

PENENTUAN JUMLAH *CLUSTER* OPTIMUM PADA SEGMENT RUTE PENERBANGAN MENGGUNAKAN DATA *AUTOMATIC DEPENDENT SURVEILLANCE-BROADCAST*

Reza Prasetya Prayogo¹⁾, Joko Lianto Buliali²⁾

^{1,2)}Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Jalan Raya ITS, Surabaya

e-mail: reza.prasetya.prayogo@gmail.com¹⁾, joko@cs.its.ac.id²⁾

ABSTRAK

Terdapat beberapa titik acuan dalam satu rute penerbangan untuk keperluan navigasi yang disebut *waypoint*. Pada penelitian ini penulis melakukan segmentasi untuk membagi satu rute penerbangan (Surabaya-Palu) menjadi 7 segmen yang terdiri dari 8 *waypoint*, dengan membuat garis imajiner secara tegak lurus melewati masing-masing *waypoint*. Pada tiap segmen dilakukan analisa terkait lokasi yang paling sering dilalui menggunakan pendekatan clustering.

Dalam penelitian ini penulis menggunakan algoritma clustering *K-means* dengan optimasi centroid yang mengimplementasikan algoritma *Ant Lion Optimizer (ALO)* atau disebut dengan *K-means-ALO*. Jumlah cluster ditentukan sebelumnya, kemudian dilakukan validasi pengelompokan internal dengan menggunakan *silhouette index*. Hasil metode pengelompokan diuji nilai performansinya. Hasil akhir dari jumlah cluster yang sudah ditentukan diambil nilai validitas cluster terbaik yaitu jumlah cluster yang optimum pada tiap segmen area penerbangan.

Pengujian dilakukan dengan membandingkan nilai *silhouette index* untuk setiap percobaan jumlah cluster terhadap kedua metode yaitu *K-means* dan *K-means-ALO*. Pada uji coba yang dilakukan, metode optimasi yang diusulkan menghasilkan validitas cluster yang lebih baik sesuai nilai *silhouette index* pada tiga segmen, yaitu segmen 2, 3, dan 5 akan tetapi signifikan di semua segmen berdasarkan uji statistik *Analysis of Variance (ANOVA)* dan uji lanjut *Least Significant Difference (LSD)*.

Kata Kunci: *Ant Lion Optimizer, Analysis of Variance, Kinerja dan Validitas Cluster, Least Significant Difference, Segmen Penerbangan.*

ABSTRACT

There are several reference points in one flight route for the purpose of navigation called a *waypoint*. In this study the author segmented the flight route (Surabaya-Palu) into 7 segments consisting of 8 *waypoints*, by making imaginary lines perpendicular to each *waypoint*. In each segment, an analysis was made related to the location that most frequently traversed using the clustering approach.

In this study the authors use the *K-means clustering algorithm* with centroid optimization which implements the *Ant Lion Optimizer algorithm (ALO)* or called the *K-means-ALO*. The number of clusters is predetermined, then internal grouping is validated using the *silhouette index*. The results of the grouping methods are tested for their performance value. The final result of the number of clusters that have been determined is the best cluster validity value, which is the optimum number of clusters in each flight area segment.

Testing is done by comparing the value of the *silhouette index* for each experiment the number of clusters between the two methods namely *K-means* and *K-means-ALO*. In the trial run, the proposed optimization method produces better cluster validity according to the value of the *silhouette index* in three segments, namely segments 2, 3, and 5 but it is significant in all segments based on *Analysis of Variance (ANOVA)* statistical tests and further tests *Least Significant Difference (LSD)*.

Keywords: *Ant Lion Optimizer, Analysis of Variance, Cluster Performance and Validity, Flight Segment, Least Significant Difference.*

I. PENDAHULUAN

FAKTOR utama dalam transportasi udara adalah keselamatan dalam penerbangan. Oleh karena itu perilaku pesawat di luar kebiasaan harus bisa terdeteksi secepat mungkin untuk mengantisipasi terjadinya hal yang tidak diinginkan. Terdapat beberapa titik acuan dalam satu rute penerbangan untuk keperluan navigasi yang disebut *waypoint*. Dalam hal ini penulis berpendapat bahwa tidak hanya dari *waypoint* saja, akan tetapi data ADS-B yang secara otomatis dikirimkan terus menerus oleh pesawat selama perjalanan bisa digunakan untuk melihat pola pergerakan pesawat pada rute tersebut. Pola yang dimaksud adalah kumpulan titik sehingga membentuk suatu area yang merupakan parameter *latitude* dan *longitude* dari data ADS-B. Dengan adanya pola yang terbentuk, diharapkan *Air Traffic Control* dapat dengan mudah melihat area mana saja yang paling sering dilewati dan begitu juga sebaliknya, area mana saja yang paling jarang dilewati oleh pesawat. Pada penelitian sebelumnya tentang *waypoint* dan deteksi anomali penerbangan, M.Y. Pusadan, J. L. Buliali, and R. V. H. Ginardi [1] meneliti tentang penentuan *waypoint* optimum untuk deteksi anomali pada rute penerbangan. Dari beberapa *waypoint* optimum tersebut, selanjutnya ditentukan segmen-segmen yang merupakan wilayah diantara suatu

waypoint dengan *waypoint* sesudahnya. Pada penelitian ini penulis melakukan segmentasi untuk membagi rute penerbangan ke dalam beberapa segmen akan tetapi menggunakan *waypoint* yang sudah ditentukan oleh *Air Navigation Indonesia*. Tujuan utama dari penelitian ini adalah menganalisa area yang sering dilewati dan sebaliknya, area yang jarang dilewati oleh pesawat. Oleh karena itu dibutuhkan proses pengelompokan data berdasarkan jarak sehingga penulis menggunakan algoritma *clustering*.

