multidrug-resistant tuberculosis*), dataset faktor risiko tuberkulosis berdasarkan umur, jenis kelamin, dataset konfirmasi kasus baru pengawasan resistant [19].

2.2 Preprocessing

Proses preprosessing dilakukan dilakukan dengan beberapa pendekatan analisa data yaitu: menghapus data yang bernilai 0, menormalisasi data dengan rumus berikut:

$$v' = \frac{v - min_A}{max_n - min_n} (new\_max_n - new\_min_n) + new\_min_n \qquad (1)$$

Kemudian mendescretization dataset sesuai dengan kategori. Berikut adalah hasilnya:

age_group
0-65
15-24
0-65
0-65
……
all
all
all

Kategori Umur	Kategori (Angka)
0-4	1
0-14	2
15-24	3
15plus	4
25-34	5
35-44	6
45-54	7
55-64	8
65plus	9
all	10

sex
a
a
m
m
….
f
a
f

Sex	Number
f	0
m	1
a	2

Gambar 2 Proses mendescreatization dataset pada tahapan preprocessing

2.3 Metode K-Means

Menurut *Macqueen*[20] ada 8 tahapan dalam algoritma k-means, berikut adalah tahapannya:

1. Siapkan dataset
2. Tentukan jumlah klaster (k)
3. Inisialisasi *centroid* awal secara *random*
4. Hitung jarak tiap dataset dengan *centroid* menggunakan rumus *euclidian distance*, berikut rumusnya:

$$d_{Euclidean}(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \quad (2)$$

Keterangan:
n : banyak klaster
Xi : *attribut* data
Yi : *attribut centroid*

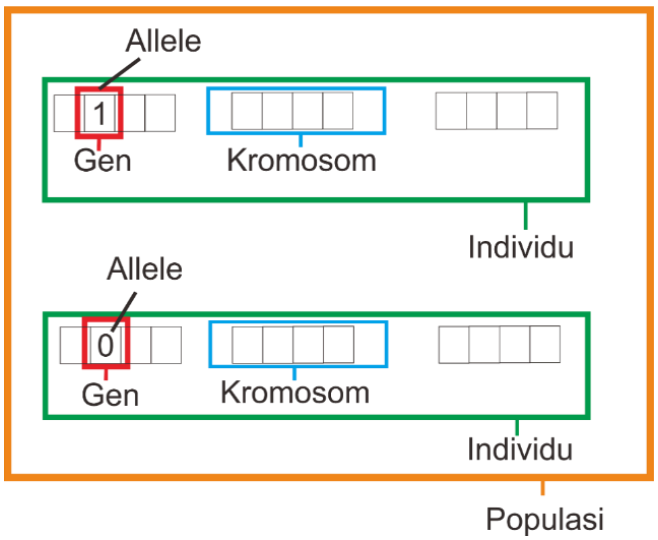
5. Kelompokkan data dengan klaster terdekat berdasarkan nilai jarak minimum, kemudian hitung nilai *Sum of Square Error* (SSE) dengan rumus berikut:

$$SSE=\sum_{i=1}^{K}\sum_{x\in C_i} d\,(x,\overline{x_{c_i}})^2 \quad (3)$$

Keterangan:
K : jumlah klaster
Xi : *attribut* data
Xci : *attribut centroid*
Ci : *centroid*

6. Menghitung kembali nilai *centroid* dengan keanggotaan klaster sekarang
   Kembali pada langkah 3-6 dengan nilai *centroid* terbaru
7. Proses akan berhenti jika nilai *centroid* tidak berubah atau tetap.

