

# PENGELOMPOKAN CITRA KUPU-KUPU MENGGUNAKAN ALGORITMA AGGLOMERATIVE HIERARCHICAL CLUSTERING

Albert Handrijono

UDINUS, fakultas ilmu komputer

Jl. Lumbang Sari IV/D02, Semarang, 50135, 6285950502947

E-mail : [bezaleelalbert@gmail.com](mailto:bezaleelalbert@gmail.com), [emailpenulis2@domain.ekstensi](mailto:emailpenulis2@domain.ekstensi)

## Abstrak

Penelitian berisikan tentang ekstraksi, pengelompokan gambar dan berikut analisa hasil pengelompokannya. Pada penelitian ini citra kupu-kupu yang terdiri dari citra non-masking dan masking diekstraksi dengan GLCM menggunakan parameter jarak 1 dan 2 serta sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  yang menghasilkan 8 offset. Setelah itu citra dikelompokkan menggunakan hierarchical agglomerative clustering menggunakan euclidian distance dan metode linkage average serta metode single, complete dan ward untuk offset dengan rata-rata purity dan cophenet tertinggi. Analisa dilakukan dengan nilai purity dan cophenet serta diagram scatter untuk membantu analisa. Hasil pengelompokan ke delapan offset dari citra masking dengan citra non-masking dibandingkan. Penelitian bertujuan untuk melakukan evaluasi kinerja algoritma GLCM dan hierarchical clustering pada database kupu-kupu menggunakan nilai perhitungan purity dan cophenet.

**Kata Kunci:** pengelompokan, hierarki clustering, kupu-kupu,

## Abstract

This research is about image extraction, images clustering and also the analysis of the results. In this research, butterfly images that consist of non-masking and masking images was extracted with GLCM using parameter of distance with 1,2 and direction with  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  which produce 8 offsets. After that the images was clustered using hierarchical agglomerative clustering using euclidian distance and average linkage method and single, complete and ward methods for offset with the best average of purity and cophenet. Analysis was done with purity and cophenet as well scatter diagrams to help the analysis. The results of the 8 offsets from masking and non-masking was compared. The research aims to evaluate GLCM and hierarchical clustering algorithm performance on the butterfly database using purity and cophenet.

**Keywords:** clustering, hierarchical clustering, butterfly

## 1. PENDAHULUAN

Teknologi informasi sekarang ini telah memberi pengaruh besar pada beberapa aspek kehidupan. Adanya teknologi informasi tidak hanya mempermudah proses komunikasi antar manusia namun juga membantu mempermudah

bahkan beberapa menggantikan peran manusia dalam melakukan berbagai macam pekerjaan. Karena hal ini teknologi informasi mengalami perkembangan yang sangat pesat. Berbagai macam studi mengenai pemanfaatan teknologi informasi di berbagai bidang dilakukan untuk

memecahkan masalah yang beberapa diantaranya sebelumnya tidak dapat dilakukan oleh manusia.

Terdapat berbagai macam spesies kupu-kupu di dunia ini. Menurut Gillott [1], kupu-kupu yang sudah dideskripsikan di dunia sekitar 200.000 spesies, 11.300 spesies dari Amerika Utara, 10.000 spesies dari Australia, dan 2.500 spesies dari Inggris. Sementara pada [2] dijelaskan ada sekitar 150.000 spesies kupu-kupu yang sudah dideskripsikan. Spesies kupu-kupu tersebut dapat dikatakan perbedaannya melalui beberapa ciri fisik seperti ukuran tubuh, antena dan terutama ciri pola pada sayap.

Berkembangnya teknologi informasi khususnya pada bidang computer vision, membuat pengenalan suatu objek yang memiliki ciri tertentu dapat dilakukan. Teknologi computer vision ini membuat pengenalan pada berbagai jenis hewan dan tumbuhan juga dapat dilakukan.

