
Visualisasi Serta Efektivitas *Colour Clustering* dengan *K-Means* dan *Bee Algorithm*

Ginda Maruli Andi S¹, Novianda²

¹Program Studi Informatika, ²Program Studi Teknik Industri, Fakultas Sains dan Teknologi,
Universitas Samudra

E-mail: ¹ginda.mas@unsam.ac.id, ²novianda_tif@unsam.ac.id

Abstrak

Clusterisasi citra digital merupakan dasar utama dalam analisis citra dan pengenalan pola yang dimungkinkan dalam proses segmentasi objek dalam gambar secara otomatis berdasarkan kemiripan warna, tekstur, atau intensitas. *Clusterisasi* umumnya dilakukan dengan algoritma *K-Means* karena kemampuannya yang sederhana dan efisien dalam mengelompokkan data, namun memiliki kelemahan seperti ketergantungan pada inisialisasi awal dan mudah terjebak pada solusi lokal, sedangkan *Bee Algorithm* menawarkan pendekatan optimasi yang lebih adaptif dan eksploratif sehingga dapat menghasilkan hasil *cluster* yang lebih optimal. Penelitian ini bertujuan untuk membandingkan kinerja algoritma *K-Means* dan *Bee Algorithm* (BA) dalam proses *clusterisasi* warna pada citra digital. Evaluasi dilakukan berdasarkan dua indikator utama, yaitu total jarak antar piksel ke pusat *cluster* dan sebaran kuantitas warna pada masing-masing *cluster*. Hasil eksperimen menunjukkan bahwa BA menghasilkan nilai jarak total yang lebih kecil dibandingkan *K-Means* pada semua jumlah *cluster* yang diuji. Nilai jarak yang lebih kecil menunjukkan bahwa BA mampu menemukan pusat *cluster* yang lebih optimal. Selain itu, distribusi kuantitas warna pada *cluster* hasil BA tampak lebih seimbang dibandingkan *K-Means*. Hal ini menunjukkan bahwa BA dapat memisahkan warna-warna dominan dengan lebih akurat. Berdasarkan hasil analisis dari gambar segmentasi dengan 2, 4, 8, dan 16 *cluster*, terlihat bahwa metode *Bee Algorithm* (BA) secara konsisten menghasilkan segmentasi warna yang lebih halus dan mendekati tampilan asli objek dibandingkan dengan metode *K-Means*. Dengan peningkatan jumlah *cluster*, kedua metode memang mengalami peningkatan kualitas visual, namun BA lebih unggul dalam mempertahankan detail halus seperti tekstur bulu dan gradasi warna alami. Sementara itu, *K-Means* cenderung menghasilkan pembagian warna yang lebih kaku dan berbentuk blok, yang mengurangi kealamian visual terutama pada jumlah *cluster* yang rendah.

Kata Kunci : *Bee Algorithm*, *Citra Digital*, *Clusterisasi*, *K-Means*, *Optimasi*

Abstract

Clustering of digital images is a key basis in image analysis and pattern recognition that enables automatic segmentation of objects in images based on color, texture, or intensity similarities. Clustering is generally done with the K-Means algorithm due to its simple and efficient ability to group data, but has disadvantages such as dependence on initial initialization and easily trapped in local solutions, while the Bee Algorithm offers a more adaptive and

explorative optimization approach so that it can produce more optimal Cluster results. This study aims to compare the performance of the K-Means and Bee Algorithm (BA) algorithms in the process of color Clustering in digital images. The evaluation is based on two main indicators, namely the total distance between pixels to the Cluster center and the distribution of color quantity in each Cluster. The experimental results show that BA produces a smaller total distance value than K-Means for all tested Clusters. The smaller distance value indicates that BA is able to find more optimal Cluster centers. In addition, the distribution of color quantity in the BA result Clusters appears more balanced than K-Means. This shows that BA can separate the dominant colors more accurately. Based on the analysis results of segmentation images with 2, 4, 8, and 16 Clusters, it can be seen that the Bee Algorithm (BA) method consistently produces smoother color segmentation that is closer to the original appearance of the object than the K-Means method. With an increase in the number of Clusters, both methods do experience an improvement in visual quality, but BA is superior in preserving fine details such as feather texture and natural color gradations. Meanwhile, K-Means tends to produce a more rigid and block-shaped color division, which reduces visual naturalness especially at low Cluster counts.

Keywords: Bee Algorithm, Digital Image, Clustering, K-Means, Optimization

1. PENDAHULUAN

Dalam era transformasi digital yang pesat, data visual seperti gambar dan video menjadi komponen utama dalam berbagai aplikasi teknologi informasi. Media sosial, pengawasan keamanan, dan sistem kecerdasan buatan memanfaatkan data citra untuk analisis yang kritical dan pengambilan keputusan yang presisi [1]. Pengolahan citra menjadi aspek penting dalam pengenalan objek, klasifikasi citra, dan kompresi gambar. Di sinilah teknik *colour Clustering* berperan, yaitu proses pengelompokan warna dalam gambar menjadi *cluster* yang merepresentasikan palet warna dominan. Teknik ini sangat bermanfaat untuk menyederhanakan representasi gambar sekaligus meningkatkan efisiensi dalam kompresi, pengenalan pola, dan segmentasi objek [2]. Selain itu, *colour Clustering* memungkinkan pengurangan kompleksitas data visual tanpa kehilangan informasi penting, sehingga mempercepat proses analisis citra. Teknik ini juga sering digunakan sebagai tahap pra-pemrosesan dalam algoritma deep learning, khususnya untuk mereduksi dimensi input citra dan memperkuat performa sistem klasifikasi visual. Dapat dikatakan bahwa *colour Clustering* menjadi fondasi penting dalam pengembangan sistem cerdas berbasis visual yang efisien dan akurat.

Algoritma *K-Means Clustering* adalah metode yang paling umum diterapkan karena kesederhanaan dan kecepatan konvergensinya. Prosesnya dimulai dengan pemilihan acak sejumlah titik awal sebagai pusat *cluster (centroid)*, di mana setiap data akan diklasifikasikan berdasarkan kedekatannya dengan pusat *cluster* menggunakan jarak *Euclidean* [3]. Namun, kelemahan utama dari *K-Means* terletak pada pemilihan acak pusat *cluster*. Ini dapat menyebabkan konvergensi ke solusi sub-optimal atau *local minimum*, yang berpotensi merugikan kualitas *Clustering* [4]. Selain itu, *K-Means* sensitif terhadap *outlier* dan distribusi data yang tidak seragam, yang dapat memengaruhi pembentukan *cluster* secara signifikan [5]. Algoritma ini juga mengharuskan pengguna menentukan jumlah *cluster (k)* di awal, yang tidak selalu mudah diperkirakan secara akurat. Untuk mengatasi keterbatasan ini, beberapa pendekatan pengoptimalan seperti algoritma genetika, *particle swarm optimization*, dan *bee*

algorithm telah dikembangkan. Tujuannya adalah untuk meningkatkan akurasi dan kestabilan hasil *clusterisasi* dengan menentukan posisi awal *centroid* yang lebih optimal.