Algoritma *clustering* berbasis centroid yang digunakan dalam penelitian ini adalah *K-means* [2] dengan penerapan metode optimasi centroid *Ant Lion Optimizer* [3] [4]. Penulis menggunakan algoritma *K-means* agar dapat bebas menentukan jumlah kelompok data (*cluster*) untuk melakukan simulasi terkait area yang paling sering dilewati (memiliki keanggotaan data paling banyak) dan sebaliknya, area yang paling jarang dilewati (memiliki keanggotaan data paling sedikit). Selain itu juga beberapa kelebihan *K-means* adalah mudah diadaptasi, dan membutuhkan proses pengelompokan data yang relatif cepat. Pada algoritma tersebut jumlah *k cluster* harus ditentukan terlebih dahulu. Oleh karena itu diperlukan pengukuran validitas *cluster* [5] untuk menentukan jumlah *cluster* yang tepat, dimana dalam hal ini peneliti menyebutnya sebagai jumlah *cluster* yang optimum.

Dengan menggunakan algoritma *clustering K-means* seperti yang telah disebutkan di atas, penentuan jumlah *cluster* yang optimum menjadi faktor yang paling penting. Terdapat beberapa pendekatan sederhana yang umum dan bisa digunakan antara lain *Davies-Bouldin Index* [6], *Calinski-Harabasz Index* [7], *Dunn's Index*, *Silhouette index* [8]. Dengan adanya penelitian ini diharapkan dapat membantu *Air Traffic Management* dalam memandu dan mengamati (*surveillance*) lalu lintas penerbangan pada suatu rute berdasarkan pola kebiasaan dari data ADS-B yang diperoleh. Selain itu juga diharapkan penelitian dapat dipakai sebagai acuan dalam penentuan deteksi dini anomali penerbangan berdasarkan analisis dari kebiasaan pesawat yang melintasi suatu rute penerbangan yang ditentukan.

II. STUDI LITERATUR

A. ADS-B

ADS-B merupakan sistem informasi pemantauan pengawasan pesawat tanpa radar. *Federal Aviation Administration* (FAA) memberikan perintah penggunaan sistem ADS-B kepada *Air Traffic Control* (ATC) karena beberapa kelebihan yaitu alat yang ringan dan biaya murah, sehingga sistem pesawat tak berawak pun tidak lagi dibatasi oleh masalah berat muatan dan kemampuan daya terbang [9]. Pesawat yang dilengkapi dengan ADS-B dapat mengirimkan informasi dengan tingkat frekuensi lebih tinggi serta lebih komprehensif daripada radar. Selain itu ADS-B juga dapat mengirimkan informasi dengan posisi lebih tinggi dan akurasi kecepatan pesawat dengan lebih baik. Sistem ini mengirimkan data sekali per detik, lebih cepat dari radar dengan kecepatan pengiriman sekali per 4.8 detik di area terminal dan sekali per 12 detik di ruang udara.

Terdapat sebuah penelitian tentang algoritma untuk peringatan dan kesadaran situasi lalu lintas menggunakan ADS-B dimana penelitian tersebut bertujuan untuk memberikan peringatan kepada pilot tentang adanya potensi tabrakan dan situasi berbahaya [10]. Dalam penelitian lain, ADS-B digunakan sebagai input untuk algoritma deteksi konflik sebagai dukungan pada sistem penghindaran tabrakan di masa yang akan datang [11]. Penelitian tentang algoritma deteksi konflik prediksi lintasan berbasis ADS-B juga sudah mendapatkan perhatian dari para peneliti [12].

B. Metode Penentuan Cluster Centroid (*K-means*)

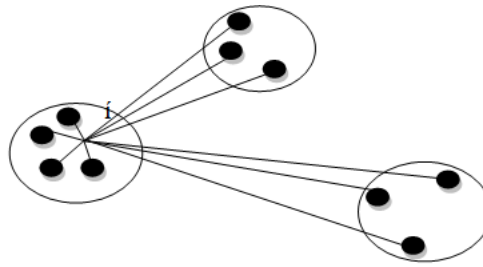
Dalam penelitian ini algoritma *clustering* yang digunakan adalah *clustering* berdasarkan jarak kedekatan yaitu *K-means* [13]. *Clustering* berbasis centroid digunakan karena tujuan dari penelitian ini adalah memperoleh area yang merupakan representasi dari *cluster-cluster* yang terbentuk dari sebaran data pada tiap segmen. *K-means* adalah salah satu algoritma *unsupervised learning* yang paling sederhana yang memecahkan masalah *clustering* yang terkenal mudah diadaptasi, dan membutuhkan proses pengelompokan data yang relatif cepat. Ide utamanya adalah mendefinisikan satu *k center* untuk setiap *cluster*. Langkah-langkah dari algoritma *K-means* adalah sebagai berikut :

$x = \{x_1, x_2, x_3, \dots, x_n\}$ adalah himpunan titik data dan $v = \{v_1, v_2, \dots, v_c\}$ adalah himpunan titik pusat *cluster* (*k center*).