2.4 Algoritma Dynamic Artificial Cromosome Genetic Algorithm



Gambar 3 Ilustrasi perbedaan antara allele, gen, kromosom dan individu

Algoritma Genetika sendiri telah dipopulerkan oleh *John Holland* menjelaskan bahwa prinsip dasar perhitungannya menggunakan seleksi alam yang pernah dikenalkan oleh tokoh ilmuan biologi *Charles Darwin* [21]. Konsep perhitungan algoritma Genetika dapat dijelaskan dengan contoh dibawah ini:

1. Menentukan kromosom yang akan dibentuk, kemudian hitung
   Rumus *Encoding* kedalam desimal

$$X - r_b + \frac{r_a - r_b}{\sum_{i=1}^{N} 2^{-1}}(g_1 \times 2^{-1} + g_2 \times 2^{-2} + .. + g_n \times 2^{-n}) \quad (4)$$

2. Menghitung nilai *fitness* dari jumlah kromosom yang dibentuk

Dalam perhitungan nilai *fitness* bergantung pada perhitungan nilai fungsi minimal yang akan diambil adalah nilai terkecil maka rumus yang digunakan adalah:

$$\text{Nilai Fungsi} = \frac{N_c}{\sum_{j=1}^{N_c} \frac{\sum_{j=1}^{P_i} d(o_j m_{ij})}{Pi}} \qquad (5)$$

Keterangan:

C_i	:	*Cluster* ke-i
N_c	:	Jumlah *Cluster*
P_i	:	Jumlah data pada cluster C_i
o_i	:	Pusat cluster C_i
m_{ij}	:	Data ke-j dan merupakan anggota dari *cluster* C_i
$d(o_i m_{ij})$	:	Jarak antara $C_i$dan M_{ij}

$$Fitness = \frac{1}{Nilai\ Fungsi+E} \qquad (6)$$

Keterangan:

E : bilangan kecil yang ditentukan sendiri misalnya adalah bilangan 0,1. Bilangan tersebut digunakan untuk menghindari jika nilai fungsi = 0.

3. Memilih seleksi induk/Orang tua

Dalam mencari nilai fungsi minimum, nilai F min terkecil menjadi nilai *fitness* terbesar. diurutkan dan dicari nilai probabilitas setiap kromosom. Nilai K dengan Probabilitas terbesar masuk menjadi generasi selanjutnya. Rumus Probabilitas

$$P(i) = \frac{F(i)}{Total\ P(i)} \qquad (7)$$

Keterangan:

P(i)	: Nilai Probabilitas
F(i)	: Hasil perhitungan dengan fungsi objektif
Total P(i)	: Total dari perhitungan dengan fungsi objektif

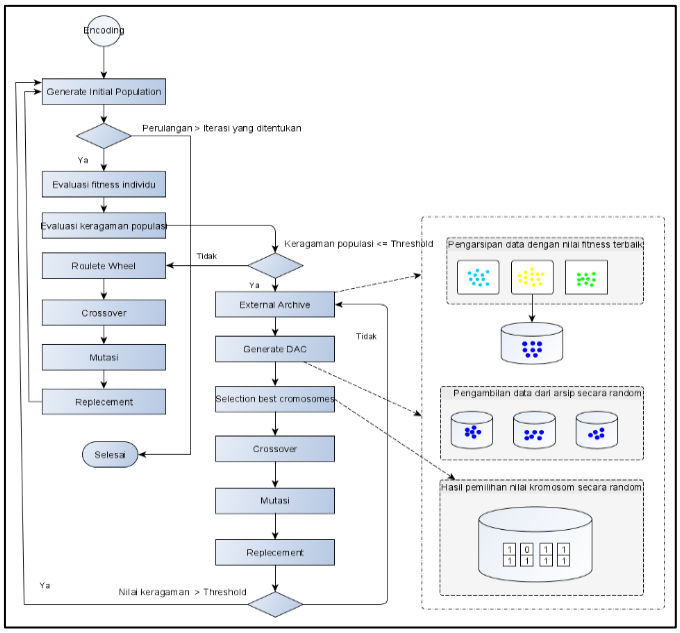
4. *Crossover* (Pindah Silang)

Probabiltas *Crossover* (PC) dilakukan dengan setengah dari kromosom yang dibentuk . Metode yang digunakan adalah *Single point Crossover*.

5. Mutasi

Proses mutasi ditentukan dari jumlah gen yang dibentuk dan kromosom yang diambil dan nilai probabilitas mutasi (PM) yang ditentukan Penentuan posisi gen yang dimutasi dilakukan secara *random.* nilai mutasi yang akan diubah adalah nilai 1 menjadi 0 dan sebaliknya.