Serangga kupu-kupu pada masing-masing spesies memiliki ciri pola pada sayap yang berbeda satu sama lain. Sebagai hasil pengenalan kupu-kupu menggunakan teknologi computer vision dapat dilakukan menggunakan ciri pola sayap pada spesiesnya. Dengan mampu dikenalnya spesies kupu-kupu ini secara terkomputasi diharapkan dapat membantu pembelajaran akan spesies kupu-kupu tanpa perlu melakukan pembelajaran terlebih dahulu.

Image retrieval merupakan bagian dari computer vision. Image retrieval merupakan salah satu bidang pada teknologi informasi yang terus dilakukan studi. Image retrieval merupakan julukan untuk sistem

komputer yang mampu melakukan pencarian atau pengambilan gambar dari sekumpulan data atau dataset gambar. Image retrieval memerlukan informasi atau data untuk dapat dilakukan. Pada metode image retrieval tradisional memerlukan suatu sekumpulan kata kunci untuk di labelkan pada masing-masing gambar pada database tersebut untuk dapat dilakukan. Karena proses melabelkan key data tersebut harus dilakukan manual maka dikembangkanlah metode untuk melakukan ekstraksi anotasi gambar secara otomatis seperti GLCM (Gray Level Co-Occurance Matrix).

GLCM adalah metode ekstraksi fitur tekstur pada gambar yang penulis gunakan pada penelitian ini. Menggunakan GLCM elemen tekstur pada gambar-gambar pada suatu database dapat di ekstraksi menjadi informasi digital yang dapat diproses oleh sistem komputer. Terdapat 4 fitur hasil ekstraksi GLCM yang digunakan pada penelitian ini yaitu *contrast*, *homogeneity*, *energy*, *correlation*. GLCM pada penelitian ini dilakukan dengan menggunakan parameter 4 sudut ( $\theta$ )=  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  dan 2 distance ( $d$ )= 1, 2. Dari kedua parameter tersebut ekstraksi GLCM ini akan menghasilkan 8 offset dengan setiap offset mewakili sudut tertentu dan distance tertentu yang berisikan nilai dari ke 4 fitur hasil ekstraksi GLCM. Menggunakan informasi digital ini pengelompokan atau clustering akan dapat dilakukan.

Clustering adalah proses pengelompokan data-data pada suatu database. Clustering sering disebut juga unsupervised learning karena pengelompokan dilakukan oleh mesin atau komputer hanya didasarkan pada objek dan fitur yang dimilikinya, objek-objek tersebut akan dikategorikan ke

dalam beberapa group. Tidak ada label untuk mengarahkan objek seperti apa yang harus masuk ke grup tertentu seperti halnya pada klasifikasi. Pada masing-masing algoritma clustering yang membedakan adalah cara atau aturan untuk menghitung jarak antar objek dan jarak antar cluster. Pada penelitian ini clustering dilakukan menggunakan algoritma hierarchical clustering. Hierarchical clustering bekerja dengan membandingkan kemiripan antar sekumpulan objek yang berisi objek-objek. Hasil akhir dari hierarchical clustering dapat digambarkan kedalam bentuk dendogram. Ada beberapa algoritma untuk menganalisanya, yang digunakan pada penelitian ini adalah purity dan cophenet. Diagram scatter juga digunakan untuk membantu menganalisa hasil akhir pengelompokan pada penelitian ini.

Database citra kupu-kupu yang digunakan pada penelitian ini terdiri dari 10 spesies dengan jumlah total 832 file. Citra yang digunakan berasal dari internet dan memiliki resolusi satu sama lain yang berbeda. Sebagai tahap awal database kupu-kupu tersebut akan diekstraksi fitur teksturnya menggunakan GLCM yang menghasilkan 8 offset. Pada tahap kedua menggunakan objek beserta fitur-fitur teksturnya pada setiap dari 8 offset tersebut dilakukan clustering menggunakan hierarchical clustering untuk kemudian dianalisa menggunakan algoritma purity dan cophenet. Offset dengan rata-rata gabungan purity dan cophenet terbaik akan di uji lagi menggunakan tiga metode linkage lain untuk didapatkan nilai purity dan cophenet nya guna memperluas analisa. Diagram scatter dan dendogram juga digunakan untuk menggambarkan sekaligus menganalisa hasil

pengelompokan.