1. Pilih titik pusat secara acak.
2. Hitung jarak antara setiap titik data dan pusat *cluster*.
3. Tetapkan titik data ke pusat *cluster* yang jaraknya dari *cluster center* minimal dari semua pusat *cluster*.
4. Hitung ulang pusat *cluster* baru dengan menggunakan :

$$v_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_j, \quad (1)$$

di mana *c* merupakan titik centroid dan *x* adalah data yang akan dihitung jaraknya.



Gambar 1. Ilustrasi elemen-elemen pada komputasi elemen I, objek dari cluster A.

5. Hitung ulang jarak antara setiap titik data dan pusat cluster yang baru didapat. Jika tidak ada titik data yang dipindahkan maka berhenti, jika tidak ulangi dari langkah 3.

C. Metode Optimasi Centroid

Ant Lion Optimizer (ALO) merupakan salah satu metode yang dapat digunakan untuk mengoptimalkan penentuan centroid dalam proses clustering [3]. Terdapat lima langkah utama pada metode ini yaitu : pergerakan acak semut, membuat jebakan (lubang), menjebak mangsa di dalam lubang jebakan, menangkap mangsa dan membangun ulang jebakan. Metode optimasi ini dilakukan setelah proses clustering pada himpunan data. Untuk setiap cluster yang terbentuk ditentukan posisi *Ant Lion*. Kemudian ukur nilai *fitness value* dari *objective function*, yaitu persamaan dari metode *K-means*. *Ant Lion* dengan nilai *fitness* minimum ditetapkan sebagai *Elite (centroid)*. *Random Walk* untuk iterasi dilakukan dengan menggunakan persamaan (1). Untuk membatasi *random walk* agar tetap di dalam area pencarian, dilakukan normalisasi menggunakan persamaan (2).

$$X[t] = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \text{cumsum}(2r(t_n) - 1)], \quad (2)$$

di mana $r(t) = 1$ jika $\text{rand} > 0.5$ atau 0 jika $\text{rand} \leq 0.5$

D. Metode Pengukuran Validitas Cluster

Faktor utama yang mempengaruhi validasi cluster adalah pengukuran validitas *cluster internal* dalam memilih jumlah cluster yang optimal. Beberapa parameter yang menjadi karakteristik adalah kerapatan pemisahan yang diukur pada jarak *intra-cluster* [14]. Dimana metode pengukuran validitas cluster dapat dikategorikan menjadi dua yaitu *external* dan *internal* [5], akan tetapi ada juga yang mengkategorikannya menjadi tiga yaitu validasi *internal*, *external*, dan relatif [15]. *Silhouette* berguna ketika mencari cluster yang rapat dan benar-benar terpisah. Ilustrasi dari perhitungan nilai *silhouette index* dapat dilihat pada Gambar 1.

Koefisien *silhouette* dari individu objek $x(j)$ didefinisikan pada persamaan (3) sebagai berikut :

$$S_{x(j)} = \frac{b_{p,j} - a_{p,j}}{\max\{a_{p,j}, b_{p,j}\}} \quad (3)$$

Hitung rata-rata jarak setiap objek ke- j dengan semua objek yang ada pada cluster p , cluster yang sama dengan objek j , dilambangkan dengan $a_{p,j}$. Kemudian, hitung rata-rata jarak dari setiap objek ke- j dengan semua objek yang ada pada cluster q , dimana p tidak sama dengan q , disebut $d_{q,j}$. Lalu, cari $a_{p,j}$ dari minimum $d_{q,j}$, yang menunjukkan perbedaan rata-rata objek $x(j)$ untuk cluster yang terdekat dengan tetangganya. Untuk $S_{x(j)}$ yang memiliki nilai tinggi, maka semakin tepat penempatan $x(j)$ ke cluster p . Nilai koefisien *silhouette* biasanya di antara rentang -1 sampai dengan 1.

E. Analysis of Variance

Analysis of variance (ANOVA) adalah kumpulan dari model statistik yang digunakan untuk menganalisis perbedaan rata-rata antara kelompok dan prosedur terkait (seperti “variasi” antara kelompok), yang dikembangkan oleh Ronald Fisher. ANOVA menggunakan uji f untuk menguji secara statistik. ANOVA termasuk jenis analisis statistik yang sangat umum, berguna dalam desain eksperimental yang sangat sederhana dan sangat rumit [16].

Dalam ANOVA hanya satu hipotesis yang digunakan, yaitu hipotesis dua arah (*two tail*) artinya hipotesis ini ingin mengetahui apakah ada perbedaan rata-rata atau tidak dan tidak spesifik yang mana yang berbeda. Metode statistik ANOVA memiliki beberapa jenis salah satunya adalah *one-way* ANOVA. *One-way* ANOVA digunakan untuk menguji perbedaan tiga kelompok atau lebih berdasarkan satu variabel independen.

F. Least Significant Difference (LSD)

Tes LSD dikembangkan oleh Fisher, Ronald Aylmer [17], yang ingin mengetahui perlakuan mana yang memiliki efek signifikan dalam analisis varian. *Least Significant Difference* (LSD) digunakan ketika F-rasio menunjukkan penolakan terhadap hipotesis nol (H_0), dan terjadi ketika terdapat perbedaan signifikan antara rata-rata populasi. Tes ini membantu mengidentifikasi populasi yang berbeda secara statistik. Persamaan LSD adalah ditunjukkan pada persamaan (4) sebagai berikut:

$$LSD = \frac{t(s\sqrt{2})}{\sqrt{n}}, \tag{4}$$

di mana nilai t yang ditabulasikan pada tingkat probabilitas (misalnya $P < 0.01$) dan dengan derajat kebebasan yang sama dengan (jumlah varietas minus 1) x (jumlah pengukuran untuk setiap varietas minus 1).