Sebagai alternatif, metode *Bee Algorithm* menawarkan keunggulan dalam hal eksplorasi ruang solusi yang lebih luas. Terinspirasi oleh perilaku lebah madu dalam mencari sumber makanan, algoritma ini menggantikan pusat *cluster* yang dipilih secara acak dalam *K-Means* dengan pusat yang lebih optimal yang dihasilkan dari pencarian lebah [6]. Setiap “lebah” mewakili solusi kandidat yang terdiri dari titik pusat *cluster* warna dalam ruang *RGB*. Evaluasi kualitas dilakukan menggunakan fungsi seperti *Within-Cluster Sum of Squares (WCSS)*, yang membantu dalam menilai seberapa baik warna tersebut mewakili *piksel* dalam gambar [7]. Proses eksplorasi dan eksploitasi dilakukan melalui dua tahap utama: pencarian lokal di sekitar solusi terbaik dan pencarian *global* di wilayah acak lainnya. Pendekatan ini memungkinkan algoritma untuk menemukan solusi yang lebih *global* dan menghindari jebakan local minimum yang sering terjadi pada *K-Means* [7]. Selain itu, fleksibilitas parameter pada *Bee Algorithm* memungkinkan penyesuaian strategi pencarian sesuai kompleksitas data citra [6]. Oleh karena itu, algoritma ini sangat potensial untuk meningkatkan performa *colour Clustering* dalam aplikasi pengolahan citra digital [8].

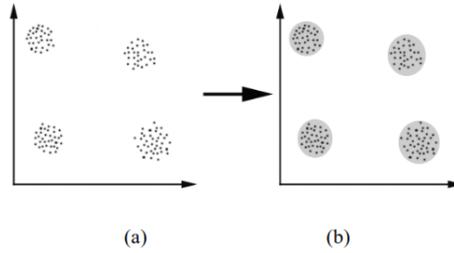
Kombinasi teknik eksplorasi *global* dan eksploitasi lokal yang diaplikasikan dalam *Bee Algorithm* membuat algoritma ini lebih tangguh dalam menghindari jebakan solusi lokal [5]. Setelah melalui siklus iterasi, pusat-pusat warna optimal dapat diperoleh, yang sangat penting dalam memaksimalkan proses pengolahan citra digital [9]. Hasil dari *colour Clustering* yang akurat dapat meningkatkan kualitas *output* dalam aplikasi seperti pengenalan pola, *visualisasi* medis, dan pencarian gambar otomatis, yang semuanya membutuhkan tingkat akurasi yang tinggi untuk efektivitas optimal [10]. Pendekatan ini juga dapat membantu mempercepat proses segmentasi objek dalam gambar yang kompleks, seperti citra medis atau citra satelit. Selain itu, algoritma ini mampu beradaptasi dengan data berdimensi tinggi, yang seringkali menjadi kendala dalam metode konvensional [11]. Keunggulan ini menjadikan *Bee Algorithm* relevan untuk diterapkan dalam sistem real-time yang membutuhkan respons cepat dan akurat [12]. Dengan karakteristik tersebut, *Bee Algorithm* mampu menyumbangkan kontribusi signifikan dalam pengembangan teknologi visual berbasis AI.

Penggunaan *Bee Algorithm* diharapkan mengatasi kelemahan metode konvensional seperti *K-Means*, memberikan hasil *Clustering* warna yang lebih stabil, optimal, dan adaptif pada berbagai aplikasi pengolahan citra digital yang kompleks [13], [14]. Ini menegaskan pentingnya evolusi teknik dalam mengelola data visual yang semakin berkembang di era informasi yang bertenaga teknologi saat ini. Seiring meningkatnya volume dan kompleksitas data visual, diperlukan algoritma yang tidak hanya akurat, tetapi juga efisien dalam waktu komputasi [15]. *Bee Algorithm* menjawab kebutuhan ini dengan pendekatan berbasis *metaheuristik* yang fleksibel dan dapat disesuaikan dengan berbagai jenis citra. Penelitian lanjutan di bidang ini sangat potensial untuk menghasilkan solusi visual yang lebih presisi dalam konteks medis, industri kreatif, dan otomatisasi industri. Oleh karena itu, *Bee Algorithm* bukan sekadar alternatif, tetapi diharapkan melalui studi ini mampu menjadi fondasi penting dalam era pengolahan citra generasi berikutnya.

2. TINJAUAN TEORITIS

Clusterisasi secara sederhana adalah proses mengorganisasi objek menjadi kelompok-kelompok yang anggotanya sama dalam beberapa hal. Jadi *Cluster* adalah kumpulan objek yang

mempunyai kesamaan tertentu pada suatu *Cluster* dan mempunyai perbedaan dengan objek pada *Cluster* lainnya.



Gambar 1 Ilustrasi *Clustering* Data

Pada Gambar a, dengan mudah dapat diidentifikasi empat buah *Cluster* dengan pembagian data pada Gambar b. Pada teknik *clusterisasi*, kriteria pengelompokan data berdasarkan jarak. Dua objek atau lebih terdapat pada *Cluster* yang sama jika objek-objek tersebut berdekatan menurut ukuran jarak (geometri) [16]. Metode *Clustering* yang paling populer adalah *K-Means* disebut juga metode *Lloyd-Forgy*[15]. Diberikan N objek, yang akan dialokasikan ke setiap objek ke salah satu “ k ” *Cluster* dengan suatu pusat, sehingga diperoleh minimum jumlah jarak data ke titik pusat *Cluster* yaitu;

$$J(w, z) = \sum_{i=1}^N \sum_{j=1}^K W_{ij} \|x_i - z_j\|^2 \quad (1)$$

Di mana

K = Banyak *Cluster*

N = Banyak Objek

x_i = Lokasi objek ke- i , $i \in [1, 2, \dots, N]$, $x_i \in R^n$

z_j = Pusat *Cluster* ke- j , $j \in [1, 2, \dots, K]$, $z_j \in R^n$

K-Means Clustering, Algoritma melakukan *Clustering* dengan langkah sebagai berikut;

1. Pilih K buah titik *centroid* secara acak
2. Kelompokkan data sehingga terbentuk K buah *Cluster* dengan titik *centroid* dari setiap *Cluster*.
3. Perbaharui nilai titik *centroid*
4. Ulangi langkah 2 dan 3 sampai nilai titik *centroid* tidak berubah

Proses pengelompokan data ke dalam suatu *Cluster* dapat dilakukan dengan cara menghitung jarak terdekat dari suatu data ke sebuah titik *centroid*. Perhitungan jarak Minkowski dapat digunakan untuk menghitung jarak antar 2 buah data. Rumus untuk menghitung jarak tersebut adalah;

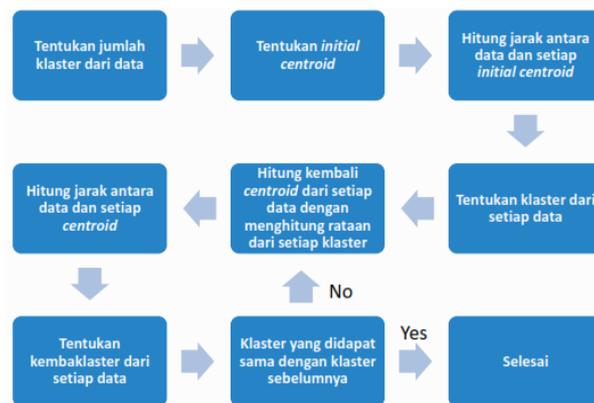
$$d(x_i, x_j) = \left(|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \dots + |x_{ip} - x_{jp}|^g \right)^{\frac{1}{g}} \quad (2)$$

Di mana:

g = koefisien pangkat,

x_i, x_j = data yang akan dihitung jaraknya

p = dimensi data



Gambar 2 Bagan langkah *Clustering* dengan *K-Means*

Bee Algoritma pada *Clustering* dilakukan oleh Populasi *scout bee* disebar sebagai titik-titik dalam data, selanjutnya di evaluasi hingga memperoleh jarak minimum. Setiap *Bee* menunjukkan potensial sebagai titik pusat dari tiap-tiap *cluster* acak. Akan dipilih populasi sebanyak *cluster* (K), yang memiliki nilai optimum. Selanjutnya, di bangkitkan populasi baru untuk mengeksplorasi daerah *elite site* (tiap *Cluster*) dan di evaluasi hingga ditemukan nilai paling optimum. Langkah-langkah *bee Algoritma* dalam komputasi akan ditunjukkan melalui *pseudocode* berikut;