Penelitian ini bertujuan untuk melakukan evaluasi kinerja algoritma hierarchical clustering untuk pengelompokan pada citra kupu-kupu yang didapat dengan menganalisa nilai purity dan cophenet antara citra non-masking dan citra masking pada kedelapan offset serta menggunakan beberapa diagram seperti diagram scatter atau dendogram untuk membantu penggambaran hasil pengelompokan.

## **2. METODE**

### **2.1 Metode Usulan**

Pada penelitian ini penulis dalam melakukan clustering pada gambar yang digunakan melalui beberapa langkah atau metode. Diawali dari dataset kupu-kupu yang terdiri dari non-masking dan masking pada keduanya dilakukan ekstraksi ciri atau feature extraction menggunakan GLCM. Hasil dari ekstraksi ciri tersebut dapat digunakan dalam melakukan clustering menggunakan hierarchical clustering.

### **2.2 Gray Level Co-occurrence Matrix (GLCM)**

Gray Level Co-occurrence (GLCM) pada [7] didefinisikan sebagai tabulasi dari seberapa sering kombinasi yang berbeda dari nilai kecerahan piksel (tingkat warna abu-abu) muncul pada suatu gambar.

Gray Level Co-occurrence Matrix (GLCM) merupakan metode yang paling banyak digunakan untuk melakukan analisis tekstur. Metode ini diperkenalkan oleh Haralick di tahun 1973 yang merupakan bagian dari project yang didukung oleh NASA yaitu tepatnya NASA Goddard Space Flight

Center. Haralick menggunakan citra dari NASA ERTS untuk melakukan klasifikasi dengan tingkat akurasi diatas 80%.

Pada GLCM tekstur mempertimbangkan relasi antara kedua pixel pada waktu yang sama, pixel pertama disebut reference pixel dan yang kedua disebut neighbour pixel penentuan pixel pertama dan kedua ditentukan sesuai arah sudut yang digunakan.

Relasi antara reference dan neighbour pixel dapat ada di salah satu dari ke 8 arah atau sudut(N, NE, E, SE, S, SW, W, NW). Hanya setengah dari arah ini yang sebenarnya digunakan, karena seperti N berlawanan dengan S terdapat cara yang lebih mudah menghitung untuk arah N daripada harus menghitungnya secara terpisah. Berikut adalah ilustrasi yang menggambarkan arah sudut dengan jarak 1 pixel dan ada 4 jenis arah yang digunakan antara lain 0=180, 45=225, 90=270, dan 135=315.

Perhitungan texture menggunakan GLCM dijelaskan pada [7] dapat dilakukan menggunakan beberapa bentuk yang disebut order. terdapat tiga order antara lain:

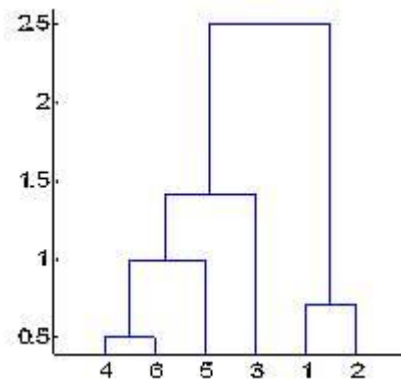
1. first order, perhitungan tekstur adalah statistik hasil dari perhitungan nilai asli suatu gambar, seperti varian, dan tidak mempertimbangkan hubungan dengan neighbour pixel.
2. second order, perhitungan dilakukan dengan mempertimbangkan hubungan antara satu pixel dengan tetangganya pada sumber gambar.
3. third order atau lebih tinggi, perhitungan tekstur mempertimbangkan hubungan antara tiga pixel atau lebih. Secara teoritis ini mungkin dilakukan namun tidak secara umum diimplementasikan karena masalah waktu kalkulasi dan kesulitan

interpretasi.