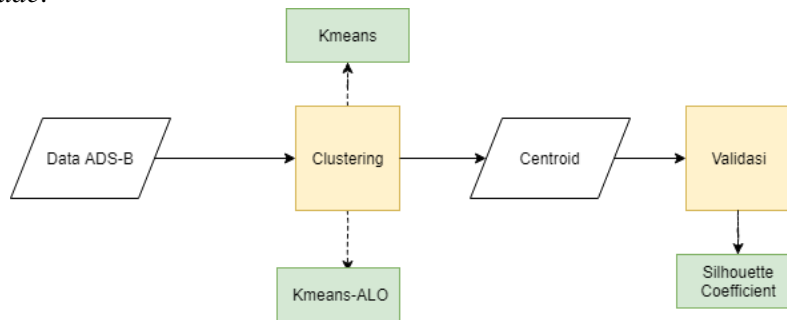
III. METODE PENELITIAN

Desain metode penelitian ditunjukkan pada Gambar 2, dimana penelitian diawali dengan membagi data ADS-B menjadi beberapa segmen berdasarkan *waypoint*. Kemudian pada tiap segmen dilakukan metode *clustering* yaitu *K-Medoids* dan *K-means*. Hasil *clustering* dari kedua metode tersebut akan membentuk beberapa kelompok titik koordinat yang merupakan representasi posisi pesawat pada rute penerbangannya.

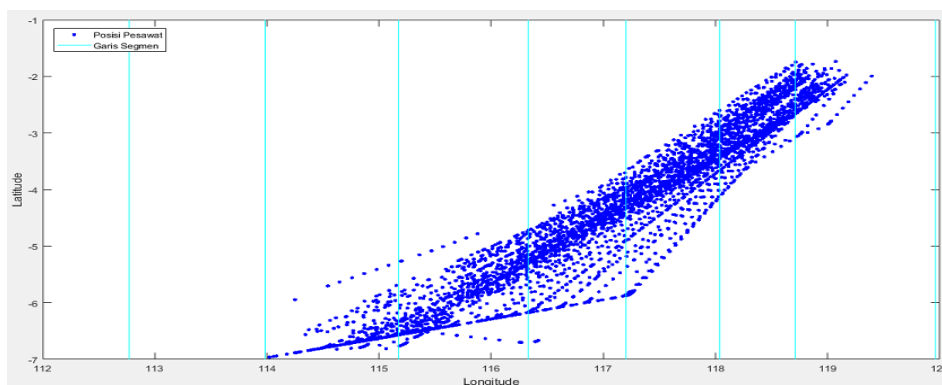
A. Pre-Processing

Pada tahap ini dilakukan pembagian segmen berdasarkan 8 *waypoint* pada rute penerbangan Surabaya – Palu dengan menggunakan data *Callsign* LNI-860. Segmen yang digunakan pada penelitian ini adalah segmen dimana pesawat berada dalam posisi *cruising*, bukan *take off*, *landing*, ataupun *approaching*. Segmen dibuat berdasarkan letak *latitude* dan *logitude* tertentu dengan jarak antar titik yang telah ditentukan. Perbedaan antar segmen dipisahkan dengan garis biru vertikal yang divisualisasikan pada Gambar 3 yang mereprestasikan posisi pesawat dalam segmen yang telah dibuat.

Segmen terbentuk dari 2 *waypoint* yang ada yaitu *waypoint* sebelum dan sesudah. Berikut adalah daftar *waypoint* yang mengelompokkan data berdasarkan segmen disajikan pada Tabel I yang berupa *waypoint* dengan posisi *latitude* dan *longitude*.



Gambar 2. Desain metode penelitian.



Gambar 3. Visualisasi segmen rute penerbangan berdasarkan *waypoint* pada rute Surabaya-Palu.

TABEL I
DAFTAR *WAYPOINT* RUTE PENERBANGAN SURABAYA – PALU.

<i>Waypoint</i>	<i>Latitude</i>	<i>Longitude</i>
Surabaya	-7.37333	112.77167
Fando	-6.97333	113.98500
Kasol	-6.56833	115.17333
Dasty	-6.17333	116.33000
Endog	-5.87667	117.20167
Gamal	-2.86333	118.03833
Rudal	-2.66167	118.71167
Palu	-0.88500	119.96167

B. Metode Clustering

Metode yang digunakan pada pengelompokan berdasarkan jarak kedekatan adalah metode *K-means* dan *K-Medoids* dengan metode pengukuran jarak yaitu *euclidian distance*. Himpunan data akan dibagi berdasarkan *latitude* dan *longitude waypoint* pada Tabel 1. Kemudian proses *clustering* dilakukan pada segmen 2 sampai segmen 7. Pada setiap segmen, dilakukan percobaan dengan inisialisasi *cluster* sejumlah 3 sampai 10. *Programming code* metode *clustering* penelitian ditunjukkan selengkapnya pada Gambar 4.

Sebagai contoh, visualisasi metode *K-means* dengan inisialisasi jumlah *cluster* sebanyak 3 *cluster* pada segmen 2 ditunjukkan pada Gambar 5. Untuk setiap segmen dilakukan uji coba jumlah *cluster* dari 3 sampai dengan 10 *cluster*.