Membangkitkan populasi	[1] <i>ScoutBee</i> (<i>nScout</i> , <i>nVar</i>);
Kondisi Pemberhentian	[2] <i>while</i> (<i>iterasi</i> ≤ <i>Max_{iter}</i> & <i>Error</i> ≤ ε)
Evaluasi Site	[3] <i>EliteSite</i> = <i>memuat nilai fungsi terbaik</i>
Bee Terbaik	[4] Pilih Bee Terbaik di setiap elite site
Eksplorasi	[6] Eksplorasi <i>EliteSite</i>
	OtherBee = <i>Waggledance</i> (BeeTerbaik)
	Bee Terbaik Pada elite site = <i>BeeBest</i> (<i>i</i>)
	[7] Eksplorasi Tetangga <i>Elitesite</i>
	Tetangga = [<i>Elitesite</i> +1: <i>Elitesite</i>]
	OtherBee = <i>Waggledance</i> (BeeTerbaik)
	Jika <i>BeeLain</i> < <i>Bee Best</i>
	Maka <i>BeeBest</i> (<i>i</i>) = <i>BeeLain</i>
Loop	[8] Kembali Langkah-6
	<i>i</i> = <i>i</i> +1;
	Nilai(<i>BeeBest</i> (<i>i</i>))<Nilai(<i>BeeBest</i> (<i>i</i> -1))
	Maka <i>BeeBest</i> (<i>i</i>)= <i>BeeBest</i> (<i>i</i>)
	Nilai(<i>BeeBest</i> (<i>i</i>))>Nilai(<i>BeeBest</i> (<i>i</i> -1))
	Maka <i>BeeBest</i> (<i>i</i>)= <i>BeeBest</i> (<i>i</i> -1)
	Output : Bee Terbaik, Nilai Solusi Terbaik
Evaluasi perubahan nilai	[9] setiap 20 iterasi : Kriteria Cauchi
	error= norm(nilai(<i>i</i>):nilai(<i>i</i>+20))

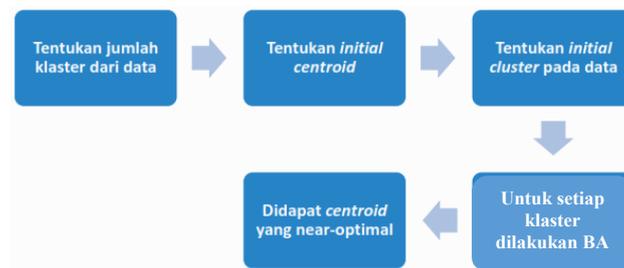
Gambar 3 *Pseudocode Bee Algoritma* [5]

Pada algoritma digunakan parameter yang ditunjukkan pada tabel 1:

Tabel 1 Parameter awal *bee algorithm*

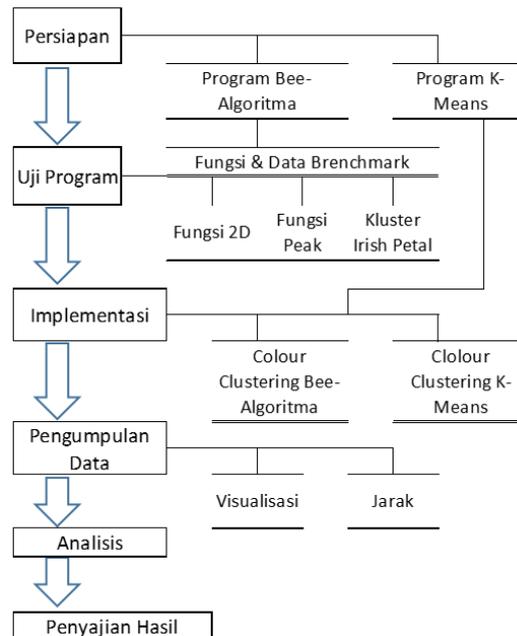
Parameter	Simbol / Notasi	Nilai / Rumus	Keterangan
Banyak Variabel	nVar	-	Jumlah variabel
Populasi Scout Bee	nScout	-	Jumlah lebah pencari
Calon Site	nSites	-	Jumlah lokasi
Elite Site	-	$\frac{1}{2} \times nSites$	Jumlah site elit
Other Bees	-	$\text{round}(0.3 \times nScout)$	Lebah yang dikirim
Epsilon	ϵ	1×10^{-10}	Ambang toleransi
Iterasi Maksimum	Max_iter	-	Batas maksimum iterasi
Redaman	rdamp	0.95	Faktor redaman
Batas Variabel	[Varmin : VarMax]	-	Rentang nilai variabel

Langkah dalam menentukan *Cluster* pada algoritma *bee* sebagai berikut:

Gambar 4 Bagan langkah *Cluster* dengan *Bee Algorithm*

2. METODE

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komputasional. Tujuan dari penelitian ini adalah untuk membandingkan efektivitas *clusterisasi* warna menggunakan algoritma *K-Means* dan *Bee Algorithm* melalui evaluasi terhadap tingkat ketepatan konvergensi, kecepatan proses, jumlah iterasi, serta visualisasi hasil *clusterisasi* warna. Tahapan penelitian dilakukan dengan diagram alur berikut:



Gambar 5 Alur Penelitian

2.1. Persiapan

Tahap persiapan dilakukan dengan menyiapkan perangkat keras berupa PC atau laptop dengan spesifikasi memadai. Pada penelitian menggunakan perangkat komputer dengan *Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz-2.81 GHz, Ram 8 Gb*, sistem operasi Windows 10. *Compile* program menggunakan aplikasi pemrograman MATLAB. Data yang digunakan berupa gambar digital berwarna dalam format *JPEG* yang diperoleh dari *dataset* publik (*Google Images*). Pemilihan gambar mempertimbangkan variasi warna yang kompleks untuk menguji kemampuan algoritma dalam memisahkan *cluster* warna dominan[14]. Gambar tersebut kemudian diuji dengan menjalankan skrip sederhana untuk memastikan dapat terbaca dengan baik oleh aplikasi yang digunakan. Tahapan ini menjadi fondasi penting sebelum melanjutkan ke proses utama seperti *clusterisasi* warna, segmentasi, atau analisis citra lainnya.

2.2. Uji Program

Bee algoritma (BA) akan di uji untuk fungsi *bervariabel* tunggal, dan *multi-variable*, untuk menunjukkan bahwa algoritma tidak terjebak pada lokal minimum, dan memperoleh solusi sesuai performanya.

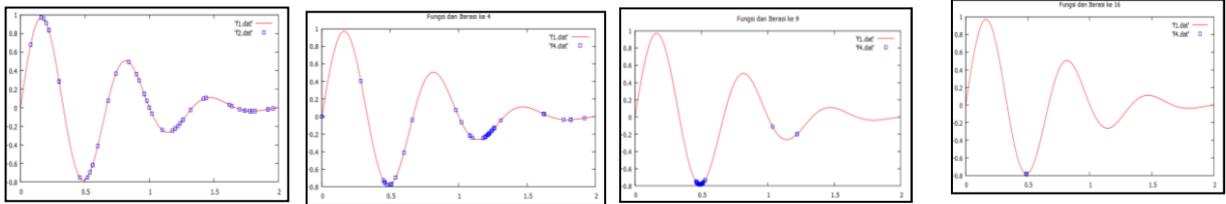
a. Fungsi *Variable* Tunggal: *Bee Algoritma* diuji dengan fungsi brenchmark 2 Dimensi;

$$f(x) = \exp(-x)^2 \sin(3\pi x) \tag{3}$$

Tabel 2 Parameter fungsi 2 dimensi

ScoutBee	Batas		Max_Iter	Eps
	Atas	Bawah		
50	-5	5	1.000	e-6

Diperoleh visualisasi persebaran scout bee;