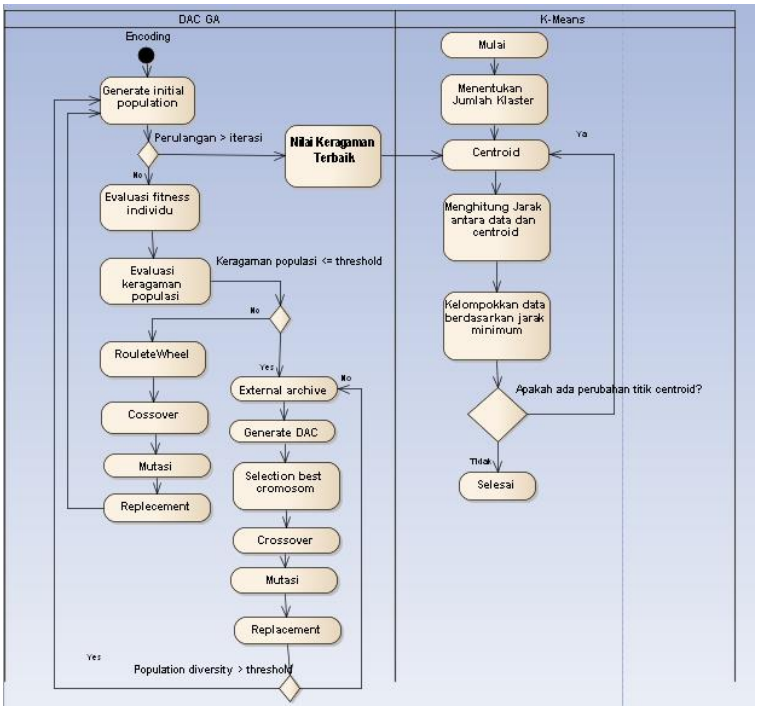
6. Proses tersebut terus dilakukan atau diulang hingga batas iterasi yang telah ditentukan.



Gambar 4 Tahapan Algoritma DAC-GA yang telah diusulkan[18]

Gambar 4 menjelaskan terkait dengan tahapan DAC-GA, yang dimulai dengan proses GA pada umumnya, kemudian proses evaluasi *fitness* dan evaluasi keragaman populasi (*population diversity*) diukur dengan rumus *linear scale measure*, apabila nilai keragaman turun kebawah, kurang dari atau sama dengan nilai dari *threshold*, maka kromosom buatan dinamis akan bekerja. Dalam penelitian GA-DAC sebelumnya telah menggunakan *threshold* dengan nilai yang terbaik sesuai dengan yang dilakukan [18, 22] yaitu: 0,5; 0,6 dan 0,7.

Model yang diusulkan



Gambar 5 metode penelitian DAC GA + K-Means

Proses optimasi Algoritma *Dynamic Artificial Kromosomes Genetic Algorithm* (DAC GA) pada K-Means dilakukan setelah data melalui tahap pengolahan awal data (Preprocessing). selanjutnya data dinormalisasikan dengan formula sebagai berikut:

$$v' = \frac{v - min_A}{max_n - min_n} (new\_max_n - new\_min_n) + new\_min_n \quad (8)$$

Keterangan:

v'	Normalisasi data
v	Awal data sebelum dinormalisasikan
max_n	Nilai maksimal dari data pada dimensi ke - n
min_n	Nilai minimum dari data pada dimensi ke-n
new_max_n	Nilai maksimum baru dari data pada dimensi ke-n
new_min_n	Nilai minimum baru dari data pada dimensi ke-n

Setelah proses normalisasi data selesai, kemudian hitung jumlah kromosom yang akan terbentuk dengan mengalikan jumlah klaster (k) dengan jumlah dimensi atau *attribut* (n) data. Selanjutnya melakukan proses algoritma genetika pada umumnya. Namun, setelah perhitungan nilai *fitness* terhadap individu yang terbentuk . Proses selanjutnya adalah menjalankan fungsi keragaman populasi dengan menggunakan rumus *linear scale measure* [18, 22] dengan nilai *threshold* sebesar 0,5. Diproses inilah kemudian nilai *fitness* yang rendah akan digantikan dengan nilai fitness yang terbaik dari eksternal arsip yang memiliki nilai *fitness* yang baik. Kemudian jalankan proses seleksi, crossover, mutasi dengan nilai probabilitas yang dihasilkan secara random. Proses *elitism* mengganti kromosomes sebelumnya dengan kromosomes baru dengan nilai *fitness* terbaik hingga syarat terpenuhi (mencapai batas maksimal iterasi). Selanjutnya hasil perhitungan tersebut dimasukkan kedalam proses klastering dengan metode K-Means berikut source codenya:

Berikut adalah bentuk *Pseudocode* Algoritma Genetika dan *Dynamic Artificial Chromosomes Genetic Algorithm* + K-Means

```
1   Gen(t) : Jumlah generasi
2   K : Jumlah nilai K
3   Indi = Jumlah individu
4   Ps:  Probabilitas % Selection
5   Pm: Probabilitas % mutasi
6   Pc:  Probabilitas % Crossover
7   Th: Threshold (0,5)
8   Inisialisasi populasi awal (membangkitkan P(0))
9   While belum memenuhi kriteria berhenti
10        Evaluasi nilai fitness dan keragaman populasi
11              If keragaman populasi > threshold then
12                    Operasi fungsi seleksi populasi
13                    Operasi fungsi Crossover (persilangan)
14                    Operasi fungsi mutasi
15              Else
16                    Operasi fungsi dynamic artificial chromosomes
17                    Operasi fungsi seleksi
18                    Operasi fungsi Crossover
19                    Operasi fungsi Mutasi
20              P(t) = Elitism (P(t), A(t));
21              Gen(t) = Gen(t) + 1
22              End if
23              P(t) = nilai fitness terbaik
24              Jalankan klustering(P(t), data)
25  End While
```

```
26  Cetak P(t) dengan nilai fitness terbaik
27  Cetak DBI (Davies Bouldin Index)
28  Cetak SSE (Sum of Square Error)
```

2.5 Metode Evaluasi

2.5.1 Davies Bouldin index (DBI)

Metode evaluasi yang digunakan pada penelitian ini adalah *Davies-Boulden Index* untuk mengukur kualitas klastering pada algoritma K-Means[23]. Dengan menggunakan fungsi sebagai berikut:

$$DBI = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max \left(\frac{s_i + s_j}{d(c_i, c_j)} \right) \quad (9)$$

Dimana n adalah jumlah klaster yang dibentuk , Si dan Sj dalam klaster untuk ith dan jth dihitung rata-rata jarak dari input dalam setiap klaster untuk *centroid* (ci, cj)

2.5.2 Metode Sum Square Error (SSE)

Metode pengukuran tingkat error pada klastering adalah *Sum Square Error*

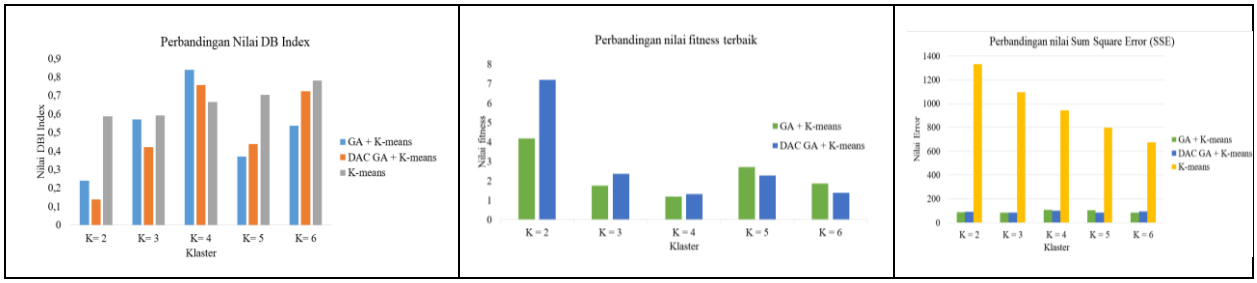
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d\,(x, \overline{x_{c_i}})^2 \quad (10)$$

3. HASIL DAN PEMBAHASAN

Tabel 1 Hasil perbandingan GA + K-Means dengan DAC GA + K-Means pada dataset beban kasus Tuberkulosis MDR (*Multidrug-resistant tuberculosis*)