### 2.3 Hierarchical Clustering

*Hierarchical clustering* merupakan salah satu algoritma untuk melakukan clustering. Pada *hierarchical clustering* dilakukan kategorisasi objek ke suatu hierarki seperti pohon diagram (bercabang-cabang) yang disebut *dendogram*.

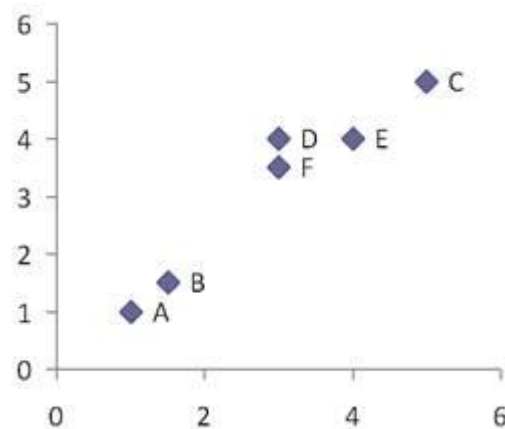
Berikut contoh dari *dendogram*:



Gambar 1. Deondogram

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Gambar 2. Fitur Objek



Gambar 3. Visualisasi Letak Objek

Berikut adalah contoh pengilustrasian

algoritma dari *hierarchical clustering*. Semisal kita memiliki 6 objek (dinamai A, B, C, D, E dan F). Setiap objek memiliki dua fitur (X1 dan X2). Kita dapat menggambarkan letak dari fitur tersebut untuk mendapat gambaran dari jarak antar objek.

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Gambar 4. Distance Matrix

Untuk menghitung jarak antar objek dapat digunakan beberapa algoritma seperti *euclidian distance* dan *manhattan distance*. Menggunakan algoritma tersebut jarak antar objek tersebut akan dihitung dan hasilnya dituliskan menjadi *distance matrix*. Rumus dari *Euclidian distance* dituliskan seperti berikut:

$$d_{ij} = \left( \sum_k (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}} \quad (2.1)$$

Euclidian  
Distance

Sebagai contoh, jarak antara objek A = (1, 1) dan B = (1.5, 1.5) dihitung:

$$d_{AB} = \left( (1-1.5)^2 + (1-1.5)^2 \right)^{\frac{1}{2}} = \sqrt{\frac{1}{2}} = 0.7071$$

contoh lain objek D = (3, 4) dan F = (3, 3.5) dihitung

$$d_{DF} = \left( (3-3)^2 + (4-3.5)^2 \right)^{\frac{1}{2}} = 0.5$$

Menggunakan cara tersebut maka semua jarak antar objek dapat dihitung dan kemudian dituliskan kedalam matrix yang disebut *distance matrix*. Karena matrix tersebut bersifat simetris(jarak antara A dan B sama dengan jarak antara B dengan A), maka perhitungan hanya perlu dilakukan pada setengah bagian atas atau bawah matrix(bagian dengan warna hijau atau merah muda). Elemen diagonal yang ada pada *distance matrix* memiliki nilai 0 menunjukkan jarak menuju objek itu sendiri.

Semisal jumlah objek adalah  $m$ , maka jumlah dari elemen jarak yang terdapat pada setengah atas atau bawah matrix adalah  $\frac{1}{2}m(m-1)$ . Sesuai contoh semisal terdapat 6 objek, maka total elemen jarak yang perlu dihitung adalah  $\frac{1}{2}m(m-1) = \frac{1}{2} * 6 * 5 = 15$

15 elemen tersebut antara lain adalah

0.71	5.66	3.61	4.24	3.20	4.95	2.92	3.54	2.50	2.24	1.41	2.50	1.00	0.50	1.12
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

Pada hierarchical cluster objek-objek yang ada akan dikelompokkan pada cluster-cluster. Dari objek-objek yang sudah ada maka jarak antar objek tersebut dapat digunakan untuk menentukan jarak antar group atau cluster.