Sebaran Populasi Awal

Iterasi : 4

Iterasi : 9

Iterasi : 16

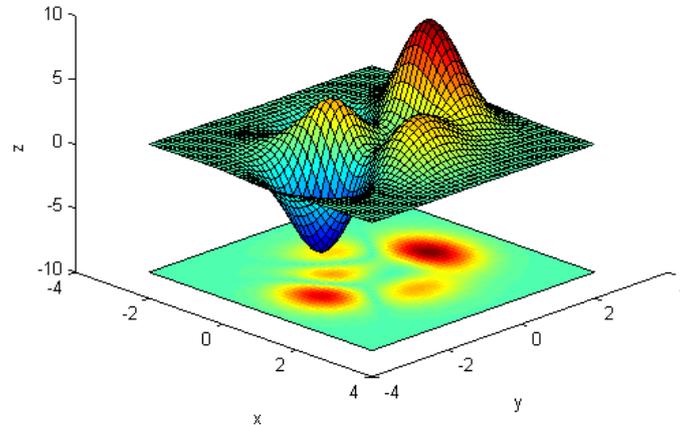
Gambar 6 Sebaran Bee Menemukan Global Minimum Fungsi 2 Variable

Terlihat bahwa fungsi tersebut memiliki 3 lokal minimum. Pada *iterasi* pertama lebah-lebah pencari menyebar kesegala arah, selanjutnya lebah berkumpul di suatu *local minimum*. Pada *iterasi* ke-3 terlihat lebah berkumpul di 3 lokal minimum, dengan jumlah lebah yang berbeda-beda, tergantung nilai $f(x)$ -nya. Nilai fungsi terendah memiliki kumpulan bee paling banyak, dan semakin berdekatan. Hingga diperoleh titik (Bee Terbaik) sebagai solusi sebagai berikut:

Tabel 3 Nilai *global minimum* fungsi 2 variabel

$y(x)$	x	Iterasi	Waktu(s)
-0,782902	0,489139	18	4,72

b. **Fungsi 3 Variable:** Bee Algoritma diuji melalui Peak Function



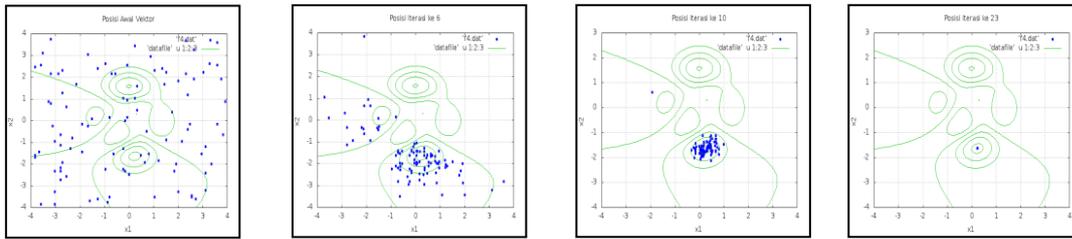
Gambar 7 Fungsi Peak

$$f(x_1, x_2) = 3(1 - x_1)^2 \cdot \exp(-x_1^2 - (x_2 + 1)^2) - 10 \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) \cdot \exp(-x_1^2 - x_2^2) - \frac{1}{3} \cdot \exp(-(x_1 + 1)^2 - x_2^2) \tag{4}$$

Tabel 4 Parameter fungsi peak

ScoutBee	Batas		Max_Iter	Eps
	Atas	Bawah		
100	-4	4	1000	e-10

Diperoleh persebaran bee;



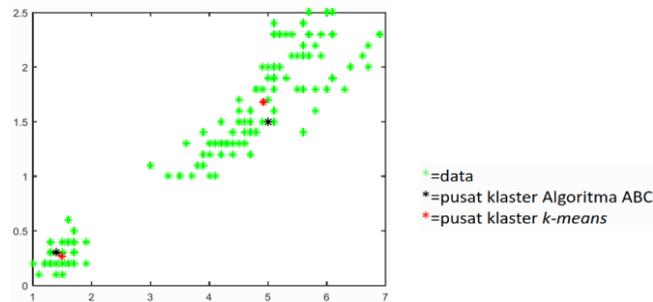
Gambar 8 Sebaran *Bee* Menemukan *Global Minimum*

Peak merupakan fungsi jika pada rentang [-4:4] memiliki 2 lokal minimum, dan 3 lokal maximum. Akan ditunjukkan cara *bee* memperoleh nilai solusi. Pada *iterasi* pertama, *bee* menyebar segala arah, selanjutnya akan berkumpul di tiap *local minimum*. Karena hanya ada 1 lokal minimum yang *extrem minimum*, maka posisi tersebut menjadi titik kumpul utama lebah. Hingga pada *iterasi* ke 23 setiap lebah berkumpul pada 1 titik sebagai solusi. Pada *iterasi* ke 24 diperlukan waktu 67.22 s untuk memperoleh nilai solusi;

Tabel 5 Solusi fungsi *Peak*

$F(x1,x2)$	<i>Variable</i>		<i>Iterasi</i>	Estimasi Waktu
	$x1$	$X2$		
-6,5511	0,2283	-1,6255	24	67,22s

c. **Validasi *Bee Algoritma* Vs *K-Means* pada *Clustering*:** BA dan *K-means* diuji dengan data *fisher irish* yang akan dikelompokkan menjadi 2 *cluster*, diperoleh hasil sebagai berikut;



Gambar 9 Posisi Pusat *Cluster K-Means* Vs BA

Dari gambar 9, persebaran sebanyak 150 data yang *dichusterisasi* kedalam 2 kelompok, terlihat perbedaan posisi pusat *cluster* antara BA dan *K-Means* sehingga dipastikan terdapat perubahan jumlah anggota. Hasil dari *running* program diperoleh:

Tabel 6 Hasil *Clustering* dengan *K-means* dan BA

Metode	Jarak	Banyak Data Tiap Cluster		Ketepatan Konvergensi	Kecepatan (detik)	iterasi
		K1	K2			
<i>K-Means</i>	1.288,7876	53	97	4,21e-7	1,02	12
<i>Bee Algoritma</i>	1.273,8769	52	98	3,23e-10	2,67	28

Dari hasil *running* program, pengujian *Bee Algoritma* dan *K-means* dalam

mengelompokkan data kedalam dua *cluster*. Hasil uji menunjukkan *bee algoritma* dapat digunakan sebagai metode *clusterisasi*, dengan jarak yang lebih rendah dibandingkan K-means, dengan selisih 14.9107 serta terjadi perpindahan anggota antar *cluster* sebanyak 1 data.