C. Optimasi Centroid

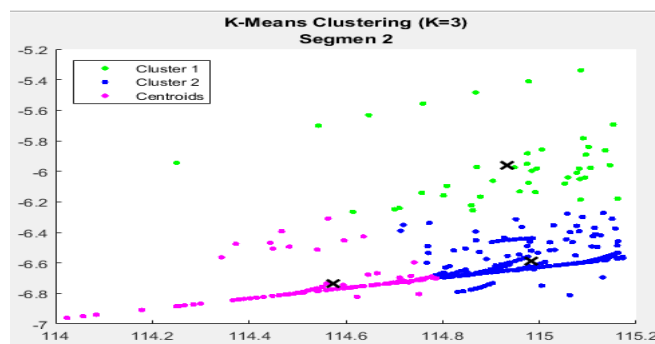
Pada setiap percobaan di tiap segmen dilakukan inisialisasi jumlah *cluster* antara 3 sampai 10. Percobaan dilakukan dengan iterasi sebanyak 100, 500, dan 1000. Centroid yang sudah terbentuk selanjutnya diperbaiki atau dioptimasi menggunakan metode *Ant Lion Optimizer*. *Programming code* untuk optimasi *centroid* ditunjukkan pada Gambar 6 dengan menggunakan 1000 iterasi. Karena *programming code* yang cukup panjang gambar dipisahkan menjadi 2 bagian yaitu Gambar 6.1 dan Gambar 6.2.

```

segmen = 7;
jumlah_k = 10;
z = 1;
load('waypoints.mat');
load(strcat('fdatas',int2str(segmen),'.mat'));
fdata = eval(strcat('fdatas',int2str(segmen)));
fdata = [fdata(:,2) fdata(:,1)];
for i=2:segmen
    list_segmen = [list_segmen;i];
    for j=3:jumlah_k
        [data_cluster,cluster_index_kmeans,centroid_kmeans]=index_kmeans(fdata,i,j);
        [data_silhouette, index_rerata, index_max, index_min] = Silhouette(data_cluster);
        r_silhouette = [r_silhouette;i,j,index_rerata];
        z = z+1;
    end
end
end

```

Gambar 4. *Programming code clustering* segmen 2 sampai 7.



Gambar 5. Visualisasi hasil *K-means clustering*.

```

t_iterasi = 1000;
jumlah_k = 10;
for t=1:t_iterasi
    [antlions_roulette_slot]=alo_roulette_wheel_slot(elites,init_antlion_number);
    [antlions_roulette]=alo_roulette_wheel(antlions,antlions_roulette_slot);
    disp(num2str(t));
    antlions_new = [];
    elites_new = [];
    compare_ants = [];
    for i=2:segmen
        load(strcat('Data/fdatas',int2str(i),'.mat'));
        fdata = eval(strcat('fdatas',int2str(i)));
        fdata = [fdata(:,2) fdata(:,1)];
        [edata] = random_walk(fdata);
        eval(['move' num2str(i) '= edata;']);
        move = eval(strcat('move',int2str(i)));
        for a=1:length(fdata)
            move_data(a,1) = fdata(a,1) + move(a,1);
            move_data(a,2) = fdata(a,2) + move(a,2);
            eval(['edata' num2str(i) '= move_data;']);
        end
    end
end

```

Gambar 6.1. *Programming code optimasi Centroid Ant Lion Optimizer* bagian 1.

```

move_data = [0,0];
for j=3:jumlah_k [edata_cluster,cluster_index_kmeans,get_centroid_kmeans]=kmedoids_classic
(eval(strcat('edata',int2str(i))),i,j,t);
[edata_silhouette, index_rerata, index_max, index_min] = Silhouette(edata_cluster);
edata_fitness_k = [edata_fitness_k;edata_silhouette];
eval(['ants_s' num2str(i) '_k' num2str(j) '= edata_silhouette;']);
get_dt = eval(strcat('ants_s',num2str(i),'_k',num2str(j)));
disp(strcat('ants_s',num2str(i),'_k',num2str(j)));
sort_ants = sortrows(get_dt,4,'descend');
for e=1:j
for d=1:length(sort_ants)
if sort_ants(d,3) == e
compare_ants_tmp(1,1) = i;
compare_ants_tmp(1,2) = j;
compare_ants_tmp(1,3) = sort_ants(d,3);
compare_ants_tmp(1,4) = sort_ants(d,1);
compare_ants_tmp(1,5) = sort_ants(d,2);
compare_ants_tmp(1,6) = sort_ants(d,4);
compare_ants = [compare_ants;compare_ants_tmp];
break;
else
continue;
end
end
end
eval(['compare_ants' num2str(i) '_k' num2str(j) '= compare_ants;']);
f_stop = size(compare_ants,1);
f_start = f_stop - j + 1;
disp(strcat(num2str(f_start),'-',num2str(f_stop)));
for f=f_start:f_stop
if antlions_roulette(f,9) < compare_ants(f,6)
antlions_new_tmp(1,1) = compare_ants(f,1);
antlions_new_tmp(1,2) = compare_ants(f,2);
antlions_new_tmp(1,3) = compare_ants(f,3);
antlions_new_tmp(1,4) = compare_ants(f,4);
antlions_new_tmp(1,5) = compare_ants(f,5);
antlions_new_tmp(1,6) = compare_ants(f,6);
else
antlions_new_tmp(1,1) = antlions_roulette(f,3);
antlions_new_tmp(1,2) = antlions_roulette(f,4);
antlions_new_tmp(1,3) = antlions_roulette(f,5);
antlions_new_tmp(1,4) = antlions_roulette(f,7);
antlions_new_tmp(1,5) = antlions_roulette(f,8);
antlions_new_tmp(1,6) = antlions_roulette(f,9);
end
antlions_new = [antlions_new;antlions_new_tmp];
antlions_new_tmp = [];
if elites(f,8) < antlions_new(f,6)
elites_new_tmp(1,1) = antlions_new(f,1);
elites_new_tmp(1,2) = antlions_new(f,2);
elites_new_tmp(1,3) = antlions_new(f,3);
elites_new_tmp(1,4) = antlions_new(f,4);
elites_new_tmp(1,5) = antlions_new(f,5);
elites_new_tmp(1,6) = antlions_new(f,6);
else
elites_new_tmp(1,1) = elites(f,3);
elites_new_tmp(1,2) = elites(f,4);
elites_new_tmp(1,3) = elites(f,5);
elites_new_tmp(1,4) = elites(f,6);
elites_new_tmp(1,5) = elites(f,7);
elites_new_tmp(1,6) = elites(f,8);
end
elites_new = [elites_new;elites_new_tmp];
elites_new_tmp = [];
end
end
end
end
end