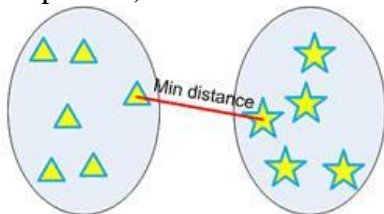
Pada tahap selanjutnya objek tersebut akan disebut cluster. Tujuan yang akan dicapai adalah mengelompokkan group tersebut sehingga pada akhir iterasi akan terdapat satu cluster yang berisi seluruh objek dari cluster-cluster di awal iterasi.

Pada langkah awal iterasi yang dilakukan adalah menemukan pasangan cluster dengan jarak paling dekat (paling mirip). Semisal  $d_{AB}$  paling dekat atau bernilai paling kecil berarti cluster A dan B paling mirip. Cluster A dan B tersebut digabung kedalam satu cluster baru yang memiliki label (AB). Lakukan penghapusan baris dan kolom yang bersesuaian dengan cluster A dan B lalu tambahkan baris dan kolom yang memberikan jarak-jarak antara cluster (AB) dengan cluster yang tersisa.

Ulangi langkah tersebut sebanyak jumlah objek minus 1 kali maka pada iterasi terakhir semua objek akan berada dalam satu cluster. Selanjutnya catat cluster-cluster yang digabungkan dan jarak atau tingkat kemiripan di mana penggabungan terjadi.

Untuk menentukan jarak antara kedua cluster terdapat beberapa kriteria perhitungan yang dapat digunakan, kriteria tersebut antara lain:

Single linkage (minimum distance / jarak terdekat)



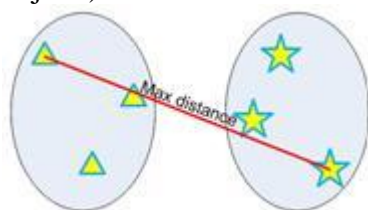
Gambar 5. Single Linkage

Single linkage equation:

$$d_{(UV)W} = \min\{d_{UW}, d_{VW}\} \quad (2.2)$$

Jarak antara kedua cluster diambil dari jarak terdekat antara dua objek dari kedua cluster.

Complete Linkage (maximum distance / jarak terjauh)



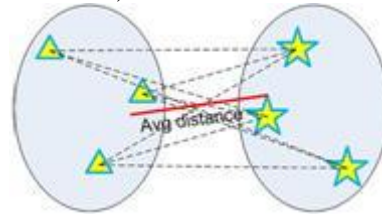
Gambar 6. Complete Linkage

Complete linkage equation:

$$d_{(UV)W} = \max\{d_{UW}, d_{VW}\} \quad (2.3)$$

Jarak antara kedua cluster diambil dari jarak terjauh antara objek dari kedua cluster.

Average Group (average distance / jarak rata-rata)



Gambar 7. Average Group

Average group equation:

$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)} N_W} \quad (2.4)$$

Jarak antara kedua cluster diambil dari jarak rata-rata antara objek dari kedua cluster.

### 3. HASIL DAN PEMBAHASAN

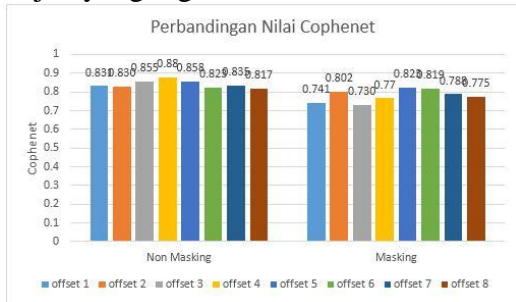
Untuk membantu menganalisa hasil pengelompokan digunakan nilai *purity*. *Purity* menjumlahkan nilai terbesar dari masing-masing cluster dan kemudian membaginya dengan jumlah keseluruhan data yang digunakan. Berikut adalah nilai *purity* dari kedelapan offset dari citra *non-masking* dan *masking*.