K	K-MEANS		GA + K-MEANS			DAC GA + K-MEANS		
	DBI	SSE	DBI	Fitness	SSE	DBI	Fitness	SSE
2	0,586	1331,61	0,238	4,18	89,23	0,138	7,21	92,56
3	0,592	1095,12	0,569	1,75	84,36	0,421	2,37	83,47
4	0,664	943,71	0,839	1,19	108,57	0,756	1,32	101,29
5	0,704	797,11	0,369	2,70	103,13	0,438	2,28	83,12
6	0,780	675,07	0,536	1,86	85,74	0,721	1,38	94,53
$\bar{X}$	0,665	968,52	0,510	2,34	94,21	0,495	2,91	90,99

Tabel 1 merupakan matrik hasil perbandingan metode K-Means, GA + K-Means dan DAC GA + K-Means. 3 metode tersebut diuji pada dataset yang sama, jumlah klaster sama yaitu: 2 sampai dengan 6 dengan iterasi sebanyak 300 kali dan 30 individu yang digunakan pada metode GA + K-Means dan DAC GA + K-Means. Hasil menunjukkan bahwa nilai DBI, *Fitness* dan SEE pada metode DAC GA + K-Means lebih baik dibandingkan dengan 2 metode tersebut yaitu: nilai DBI sebesar 0,138, *Fitness* sebesar 7,21 dan 92,56. Jika dihitung secara rata-rata keseluruhan dari klaster 2 sampai dengan 6 maka metode DAC GA + K-Means masih tetap unggul dari 2 metode tersebut.

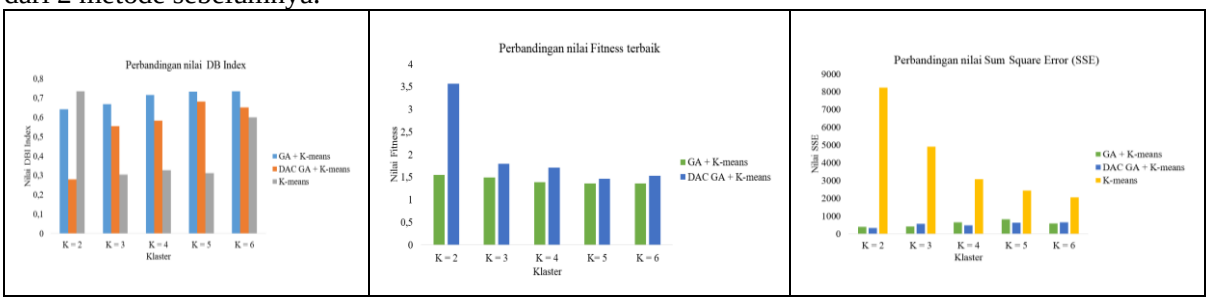


Gambar 6 Grafik hasil perbandingan nilai SSE antara GA + K-Means dengan DAC GA + K-Means

Tabel 2 Hasil perbandingan perhitungan menggunakan metode K-Means, GA + K-Means dan DAC GA + K-Means pada dataset konfirmasi baru kasus pengawasan resistant Tuberkulosis

K	K-MEANS		GA + K-MEANS			DAC_GA + K-MEANS		
	DBI	SSE	DBI	Fitness	SSE	DBI	Fitness	SSE
2	0,735	8243,06	0,641	1,55	397,29	0,279	3,57	332,39
3	0,305	4920,60	0,669	1,49	410,59	0,554	1,80	575,24
4	0,326	3075,09	0,716	1,39	642,52	0,583	1,71	478,49
5	0,311	2449,39	0,733	1,36	818,79	0,681	1,46	637,73
6	0,599	2065,13	0,734	1,36	587,97	0,652	1,53	642,05
$\bar{X}$	0,455	4150,65	0,6986	1,43	571,43	0,549	2,014	533,18

Tabel 2 merupakan matrik hasil perbandingan antara metode K-Means, GA + K-Means dan DAC GA + K-Means yang diujikan pada dataset kasus konfirmasi baru pada pengawasan resistan Tuberkulosis. Ujicoba dilakukan sebanyak 6 kali dengan 5 kali dengan nilai k yaitu: 2,3,4,5, dan 6. Hasil menunjukkan bahwa nilai DBI, *Fitness* dan SSE terbaik pada klaster 2 yaitu: 0,297, 3,57 dan 332,39. Hal ini menunjukkan bahwa metode DAC GA + K-Means lebih baik dari 2 metode sebelumnya.