3. HASIL DAN PEMBAHASAN

Teknik *Clustering* dapat digunakan untuk men-kuantitas warna, banyaknya warna yang dimunculkan berdasarkan banyaknya *cluster*. Warna yang akan dipilih berdasarkan rata-rata masing-masing kelompok warna. Berikut gambar digital yang akan di*clusterisasi*;

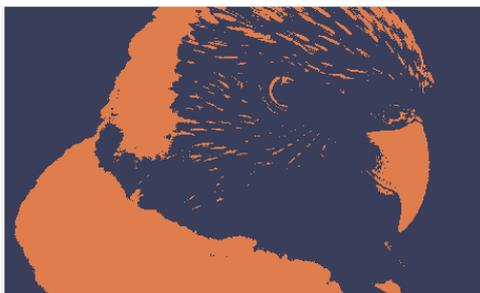
a. 2 Cluster

Terlihat hasil jarak antar *cluster* dan pusat dari 2 *clusterisasi* warna:

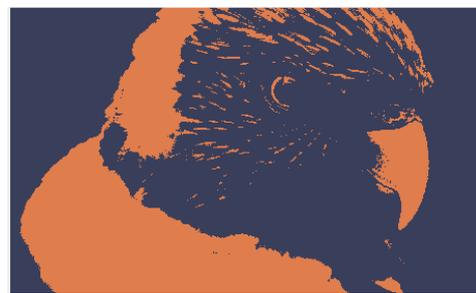
Tabel 7 Hasil jarak 2 *clusterisasi*

Metode	Jarak	Banyak Data Tiap Cluster		Ketepatan Konvergensi	Kecepatan iterasi (detik)	iterasi
		K1	K2			
<i>K-Means</i>	228.050,179526	737.450	1.566.550	4,36e-6	1,15	14
<i>Bee Algoritma</i>	227.862,224470	1.592.141	711.859	2,11e-10	2,93	30

Tabel 7 menunjukkan *clusterisasi* antara algoritma K-Means dan Bee Algorithm (BA) memiliki beberapa implikasi berupa: nilai jarak total yang dihasilkan BA (227.862,22) sedikit lebih rendah dibanding K-Means (228.050,18), yang menunjukkan bahwa BA mampu menghasilkan *cluster* dengan tingkat kedekatan antar data yang lebih optimal. Ini mengindikasikan bahwa BA memiliki keunggulan dalam pencarian solusi global yang lebih baik, sehingga lebih efektif dalam meminimalkan jarak antar data dan pusat *cluster*. Selain itu, distribusi kuantitas warna yang terbentuk pada masing-masing *cluster* menunjukkan karakteristik berbeda. K-Means menghasilkan *cluster* yang tidak seimbang dengan jumlah warna lebih sedikit di C1 dan lebih banyak di C2, sedangkan BA membentuk pembagian yang berbanding terbalik, dengan C1 lebih besar dan C2 lebih kecil. Perbedaan ini menunjukkan bahwa BA dapat menghasilkan pembagian warna yang lebih merata dan potensial memberikan hasil visual yang lebih seimbang dalam konteks pengolahan citra. Hasil citra ditunjukkan pada gambar berikut:



K-Means



bee algoritma

Gambar 10 Kuantitas warna 2 *cluster*

Gambar 10 tersebut menunjukkan hasil segmentasi citra menggunakan dua metode dengan pendekatan berbeda, yakni *K-Means* dan BA, masing-masing dengan dua *cluster* warna.

Keduanya menghasilkan visualisasi gambar burung dengan dominasi dua warna utama, yaitu oranye dan biru tua. Penggunaan dua warna ini mencerminkan proses segmentasi yang membagi area citra berdasarkan kesamaan warna atau fitur visual lainnya. Secara umum, kedua metode memiliki hasil yang cukup mirip dari segi bentuk dan struktur objek. Burung tetap dikenali dengan jelas pada kedua gambar, menunjukkan bahwa metode segmentasi bekerja efektif dalam mempertahankan informasi bentuk *global* dari objek. Persamaan lainnya adalah pada pilihan warna yang digunakan, di mana kedua metode mengelompokkan piksel menjadi dua warna dominan yang sama, menunjukkan konsistensi hasil segmentasi dalam pengelompokan warna utama.

Namun, terdapat perbedaan mencolok dalam distribusi dan ketajaman batas *cluster* warna. Pada hasil *K-Means*, batas antar *cluster* cenderung lebih tegas dan tersegmentasi secara kasar, terutama pada area peralihan warna seperti bagian kepala dan paruh. Sebaliknya, hasil dari BA terlihat lebih halus dan adaptif dalam menangkap perubahan tekstur, seperti bulu halus di sekitar kepala, sehingga memberikan kesan segmentasi yang lebih alami dan detail. Detail tekstur menjadi keunggulan utama metode BA. Algoritma ini tampaknya lebih sensitif terhadap variasi lokal dalam citra, memungkinkan penggambaran pola-pola kecil yang tidak terlalu terlihat pada hasil *K-Means*. Hal ini tampak dari keberhasilan BA dalam mempertahankan tekstur halus di area mata dan bagian atas kepala burung. Sementara itu, hasil dari *K-Means* cenderung menyederhanakan informasi visual yang halus menjadi bentuk yang lebih bloky atau datar. Secara keseluruhan, kedua metode memiliki kelebihan masing-masing. *K-Means* cocok digunakan untuk segmentasi cepat dengan batas *cluster* yang sederhana dan jelas. Di sisi lain, BA menawarkan segmentasi yang lebih rinci dan realistis, cocok untuk aplikasi yang membutuhkan pemrosesan tekstur dan detail lebih tinggi. Pilihan metode terbaik sangat tergantung pada kebutuhan analisis dan kompleksitas citra yang digunakan.

b. 4 Cluster

Berdasarkan *running program* yang memberikan hasil jarak antar *cluster* dan pusat dari 4 *clusterisasi* warna:

Tabel 8 Hasil jarak 4 *clusterisasi*

	<i>K-means</i>	BA
Jarak	94.783,7521	81.522,8767
C1	432.510	661.001
C2	233.781	381.434
C3	1.234.137	661.856
C4	233.781	566.709
Ketepatan Konvergensi	3,98e-7	1,89e-10
Kecepatan (detik)	1,35	3,21
Iterasi	16	35

Tabel 8 menunjukkan perbandingan hasil *Clustering* antara algoritma *K-Means* dan *Bee Algorithm* (BA) dengan pembagian data ke dalam empat *cluster*. Berdasarkan data yang ditampilkan, terdapat perbedaan pada nilai jarak total, di mana *K-Means* menghasilkan jarak

sebesar 94783.75, sedangkan BA memiliki jarak yang lebih rendah, yaitu 81522.88. Ini menunjukkan bahwa BA lebih efektif dalam meminimalkan jarak antar titik dalam *cluster*, yang mengindikasikan performa yang lebih baik dalam hal optimasi distribusi data antar *cluster*. Melihat distribusi kuantitas warna pada masing-masing *cluster*, *K-Means* dan BA menunjukkan pola yang berbeda. Pada *K-Means*, *cluster* pertama (C1) memiliki nilai 432510, *cluster* kedua (C2) 233781, *cluster* ketiga (C3) 1234137, dan *cluster* keempat (C4) 233781. Sebaliknya, BA menunjukkan nilai yang lebih besar pada *cluster* pertama (C1 = 661001), *cluster* kedua (C2 = 381434), *cluster* ketiga (C3 = 661856), dan *cluster* keempat (C4 = 566709). Ini menunjukkan bahwa BA menghasilkan distribusi kuantitas warna yang lebih merata antar *cluster*, dengan perbedaan yang lebih kecil antara masing-masing *cluster* dibandingkan dengan *K-Means*.

Dari hasil ini, dapat disimpulkan bahwa meskipun *K-Means* berhasil mengelompokkan data ke dalam empat *cluster*, BA cenderung memberikan pembagian data yang lebih seimbang antar *cluster*, baik dari segi jarak maupun kuantitas warna. Dengan jarak yang lebih kecil dan distribusi kuantitas warna yang lebih merata, BA dapat dianggap lebih optimal dalam membentuk *cluster* yang lebih representatif dan terstruktur dengan baik, yang dapat memberikan hasil yang lebih akurat dalam analisis data yang membutuhkan pembagian data yang lebih proporsional. Hasil *clusterisasi* dalam bentuk visual gambar adalah sebagai berikut:



Gambar 11 Kuantitas warna 4 *cluster* *K-Means* (Kiri), dan BA (Kanan)

Gambar 11 menampilkan hasil segmentasi citra burung dengan menggunakan empat *cluster* warna melalui dua metode yang berbeda. Keduanya menghasilkan visualisasi dengan warna dominan yang serupa, yaitu kombinasi antara biru, oranye, kuning, dan abu-abu/hitam. Penggunaan empat *cluster* memungkinkan algoritma untuk menangkap lebih banyak detail visual dibanding segmentasi dengan dua *cluster* sebelumnya, menghasilkan pemisahan warna yang lebih kompleks dan deskriptif pada objek. Secara umum, kedua metode mempertahankan struktur bentuk utama burung, seperti kepala, mata, paruh, dan bulu, dengan cukup baik. Warna biru secara konsisten digunakan pada area bulu di kepala, sedangkan oranye pada paruh dan tubuh bagian bawah. Hal ini menunjukkan bahwa baik *K-Means* maupun BA mampu mengidentifikasi area dengan intensitas warna berbeda secara efektif berdasarkan distribusi piksel di dalam citra.