```

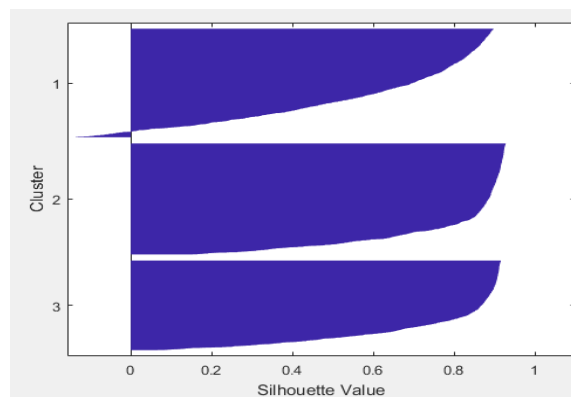
Gambar 6.2. Programming code optimasi Centroid Ant Lion Optimizer bagian 2.

D. Validasi Cluster

Setelah ketiga langkah di atas menghasilkan *cluster* sesuai jumlah *k* yang ditentukan, dibutuhkan pengukuran validasi untuk menentukan jumlah *cluster* yang paling sesuai untuk tiap segmen. Pengukuran validasi ditunjukkan dalam *programming code* pada Gambar 7 dengan inputan masing-masing data cluster yang telah dibuat. Sebagai hasilnya, Gambar 8 merupakan representasi salah satu metode pengukuran validitas *cluster internal* yaitu *silhouette coefficient*. Jumlah *cluster* yang optimum adalah ketika nilai indeks mendekati 1.

```
function[output1, output2, output3, output4]=Silhouette(dataCluster)
[s]=silhouette(dataCluster(:,1:2), dataCluster(:,3), 'SqEuclidean'); %[s,h]
sort_s=sort(s);
max_s=max(s);
min_s=min(s);
for i=1:length(dataCluster)
    Silhouette_result(i,1) = dataCluster(i,1);
    Silhouette_result(i,2) = dataCluster(i,2);
    Silhouette_result(i,3) = dataCluster(i,3);
    Silhouette_result(i,4) = s(i,1);
    Silhouette_average(i,1) = Silhouette_result(i,4);
end
output1 = Silhouette_result;
output2 = sum(Silhouette_average)/length(Silhouette_average);
output3 = max_s;
output4 = min_s;
end
```

Gambar 7. Programming code validasi Cluster Silhouette Coefficient.



Gambar 8. Silhouette index pada metode K-means dengan 3 cluster pada segmen 2.

IV. HASIL DAN PEMBAHASAN

Pengujian terkait pengelompokan data yang dilakukan meliputi: Pengukuran validitas *cluster* dengan membandingkan nilai *silhouette index* untuk setiap percobaan jumlah *cluster* terhadap kedua metode yaitu metode *K-means* dan metode yang diusulkan yaitu *K-means - ALO*. Melakukan analisis hasil *cluster* berupa nilai *silhouette index* menggunakan metode uji statistik dan uji lanjut. Metode uji statistik yang digunakan adalah metode *one-way Analysis of Variance (one-way ANOVA)* dan *Least Significant Difference (LSD)* sebagai uji lanjut.

A. Data Uji Coba

Data uji coba yang dipakai dalam penelitian ini adalah data penerbangan rute Surabaya – Palu (*Callsign LNI-860*). Dengan melakukan pembagian segmen berdasarkan 8 *waypoint* dengan kondisi pesawat berada dalam posisi *cruising* (tidak mengalami perubahan ketinggian), bukan *take off*, *landing*, ataupun *approaching*. Pengujian dilakukan pada segmen 2 sampai 7 dengan jumlah *cluster* mulai dari 3 sampai 10.