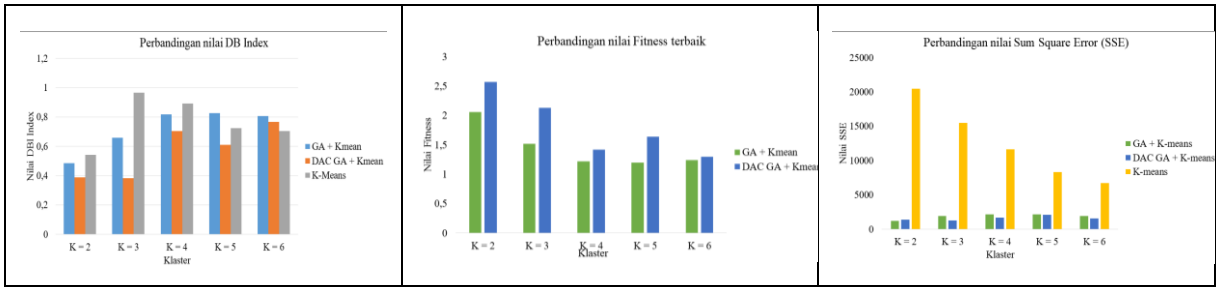


Gambar 7 Grafik hasil perbandingan nilai DBI pada dataset konfirmasi baru kasus pengawasan resistant Tuberkulosis

Tabel 3 Hasil perbandingan perhitungan menggunakan metode K-Means, GA + K-Means dan DAC GA + K-Means pada dataset faktor risiko penyakit Tuberkulosis.

K	K-MEANS		GA + K-MEANS			DAC GA + K-MEANS		
	DBI	SSE	DBI	Fitness	SSE	DBI	Fitness	SSE
2	0,542	20436,48	0,484	2,06	1203,77	0,389	2,57	1401,59
3	0,966	15462,83	0,657	1,52	1923,31	0,382	2,13	1280,68
4	0,891	11626,95	0,817	1,22	2188,06	0,704	1,42	1670,87
5	0,723	8300,86	0,827	1,20	2188,03	0,609	1,64	2122,19
6	0,705	6741,89	0,805	1,24	1919,42	0,767	1,30	1552,62
$\bar{X}$	0,765	12513,80	0,718	1,45	1884,52	0,570	1,81	1605,59

Tabel 3 merupakan matrik hasil perbandingan perhitungan menggunakan metode K-Means, GA + K-Means dan DAC GA + K-Means pada dataset faktor risiko penyakit Tuberkulosis. Matrik tersebut menyajikan nilai k sebanyak 2 hingga 6 dimana nilai k terbaik pada k sebanyak 2 dan 3 dengan nilai DBI adalah 0,389 dan 0,382, *Fitness* 2,57 dan 2,13 dan tingkat errornya sebesar 1401,59 dan 1280,68. Jika dirata-rata dari ketiga metode tersebut menunjukkan bahwa metode DAC GA + K-Means lebih baik dari K-Means dan GA + K-Means.



Gambar 8 Grafik hasil perbandingan nilai DBI pada dataset faktor risiko penyakit tuberculosis

Data hasil eksperimen menggunakan 3 jenis dataset yang berkaitan dengan penyakit Tuberkulosis adalah sebagai berikut:

1. Hasil perhitungan menggunakan dataset beban kasus Tuberkulosis MDR (*multidrug resistant Tuberkulosis*) terdapat 6 klaster yang telah diuji. Namun, dari ke enam klaster tersebut paling optimal ada pada klaster kedua dimana nilai DBI sebesar 0,138, nilai *fitness* terbaik sebesar 7,21 dan *Sum Square Error* 92,56 dengan menggunakan algoritma DAC GA + K-means. Sedangkan perhitungan menggunakan algoritma GA + K-Means dihasilkan nilai sebesar 0,238 dan nilai *fitness* sebesar 4,18. metode K-Means memperoleh nilai DBI sebesar 0,586 dan nilai tingkat error (SSE) sebesar 1331,61 secara keseluruhan nilai DAC GA + K-Means lebih unggul dibandingkan dengan perhitungan 2 metode tersebut.
2. Hasil perhitungan menggunakan dataset konfirmasi baru kasus pengawasan resistant Tuberkulosis. Ada 6 jumlah klaster yang diuji baik menggunakan metode GA + K-Means maupun DAC GA + K-means. Hasil menunjukkan bahwa jumlah klaster yang terbaik adalah k = 2. DAC GA + K-Means memperoleh Nilai DBI yang diperoleh adalah 0,279, *best fitness* sebesar 3,57 dan tingkat *error* sebesar 732,39 sedangkan GA + K-Means memperoleh nilai DBI sebesar 0,641, *best fitness* sebesar 1,55 dan tingkat errornya lebih rendah yaitu sebesar 397,29. metode K-Means memperoleh nilai DBI sebesar 0,735 dan tingkat error sebesar 8243,06. secara keseluruhan perhitungan DAC GA + K-Means lebih baik dibandingkan dengan 2 metode tersebut.
3. Hasil pengujian dengan dataset faktor risiko penyakit Tuberkulosis berdasarkan jenis kelamin, umur menunjukkan bahwa nilai k terbaik pada klaster 3 dimana nilai DBI sebesar 0,382, *best fitness* sebesar 2,13 dan tingkat errornya mencapai nilai 1280,68. hasil perhitungan tersebut diperoleh menggunakan metode DAC GA + K-Means sedangkan perhitungan menggunakan GA + K-Means memperoleh nilai DBI sebesar 0,657, fitness terbaik sebesar 1,52 dan tingkat *error* sebesar 1923,31. kemudian metode K-Means memperloleh nilai DBI sebesar 0,966 dan tingkat error sebesar 15462,83.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil dan pembahasan tersebut diatas dan untuk menjawab pertanyaan penelitian maka dapat disimpulkan bahwa metode DAC GA + K-Means lebih unggul dibandingkan dengan 2 metode lain yaitu K-Means dan GA + K-Means terhadap penentuan centorid awal pada algoritma K-Means untuk tuberculosis dataset. Hal ini dibuktikan dengan hasil perhitungan ke 3 dataset Tuberkulosis yaitu: dataset beban kasus Tuberkulosis MDR (*multidrug* resistant Tuberkulosis) dengan k terbaik adalah 2 dan nilai DBI sebesar 0,138, nilai fitness terbaik sebesar 7,21 dan *Sum of Square error* 92,56. dataset konfirmasi baru kasus pengawasan *resistant tuberculosis* k terbaik adalah 2 dan nilai DBI yang diperoleh adalah 0,279, *best fitness* sebesar 3,57 dan tingkat error sebesar 332,39. Dataset faktor risiko penyakit Tuberkulosis berdasarkan jenis kelamin, umur menunjukkan bahwa nilai k terbaik pada klaster 3 dan nilai DBI sebesar 0,382, *best fitness* sebesar 2,13 dan tingkat errornya mencapai nilai 1280,68.

4.2 Saran

Pada penelitian selanjutnya perlu adanya penerapan *feature selection* untuk mengurangi jumlah attribut pada dataset uji. Hal ini dimaksdunya untuk mempercepat komputasi dengan metode tersebut. Diharapkan dapat menghasilkan nilai DBI, *fitness* terbaik dan tingkat *error* yang lebih baik.

DAFTAR PUSTAKA

[1] P. Bhatia, "Introduction to Data Mining," *Data Min. Data Warehous.*, pp. 17–27, 2019, doi: 10.1017/9781108635592.003.

[2] Y. Fu, *Data mining*, vol. 16, no. 4. 1997.

[3] S. Agarwal, *Data mining: Data mining concepts and techniques*. 2014.

[4] J. Nayak, B. Naik, and H. S. Behera, "Computational Intelligence in Data Mining," vol. 711, 2019, doi: 10.1007/978-981-10-8055-5.