Perbedaan utama terletak pada tingkat kedetailan dan distribusi warna hasil segmentasi. Pada metode *K-Means* (kiri), batas antar warna terlihat lebih jelas dan kaku, serta beberapa warna seperti kuning muncul dalam bercak-bercak kecil di area tubuh, memberi kesan sedikit acak. Sebaliknya, hasil segmentasi menggunakan BA (kanan) terlihat lebih menyatu dan realistis, di mana transisi antar warna lebih halus dan tidak terlalu terfragmentasi. Area latar belakang juga tampak lebih alami dalam segmentasi BA karena tidak terlalu dipenuhi variasi

warna yang tidak perlu. Kelebihan BA dalam menangkap tekstur dan pola halus kembali terlihat dalam hasil ini. Misalnya, bagian mata dan area bulu di sekitar kepala burung memiliki detail yang lebih halus dan kontras di metode BA, yang menjadikan gambar tampak lebih hidup. Sementara itu, *K-Means* cenderung menyederhanakan detail tersebut menjadi blok warna besar, yang mengurangi ketajaman visual dan nuansa tekstur dari objek burung.

Secara keseluruhan, meskipun kedua metode mampu melakukan segmentasi warna dengan jumlah *cluster* yang sama, BA menunjukkan keunggulan dalam hal akurasi dan kealamian hasil visual. *K-Means* lebih cocok untuk segmentasi cepat dan sederhana, sedangkan BA lebih efektif dalam menjaga detail halus dan gradasi warna alami, menjadikannya lebih cocok untuk aplikasi yang memerlukan ketelitian visual tinggi, seperti pengenalan objek atau klasifikasi citra berbasis tekstur.

c. 8 Cluster

Berdasarkan perhitungan jarak antar *cluster* dan pusat, menggunakan kedua algoritma, dari 8 *clusterisasi* warna, diperoleh hasil berikut:

Tabel 9 Hasil jarak 8 *clusterisasi*

	<i>K-means</i>	BA
Jarak	33.013,092864	33.011,314401
C1	329.094	381.588
C2	598.977	325.407
C3	135.813	390.077
C4	109.215	206.986
C5	222.443	256.277
C6	221.088	361.379
C7	204.651	136.653
C8	482.718	245.633
Ketepatan Konvergensi	1,18e-6	1,74e-10
Kecepatan (detik)	1,89	3,84
Iterasi	18	42

Tabel 9 menyajikan hasil *Clustering* dengan delapan *cluster* yang menggunakan algoritma *K-Means* dan *Bee Algorithm (BA)*. Dalam hal jarak, terdapat sedikit perbedaan antara kedua algoritma, di mana *K-Means* menghasilkan jarak total sebesar 33.013,09, sedangkan BA sedikit lebih kecil dengan jarak 33011.31. Meskipun selisihnya sangat kecil, perbedaan ini menunjukkan bahwa BA sedikit lebih efisien dalam meminimalkan jarak antar titik dalam *cluster*, meskipun perbedaannya tidak signifikan. Melihat distribusi kuantitas warna pada setiap *cluster*, terlihat bahwa *K-Means* dan BA memiliki pola distribusi yang cukup berbeda. Pada *K-Means*, *cluster* pertama (C1) memiliki nilai 329094, *cluster* kedua (C2) 598.977, *cluster* ketiga (C3) 135813, *cluster* keempat (C4) 109215, *cluster* kelima (C5) 222443, *cluster* keenam (C6) 221088, *cluster* ketujuh (C7) 204.651, dan *cluster* kedelapan (C8) 482.718. Sementara itu, BA menunjukkan nilai yang berbeda pada masing-masing *cluster*: C1 = 381.588, C2 = 325.407, C3 = 390.077, C4 = 206.986, C5 = 256.277, C6 = 361379, C7 = 136653, dan C8 = 245.633.

Dari perbandingan ini, BA menunjukkan distribusi kuantitas warna yang lebih merata di antara *cluster-cluster* yang ada. Khususnya, BA menghasilkan nilai yang lebih seimbang pada *cluster* pertama hingga ketujuh, sedangkan *K-Means* cenderung memiliki ketidakseimbangan, seperti pada *cluster* kedua (C2) yang memiliki nilai jauh lebih tinggi dibandingkan *cluster-cluster* lainnya. Secara keseluruhan, meskipun jarak total antar kedua algoritma hampir sama, BA memberikan pembagian data yang lebih seimbang dan terstruktur dengan baik, yang menunjukkan bahwa BA mungkin lebih efektif dalam menjaga keseragaman dalam pembentukan *cluster* dibandingkan dengan *K-Means* pada jumlah *cluster* yang lebih besar. Perbedaan juga terlihat pada hasil visualisasi berikut:



Gambar 12 Kuantitas warna 8 *cluster K-Means* (Kiri), dan BA (Kanan)

Pada gambar 12, segmentasi warna menggunakan delapan *cluster* menghasilkan representasi warna yang jauh lebih rinci dibandingkan dengan jumlah *cluster* sebelumnya. Kedua metode berhasil memisahkan lebih banyak variasi warna, khususnya pada bagian bulu burung yang kini memperlihatkan lebih banyak gradasi dan tekstur. Warna-warna seperti biru, merah, kuning, hijau, dan abu-abu muncul dengan intensitas yang berbeda, menciptakan detail visual yang lebih kaya. Kedua metode, baik *K-Means* maupun BA, sama-sama mampu mempertahankan bentuk asli burung, termasuk detail pada area mata, paruh, dan tekstur bulu di kepala dan tubuh. Namun, tampak bahwa hasil segmentasi menggunakan BA (kanan) menunjukkan transisi warna yang lebih lembut dan realistis, sedangkan *K-Means* (kiri) menghasilkan batasan warna yang lebih tegas dan terfragmentasi. Ini memperlihatkan bahwa BA lebih efektif dalam menangkap perubahan halus pada intensitas warna di citra.

Perbedaan signifikan terlihat pada area transisi antara bulu biru dan latar belakang, serta antara paruh dan tubuh. Pada *K-Means*, perbedaan antar warna terasa cukup kasar, menyebabkan sebagian detail visual hilang atau terlihat pecah. Sedangkan pada hasil BA, peralihan antar *cluster* warna terlihat lebih alami dan membaur, menghasilkan tekstur yang tampak lebih nyata dan lembut, khususnya di bagian bulu dan latar belakang. Dari segi kejelasan objek, metode BA tampak lebih unggul dalam mempertahankan ketajaman di area-area penting seperti mata dan kontur paruh burung. Pada gambar *K-Means*, sebagian kecil detail di sekitar mata dan perbatasan paruh tampak lebih kabur karena batas warna yang terlalu keras. Dengan penggunaan delapan *cluster*, BA mampu membedakan antara detail halus dalam tekstur bulu tanpa mengorbankan bentuk keseluruhan objek.