Gambar 8. perbandingan purity

Pada hierarchical clustering jarak antar kedua cluster yang digambarkan oleh tinggi link dinamakan *cophenetic distance*. Clustering dapat dinyatakan valid jika objek-objek yang dihubungkan memiliki korelasi yang kuat dengan jarak objek satu sama lain. Fungsi *cophenet* akan membandingkan kedua nilai ini dan kemudian

menghitung korelasinya mengembalikan hasil nilai yang disebut *cophenetic correlation coefficient*. Semakin dekat nilainya ke 1, maka semakin akurat solusi clustering tersebut menggambarkan dataset atau objek yang digunakan.

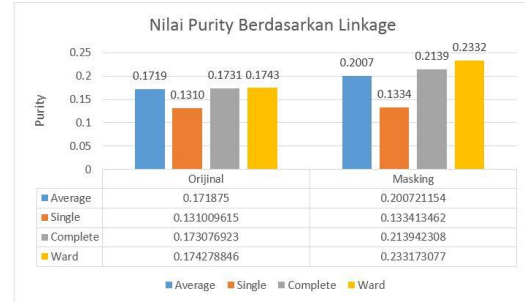


Gambar 9. perbandingan cophenet

Dari gambar 8 dan 9 nilai *purity* pada kesemua offset pada citra *masking* memiliki nilai yang lebih baik daripada offset yang sama pada citra *non-masking* sehingga rata-rata nilai *purity* pada citra *masking* 0.19171 lebih tinggi dari nilai rata-rata *purity* 0.17773 pada citra *non-masking*. Sedangkan nilai *cophenet* pada setiap offset pada citra *masking* bernilai lebih rendah dibandingkan nilai *cophenet* pada offset yang sama pada citra *non-masking* sehingga rata-rata nilai *cophenet* pada citra *masking* 0.78067 bernilai lebih rendah daripada nilai rata-rata *cophenet* 0.84086 pada citra *non-masking*.

Dari gambar 8 dan 9 tersebut juga dapat dilihat pada citra *non-masking* dengan nilai *purity* tertinggi dimiliki oleh offset 6, lalu nilai *cophenet* tertinggi dimiliki oleh offset 4, dan jika dirata-rata dari nilai keduanya maka offset 4 memiliki nilai tertinggi. Pada citra *masking* *purity* tertinggi dimiliki oleh offset 6 dan offset 8 yang kebetulan memiliki nilai yang sama persis, lalu nilai *cophenet* tertinggi dimiliki oleh offset 5, dan jika dirata-rata dari nilai keduanya maka offset 6 memiliki nilai rata-rata tertinggi.

Dari offset yang memiliki nilai rata-rata gabungan tertinggi yakni offset 4 pada citra *non-masking* dan offset 6 pada citra *masking* diujikan untuk algoritma penentuan *linkage* lainnya yaitu *single*, *complete* dan *ward*. Hasilnya sebagai berikut



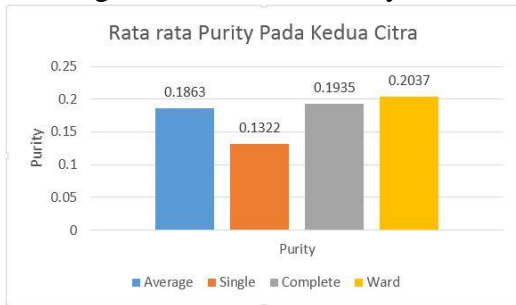
Gambar 10. perbandingan purity berdasarkan metode linkage

Pada gambar 10 dapat dilihat pada citra *masking* hasil nilai *purity* mengalami kenaikan pada semua metode dibandingkan citra *non-masking*. Pada metode *single* nilai *purity* 0.1334 pada citra *masking* lebih tinggi dari 0.1310 pada citra *non-masking*, pada metode *average* nilai *purity* 0.2007 pada citra *masking* lebih tinggi dari 0.1719 pada citra *non-masking*, pada metode *complete* nilai *purity* 0.2139 citra *masking* berbanding 0.1731 citra *non-masking*, dan pada metode *ward* nilai *purity* 0.2332 citra *masking* berbanding 0.1743 citra *non-masking*.