[5] J. Yadav and M. Sharma, "A Review of K-mean Algorithm," *Int. J. Eng. Trends Technol.*, vol. 4, no. 7, pp. 2972–2976, 2013.

[6] L. E. Agustín-Blas, S. Salcedo-Sanz, S. Jiménez-Fernández, L. Carro-Calvo, J. Del Ser, and J. A. Portilla-Figueras, "A new grouping genetic algorithm for clustering problems," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9695–9703, 2012, doi: 10.1016/j.eswa.2012.02.149.

[7] P. A. Ariawan, "Optimasi Pengelompokan Data Pada Metode K-means dengan Analisis Outlier," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 2, pp. 88–95, 2019, doi: 10.25077/teknosi.v5i2.2019.88-95.

[8] S. Bhatia, "New improved technique for initial cluster centers of K means clustering using Genetic Algorithm," *2014 Int. Conf. Converg. Technol. I2CT 2014*, pp. 1–4, 2014, doi: 10.1109/I2CT.2014.7092112.

[9] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, 2013, doi: 10.1016/j.eswa.2012.07.021.

[10] M. Erisoglu, N. Calis, and S. Sakallioglu, "A new algorithm for initial cluster centers in k-means algorithm," *Pattern Recognit. Lett.*, vol. 32, no. 14, pp. 1701–1705, 2011, doi: 10.1016/j.patrec.2011.07.011.

[11] T. P. Hong, C. H. Chen, and F. S. Lin, "Using group genetic algorithm to improve performance of attribute clustering," *Appl. Soft Comput. J.*, vol. 29, pp. 371–378, 2015, doi: 10.1016/j.asoc.2015.01.001.

[12] A. C. Jinyin *et al.*, "A Novel Cluster Center Fast Determination Clustering Algorithm," *Appl. Soft Comput. J.*, 2017, doi: 10.1016/j.asoc.2017.04.031.

[13] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, "Incremental genetic K-means algorithm and its application in gene expression data analysis," *BMC Bioinformatics*, vol. 5, pp. 1–10, 2004, doi: 10.1186/1471-2105-5-172.

[14] J. Qiao and Y. Lu, "A new algorithm for choosing initial cluster centers for k-means," no. Iccsee, pp. 527–530, 2013, doi: 10.2991/iccsee.2013.135.

[15] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and S. R. M. Zeebaree, "Combination of k-means clustering with genetic algorithm: A review," *Int. J. Appl. Eng. Res.*, vol. 12, no. 24, pp. 14238–14245, 2017.

[16] E. Utik Wahyuningtyas, R. Regasari Mardi Putri, and Sutrisno, "Optimasi K-Means Untuk Clustering Dosen Berdasarkan Kinerja Akademik Menggunakan Algoritme Genetika Paralel," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 8, pp. 2548–964, 2018, [Online]. Available: http://j-ptiik.ub.ac.id.

[17] M. A. Rahman and M. Z. Islam, "A hybrid clustering technique combining a novel genetic algorithm with K-Means," *Knowledge-Based Syst.*, vol. 71, pp. 345–365, 2014, doi: 10.1016/j.knosys.2014.08.011.

[18] M. R. Kamal, R. Satria, A. Syukur, F. I. Komputer, and U. D. Nuswantoro, "Integrasi Kromosom Buatan Dinamis untuk Memecahkan Masalah Konvergensi Prematur pada Algoritma Genetika untuk Traveling Salesman Problem," *J. Intell. Syst.*, vol. 1, no. 2, pp. 61–66, 2015.

[19] W. O. Health, "Tuberculosis (TB) World Health Organization," 2019. https://www.who.int/tb/en/ (accessed Jan. 01, 2019).

[20] J. B. MacQueen, "Some methods for classification and analysis of multivariate

observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[21] S. N. D. S.N. Sivanandam, *Introduction to Genetic Algorithms*. Springer-Verlag Berlin Heidelberg, 2008.

[22] P. C. Chang, W. H. Huang, and C. J. Ting, "Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 1863–1878, 2010, doi: 10.1016/j.eswa.2009.07.066.

[23] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, 1979, doi: 10.1109/TPAMI.1979.4766909.