Secara keseluruhan, meskipun kedua metode menunjukkan kemampuan segmentasi yang lebih baik dengan delapan *cluster* dibandingkan jumlah *cluster* yang lebih sedikit, hasil dari BA memberikan kualitas visual yang lebih baik dalam hal kontinuitas warna, kehalusan tekstur, dan kejelasan detail. *K-Means* tetap lebih cepat dalam proses segmentasi, namun untuk aplikasi yang membutuhkan pelestarian detail dan kealamian citra, BA menawarkan hasil yang lebih unggul.

d. 16 Cluster

Berdasarkan perhitungan jarak antar *cluster* dan pusat, menggunakan kedua algoritma, dari 8 *clusterisasi* warna, diperoleh hasil berikut:

Tabel 10 Hasil 16 *clusterisasi*

	<i>K-Means</i>	BA		<i>K-Means</i>	BA
C1	162418	123367	C9	63462	62602
C2	134501	158598	C10	110831	166178
C3	134806	155320	C11	302711	145157
C4	110016	108700	C12	147104	122706
C5	211639	82158	C13	46807	206018
c6	198506	149043	C14	172707	150792
C7	110724	166964	C15	93751	172832
C8	76722	167923	C16	227295	165642

Dengan parameter evaluasi

Tabel 11 Parameter evaluasi

Parameter Evaluasi	K-Means	Bee Algorithm (BA)
Total Jarak	2.294.061	2.255.827
Ketepatan Konvergensi	2.98×10^{-10}	1.42×10^{-10}
Kecepatan (detik)	2.34	5.12
Iterasi	24	53

Tabel 10 menunjukkan hasil *Clustering* dengan 16 *cluster* menggunakan algoritma *K-Means* dan *Bee Algorithm* (BA). Secara keseluruhan, kedua algoritma menghasilkan jarak yang cukup bervariasi antar *cluster*, namun terdapat beberapa perbedaan mencolok dalam distribusi kuantitas warna di setiap *cluster*. *K-Means* dan BA menunjukkan nilai yang cukup berbeda pada setiap *cluster*, baik pada *cluster* awal (C1 hingga C8) maupun pada *cluster* yang lebih tinggi (C9 hingga C16). Dari data ini, terlihat bahwa *K-Means* cenderung menghasilkan nilai yang lebih tersebar, dengan nilai kuantitas warna yang tidak terlalu merata antara *cluster-cluster*. Misalnya, pada *cluster* C1, *K-Means* memiliki nilai 162418, sementara BA hanya 123367, menunjukkan perbedaan distribusi yang lebih besar pada *cluster* pertama. Perbedaan yang signifikan juga terlihat pada *cluster* C5, di mana *K-Means* memiliki nilai 211639, sementara BA hanya 82158, menunjukkan ketidakseimbangan dalam distribusi warna di antara *cluster-cluster* tertentu.

BA cenderung menghasilkan distribusi yang lebih merata di antara *cluster-cluster*. Sebagai contoh, pada *cluster* C2, BA menunjukkan nilai 158598, sedikit lebih tinggi dibandingkan *K-Means* yang memiliki 134501. Selain itu, pada *cluster-cluster* seperti C9 hingga C16, BA lebih konsisten dalam menghasilkan nilai yang tidak terlalu fluktuatif, seperti pada *cluster* C9 (BA 62602, *K-Means* 63462) dan C10 (BA 166178, *K-Means* 110831). Hal ini menunjukkan bahwa BA lebih efektif dalam membagi data secara lebih seimbang ke dalam *cluster*, sementara *K-Means* mungkin kurang optimal dalam mencapai keseimbangan distribusi warna antara *cluster-cluster* pada jumlah *cluster* yang lebih tinggi.

Secara keseluruhan, meskipun perbedaan jarak antar kedua algoritma cukup kecil, BA cenderung lebih baik dalam menjaga keseimbangan distribusi warna antar *cluster* dibandingkan dengan *K-Means*, terutama pada jumlah *cluster* yang lebih besar. Perbedaan juga dapat dilihat dari visualisasi berikut:

Gambar 13 Kuantitas warna 16 *cluster*

Peningkatan jumlah *cluster* menjadi 16 menghasilkan segmentasi warna yang jauh lebih rinci dan kompleks dibandingkan gambar-gambar sebelumnya. Kedua metode berhasil memisahkan warna dengan lebih presisi, sehingga menghasilkan representasi visual yang lebih mendekati citra asli. Secara umum, baik *K-Means* maupun BA menunjukkan peningkatan signifikan dalam hal pelestarian fitur visual. Namun, metode BA (kanan) kembali menunjukkan hasil yang lebih halus dan realistis. Secara kuantisasi pengelompokan warna di ganti dengan pusat *clusternya*, ditunjukkan pada tabel berikut:

Tabel 12 Warna pada *cluster*

<i>Cluster</i>	K-Means		BA	
	RGB	HEX	RGB	HEX
1	RGB(6, 10, 19)	#060a13	RGB(1, 6, 14)	#01060e
2	RGB(193, 34, 20)	#c12214	RGB(227, 61, 22)	#e33d16
3	RGB(246, 210, 31)	#f6d21f	RGB(242, 195, 32)	#f2c320
4	RGB(86, 75, 72)	#564b48	RGB(73, 61, 65)	#493d41
5	RGB(243, 119, 32)	#f37720	RGB(236, 99, 29)	#ec631d
6	RGB(144, 20, 16)	#901410	RGB(104, 23, 15)	#68170f
7	RGB(61, 100, 186)	#3d64ba	RGB(64, 79, 166)	#404fa6
8	RGB(206, 68, 29)	#ce441d	RGB(153, 26, 15)	#991a0f
9	RGB(20, 39, 98)	#142762	RGB(20, 33, 81)	#142151
10	RGB(11, 16, 41)	#0b1029	RGB(5, 16, 38)	#051026
11	RGB(14, 58, 27)	#0e3a1b	RGB(10, 41, 24)	#0a2918
12	RGB(244, 147, 32)	#f49320	RGB(248, 142, 32)	#f88e20
13	RGB(97, 22, 15)	#61160f	RGB(109, 20, 14)	#6d140e
14	RGB(29, 34, 84)	#1d2254	RGB(16, 26, 52)	#101a34
15	RGB(51, 72, 162)	#3348a2	RGB(39, 94, 180)	#275eb4
16	RGB(3, 17, 32)	#031120	RGB(16, 25, 62)	#10193e

Gradasi warna tampak lebih alami, khususnya pada bagian bulu biru, yaitu warna ke biru-biruan, serta bayangan di sekitar mata. Hal ini menunjukkan kemampuan BA dalam menangkap variasi warna dan intensitas cahaya secara lebih efektif. Sementara itu, hasil dari *K-Means* (kiri) masih menunjukkan segmentasi yang cenderung terfragmentasi, meskipun dengan 16 *cluster*, batas antar warna menjadi lebih samar. Namun, transisi antar warna tetap terlihat lebih keras dibandingkan dengan hasil BA. Beberapa bagian latar belakang dan area paruh masih terlihat cukup datar dalam segmentasinya.

Persamaan yang dapat diamati adalah bahwa kedua metode berhasil mempertahankan struktur utama objek seperti bentuk mata, paruh, dan bulu secara utuh. Selain itu, peningkatan detail juga dapat dilihat pada bagian latar belakang dan peralihan warna di tubuh burung. Keduanya memanfaatkan jumlah *cluster* yang besar untuk membedakan nuansa warna yang

sebelumnya tidak terdeteksi dalam *cluster* lebih rendah. Sebagai kesimpulan, BA menghasilkan segmentasi yang lebih menyerupai foto asli dalam hal tekstur dan kedalaman warna. Sementara *K-Means* tetap unggul dalam kecepatan dan kesederhanaan pemrosesan, tetapi mulai menunjukkan keterbatasan dalam representasi gradasi halus. Dengan 16 *cluster*, BA lebih mampu mempertahankan keseimbangan antara detail dan kehalusan visual, menjadikannya lebih cocok untuk aplikasi yang membutuhkan visualisasi citra berkualitas tinggi.