B. Uji Coba Validitas Cluster K-means dan K-means - ALO

Metode yang diusulkan dibandingkan dengan *K-means* tradisional. Pengujian dilakukan pada segmen 2 sampai 7 dengan jumlah *cluster* mulai dari 3 sampai 10. Pengujian pada masing-masing segmen dilakukan dengan 500 iterasi. Dalam pengujian ini didapatkan hasil evaluasi *cluster* berdasarkan nilai *silhouette index*. Tabel II menunjukkan hasil nilai rata-rata *silhouette index* pada masing-masing segmen dan *cluster* yang telah diuji coba. Semakin tinggi nilai rata-rata *silhouette index* semakin tinggi pula nilai validitas dari suatu *cluster*.

C. Analisa Uji Coba Statistik (ANOVA)

Dalam tahap ini, uji coba statistik dilakukan untuk mengetahui apakah ada perbedaan diantara jumlah *k cluster*. Pengujian ANOVA dilakukan pada masing-masing segmen untuk menghasilkan nilai $Prob > F$ atau *p-value* sebesar 0, dimana apabila nilai tersebut kurang dari 0.05, jumlah *k cluster* berpengaruh terhadap himpunan data yang digunakan. Untuk hasil perhitungan uji statistik pada masing-masing segmen ditunjukkan pada Tabel III dimana perlu atau tidaknya uji lanjut LSD dilakukan.

TABEL II
NILAI RATA-RATA *SILHOUETTE INDEX* TIAP SEGMENT DAN *CLUSTER*.

Segmen	Metode	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
2	Classic <i>K-means</i>	0.6849	0.6823	0.6824	0.6720	0.6588	0.6452	0.6423	0.6479
	<i>K-means</i> -ALO	0.7732	0.7550	0.7402	0.7284	0.5427	0.5798	0.4582	0.5332
3	Classic <i>K-means</i>	0.6849	0.6823	0.6807	0.6583	0.6592	0.6412	0.6417	0.6483
	<i>K-means</i> -ALO	0.7732	0.7169	0.7402	0.7284	0.7212	0.6815	0.5427	0.4716
4	Classic <i>K-means</i>	0.6677	0.6389	0.5974	0.5908	0.5884	0.5583	0.5539	0.5564
	<i>K-means</i> -ALO	0.6574	0.6576	0.6579	0.4960	0.4775	0.4980	0.4981	0.1853
5	Classic <i>K-means</i>	0.6620	0.5727	0.5652	0.5737	0.5623	0.5657	0.5636	0.5548
	<i>K-means</i> -ALO	0.7113	0.7324	0.7350	0.7374	0.2540	0.7029	0.2497	0.2932
6	Classic <i>K-means</i>	0.9546	0.9488	0.9459	0.9467	0.9465	0.9464	0.9462	0.9435
	<i>K-means</i> -ALO	0.9290	0.6259	0.8346	0.3444	0.8434	0.3516	0.3552	0.3588
7	Classic <i>K-means</i>	0.6428	0.6438	0.6236	0.6360	0.6344	0.6296	0.6269	0.6214
	<i>K-means</i> -ALO	0.5932	0.5642	0.5600	0.5499	0.5476	0.3264	0.1485	0.1193

TABEL III
DATA UJI COBA STATISTIK DENGAN METODE ANOVA.

Segmen	Metode	Nilai Prob > F	Uji Lanjut LSD
2	Classic <i>K-means</i>	0	Perlu
	<i>K-means</i> -ALO	0	Perlu
3	Classic <i>K-means</i>	0	Perlu
	<i>K-means</i> -ALO	0	Perlu
4	Classic <i>K-means</i>	0	Perlu
	<i>K-means</i> -ALO	0	Perlu
5	Classic <i>K-means</i>	0	Perlu
	<i>K-means</i> -ALO	0	Perlu
6	Classic <i>K-means</i>	2.3233	Tidak Perlu
	<i>K-means</i> -ALO	0	Perlu
7	Classic <i>K-means</i>	1.3980	Tidak Perlu
	<i>K-means</i> -ALO	0	Perlu

TABEL IV
DATA JUMLAH *CLUSTER* TERBAIK PADA MASING-MASING SEGMENT DENGAN UJI LANJUT LSD.

Segmen	Metode	K terbaik
2	Classic <i>K-means</i>	K = 3
	<i>K-means</i> -ALO	K = 3
3	Classic <i>K-means</i>	K = 3
	<i>K-means</i> -ALO	K = 3
4	Classic <i>K-means</i>	K = 3
	<i>K-means</i> -ALO	K = 3
5	Classic <i>K-means</i>	K = 3
	<i>K-means</i> -ALO	K = 3
6	Classic <i>K-means</i>	-
	<i>K-means</i> -ALO	K = 5
7	Classic <i>K-means</i>	-
	<i>K-means</i> -ALO	K = 3

Dari uji coba statistik ANOVA diketahui bahwa dari ketujuh segmen data percobaan, metode *K-means* – ALO yang diusulkan memperlihatkan semua jumlah *k cluster* yang dihasilkan berpengaruh terhadap himpunan data. Sedangkan untuk metode *K-means* menunjukkan pada kedua segmen yaitu segmen 6 dan segmen 7 tidak memenuhi syarat uji ANOVA karena melebihi batas maksimum nilai Prob > F. Untuk nilai uji ANOVA yang memenuhi batas perlu dilakukan uji coba lanjut dengan metode uji lanjut LSD.