4. KESIMPULAN

Berdasarkan keseluruhan hasil analisis dari *clusterisasi* data warna dengan jumlah *cluster* 2, 4, 8, dan 16, dapat disimpulkan bahwa algoritma *Bee Algorithm* (BA) menunjukkan performa yang lebih unggul dibandingkan *K-Means*. Hal ini terlihat dari nilai total jarak (inertia) yang secara konsisten lebih rendah pada BA, yang menandakan bahwa data lebih dekat ke pusat *clusternya*. Semakin kecil nilai jarak tersebut, semakin baik kualitas *cluster* yang terbentuk, sehingga BA mampu menghasilkan segmentasi warna yang lebih presisi. Selain itu, BA juga memperlihatkan distribusi kuantitas warna antar *cluster* yang lebih merata dibandingkan *K-Means*, terutama saat jumlah *cluster* bertambah. *K-Means* cenderung menghasilkan *cluster* dengan ukuran yang timpang, di mana satu atau dua *cluster* mendominasi jumlah data. Sebaliknya, BA mampu mengelompokkan data secara lebih seimbang, yang penting untuk menjaga proporsionalitas dalam analisis warna. Oleh karena itu, BA dapat dianggap sebagai metode yang lebih efektif untuk proses *clusterisasi* warna dibandingkan *K-Means*.

Berdasarkan hasil analisis dari gambar segmentasi dengan 2, 4, 8, dan 16 *cluster*, terlihat bahwa metode *Bee Algorithm* (BA) secara konsisten menghasilkan segmentasi warna yang lebih halus dan mendekati tampilan asli objek dibandingkan dengan metode *K-Means*. Dengan peningkatan jumlah *cluster*, kedua metode memang mengalami peningkatan kualitas visual, namun BA lebih unggul dalam mempertahankan detail halus seperti tekstur bulu dan gradasi warna alami. Sementara itu, *K-Means* cenderung menghasilkan pembagian warna yang lebih kaku dan berbentuk blok, yang mengurangi kealamian visual terutama pada jumlah *cluster* yang rendah.

Kesamaan antara kedua metode adalah kemampuannya dalam mengidentifikasi warna-warna dominan pada gambar dan menjaga struktur bentuk utama burung. Namun, untuk keperluan segmentasi gambar yang membutuhkan representasi detail dan realistis, BA lebih direkomendasikan karena memberikan hasil yang lebih baik dalam menyampaikan informasi visual. Sementara itu, *K-Means* tetap dapat digunakan untuk tugas-tugas segmentasi yang lebih ringan dan tidak memerlukan detail tinggi, seperti klasifikasi kasar atau *preprocessing* awal dalam sistem pengolahan citra.

Untuk penelitian selanjutnya, disarankan agar pengembangan algoritma *Bee Algorithm* (BA) difokuskan pada variasi parameter pencarian dan strategi eksplorasi yang lebih adaptif agar mampu menangani data berukuran besar atau berdimensi tinggi secara lebih efisien. Selain itu, kombinasi BA dengan algoritma lain seperti *Particle Swarm Optimization* (PSO) atau Ant Colony Optimization (ACO) juga layak dieksplorasi guna meningkatkan akurasi dan konvergensi dalam proses *clusterisasi*. Penelitian lanjutan juga dapat memperluas penerapan BA dalam bidang citra digital lainnya, seperti segmentasi medis, pengenalan pola, dan pemrosesan video berbasis warna.

REFERENSI

- [1] K. Adnan and R. Akbar, "An analytical study of information extraction from unstructured and multidimensional big data," *J Big Data*, vol. 6, no. 1, p. 91, Dec. 2019, doi: 10.1186/s40537-019-0254-8.
- [2] C. D. S. Indrawati, P. Ninghardjanti, A. Subarno, C. H. Atma Dirgatama, and A. Wahyu Wirawan, "Pelatihan manajemen editing video pembelajaran Pada MGMP otomatisasi dan tata kelola perkantoran di Surakarta," *pkmsr*, vol. 3, Jun. 2021, doi: 10.37695/pkmsr.v3i0.787.
- [3] H. Hernawan and P. Sidiq, "Pengembangan Sistem Pakar sebagai Bahan Ajar dalam Pembelajaran Fisiologi Hewan sub Materi Fisiologi Sirkulasi," *JPETIK*, vol. 2, no. 2, p. 17, May 2018, doi: 10.31980/jpetik.v2i2.70.
- [4] B. Fu, X. Zhao, C. Song, X. Li, and X. Wang, "A salt and pepper noise image denoising method based on the generative classification," *Multimed Tools Appl*, vol. 78, no. 9, pp. 12043–12053, May 2019, doi: 10.1007/s11042-018-6732-8.
- [5] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4761–4767, Jul. 2010, doi: 10.1016/j.eswa.2009.11.003.
- [6] Z. Peng, Y. Huang, and Y. Zhong, "A discrete artificial bee colony algorithm for quadratic assignment problem," *JHS*, vol. 28, no. 2, pp. 131–141, May 2022, doi: 10.3233/JHS-220684.
- [7] Y. Li, M. Bai, X. Wang, D. Wu, and Q. Zhao, "Artificial Bee Colony Algorithm-Based Ultrasound Image Features in the Analysis of the Influence of Different Anesthesia Methods on Lung Air Volume in Orthopedic Surgery Patients," *Scientific Programming*, vol. 2021, pp. 1–12, Aug. 2021, doi: 10.1155/2021/9958392.
- [8] R. Durgut, "Improved Binary Artificial Bee Colony Algorithm," 2020, *arXiv*. doi: 10.48550/ARXIV.2003.11641.
- [9] Zhang, Y., & Li, S. (2012). Dynamic image segmentation method using hierarchical clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (pp. 185–194). Springer. https://doi.org/10.1007/978-3-642-10268-4_21.
- [10] Afzal, S., & Khan, M. A. (2019). Image segmentation based on fuzzy clustering with cellular automata and features weighting. *EURASIP Journal on Image and Video Processing*, 2019(1). <https://doi.org/10.1186/s13640-019-0436-5>.
- [11] P. Modey *et al.*, "K-Means Based Bee Colony Optimization for Clustering in Heterogeneous Sensor Network," *Sensors*, vol. 24, no. 23, p. 7603, Nov. 2024, doi: 10.3390/s24237603.
- [12] V. Agrawal and S. Chandra, "Feature selection using Artificial Bee Colony algorithm for medical image classification," in *2015 Eighth International Conference on Contemporary Computing (IC3)*, Noida, India: IEEE, Aug. 2015, pp. 171–176. doi: 10.1109/IC3.2015.7346674.
- [13] Xu, L., Zhao, W., & Wang, J. (2024). Sonar image segmentation using a multi-spatial information constraint fuzzy C-means clustering algorithm based on KL divergence. *International Journal of Machine Learning and Cybernetics*. <https://doi.org/10.1007/s13042-024-02419-6>
- [14] Barcelos, I. B., Belém, F. de C., João, L. de M., Patrocínio Jr., Z. K. G. do, Falcão, A. X., & Guimarães, S. J. F. (2024). A comprehensive review and new taxonomy on superpixel segmentation. *ACM Computing Surveys*, 57(6), Article 124. <https://doi.org/10.1145/3652509>.
- [15] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*, 1st ed. in Wiley Series in Probability and Statistics. Wiley, 2011. doi: 10.1002/9780470977811.
- [16] G. J. Oyewole and G. A. Thopil, "Data clustering: application and trends," *Artif Intell Rev*, vol. 56, no. 7, pp. 6439–6475, Jul. 2023, doi: 10.1007/s10462-022-10325-y.