D. Analisa Uji Lanjut Least Significant Difference

Uji coba selanjutnya adalah uji coba lanjut dengan menggunakan metode LSD. Uji coba lanjut ini dilakukan untuk mendapatkan jumlah *k cluster* terbaik pada kedua metode pada setiap segmen. Uji coba lanjut ini dilakukan pada setiap segmen uji coba yang lulus pada uji coba statistik ANOVA. Contoh hasil uji lanjut menggunakan Least Significant Difference (LSD) untuk segmen 2 untuk metode *K-means* - ALO menghasilkan enam kelompok dari percobaan jumlah *k cluster*. K = 3 menjadi jumlah *cluster* terbaik dengan nilai rata-rata *silhouette index* sebesar 0.6384 dan standard error sebesar 8.2026E-04. Hasil dari LSD, semua p-value yang dihasilkan bernilai dibawah 0.05 yang berarti jumlah *k cluster* ini merupakan jumlah *cluster* yang paling signifikan diantara yang lain. Semakin besar nilai rata-rata *silhouette index* maka semakin bagus kualitas *cluster* yang dihasilkan.

Hasil uji lanjut dengan metode LSD yang menunjukkan jumlah *cluster* terbaik pada masing-masing segmen akan ditampilkan pada Tabel IV dengan hasil berupa perlu atau tidaknya dilakukan uji lanjut LSD.

V. KESIMPULAN

Optimasi centroid *K-means* menggunakan *Ant Lion Optimizer* belum bisa meningkatkan performa pengelompokan data pada semua segmen karena similaritas data yang berbeda pada masing-masing segmen, yang dibuktikan dengan jumlah *cluster* optimum dari pengujian statistik ANOVA. Perbaikan data sebelum diproses dan menghilangkan *noise* masih sangat perlu dilakukan. Di sisi lain eksplorasi metode pengelompokan data serta optimasi juga dapat membantu untuk menghasilkan nilai validitas *cluster* yang lebih optimal lagi.

DAFTAR PUSTAKA

- [1] M.Y. Pusadan, J. L. Buliali, and R. V. H. Ginardi, "Anomaly detection of flight routes through optimal waypoint," *J. Phys. Conf. Ser.*, vol. 755, p. 011001, hal. 1-8, 2016.
- [2] X. Tang, J. Gu, Z. Shen, and P. Chen, "A flight profile *clustering* method combining twed with *K-means* algorithm for 4D trajectory prediction," dalam *Integr. Commun. Navig. Surveill. Conf.*, Herdon, VA, USA, 2015, hal. S3-1-S3-9.
- [3] S. Mirjalili, "The ant lion optimizer," ScienceDirect *Adv. Eng. Softw.*, vol. 83, hal. 80–98, Mei. 2015.
- [4] S. K. Majhi and S. Biswal, "Optimal *cluster* analysis using hybrid *K-means* and Ant Lion Optimizer," ScienceDirect *Karbala Int. J. Mod. Sci.*, vol. 4, no 1, hal. 347-392, Des. 2018.
- [5] Y. Liu, H. Xiong and Z. Li, "Understanding and Enhancement of Internal *Clustering* Validation Measures," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, hal. 982–994, Jun. 2013.
- [6] D. L. Davies and D. W. Bouldin, "A *Cluster* Separation Measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, hal. 224–227, Apr. 1979.
- [7] T. Calinski and J. Harabasz, "A dendrite method for *cluster* analysis," *Commun. Stat. - Theory Methods*, vol. 3, no. 1, hal. 1–27, 1974.
- [8] I P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of *cluster* analysis," ScienceDirect *J. Comput. Appl. Math.*, vol. 20, hal. 53–65, Nov. 1987.
- [9] F. Martel, R. R. Schultz, W. H. Semke, Z. Wang, and M. Czarnomski, "Unmanned Aircraft Systems Sense and Avoid Avionics Utilizing ADS-B Transceiver," dalam *AIAA Infotech@Aerospace*, Seattle, Washington, 2009, hal. 1–8.
- [10] M. Gariel, F. Kunzi, and R. J. Hansman, "An algorithm for conflict detection in dense traffic using ADS-B," dalam *IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, hal. 1–12.
- [11] M. Orefice, V. D. Vito, F. Corrado, G. Fasano, and D. Accardo, "Aircraft conflict detection based on ADS-B surveillance data," dalam *IEEE Metrology for Aerospace (MetroAeroSpace)*, 2014, hal. 277–282.
- [12] K. Y. Baek and H. C. Bang, "ADS-B based Trajectory Prediction and Conflict Detection for Air Traffic Management," *Int. J. Aeronaut. Sp. Sci.*, vol. 13, no. 3, hal. 377–385, 2012.
- [13] A. K. Jain, "Data *clustering*: 50 years beyond *K-means*," ScienceDirect *Pattern Recognit. Lett.*, vol. 31, no. 8, hal. 651–666, Jun. 2010.
- [14] L. J. Deborah, R. Baskaran, and A. Kannan, "A Survey on Internal Validity Measure for *Cluster* Validation," *Int. J. Comput. Sci. Eng. Surv.*, vol. 1, no. 2, hal. 85–102, Nov. 2010.
- [15] A. Vij and P. Khandnor, "Validity of internal *cluster* indices," dalam *Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS*, 2016, hal. 388–395.
- [16] J. Miller and P. Haden, "Analysis of Variance," dalam *Statistical Analysis with The General Linear Model*, San Fransisco, USA, 2006, bab. I, hal. 7–117.
- [17] R. a. Fisher, "The Principles of Experimentation Illustrated by A Psycho-Physical Experiment", dalam *The Design of Experiment.pdf*, London, 1935, bab. II, hal.13-26.