Metode *linkage average* seperti yang telah dilakukan dengan offset lainnya pada citra *non-masking* maupun citra *masking* memiliki nilai yang lebih tinggi dari metode *linkage single* namun lebih rendah dari metode *complete* dan *average*. Hasil ini menghasilkan urutan dari tinggi ke rendah yaitu *ward*, *complete*, *average*, *single*. Meskipun pada citra *masking* lebih terlihat perbedaan yang signifikan antara nilai *purity ward* dengan *complete*.

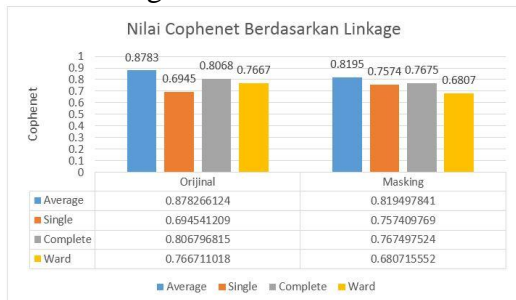
Untuk mengetahui perbedaan nilai

*purity* antar metode *linkage* pada kedua citra maka rata-rata *purity* dari masing-masing metode pada kedua citra dihitung berikut adalah hasilnya.



**Gambar 11.** perbandingan *purity* berdasarkan metode *linkage*

Pada gambar 11 dapat dilihat metode *single* memiliki nilai *purity* terendah dengan nilai 0.1322, pada urutan ketiga metode *average* dengan nilai *purity* 0.1863, pada urutan kedua metode *complete* dengan nilai *purity* 0.1935 dan metode *ward* memiliki nilai *purity* terbaik dengan nilai 0.2037.



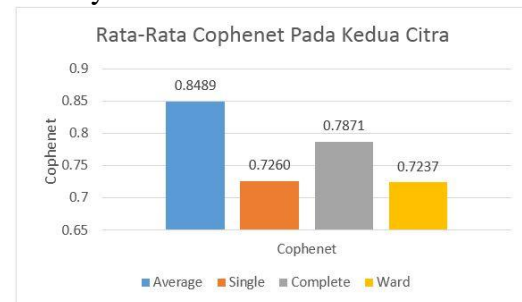
**Gambar 12.** perbandingan *cophenet* berdasarkan metode *linkage*

Pada gambar 12 menunjukkan metode *average* memiliki nilai paling tinggi dari ketiga *linkage* lain baik pada citra *non-masking* maupun citra *masking*. Pada citra *non-masking* *cophenet* hasil metode *average* memiliki nilai 0.8783 paling tinggi dibanding 0.8068 menggunakan metode *complete*, 0.7667 menggunakan *ward* dan yang paling rendah 0.6945 menggunakan metode *single*. Pada citra *masking* nilai *cophenet* 0.8195 hasil metode *average* paling tinggi dibanding 0.7675 menggunakan metode *complete*, 0.7574 menggunakan

*single* dan 0.6807 menggunakan metode *ward* sebagai nilai paling rendah.

Jika dibandingkan nilai *cophenet* dari metode yang sama dari citra *non-masking* dan *masking* dapat dilihat nilai *cophenet* pada metode *average*, *complete* dan *ward* pada citra *non-masking* memiliki nilai yang lebih tinggi dari citra *masking*. Pada metode *average* nilai *cophenet* 0.8783 pada citra *non-masking* lebih tinggi dibanding 0.8195 pada citra *masking*, pada metode *complete* 0.8068 pada citra *non-masking* lebih tinggi dibanding 0.7675 pada citra *masking*, dan 0.7667 citra *non-masking* lebih tinggi dibanding 0.6807 citra *masking*. Hanya metode *single* yang berbeda dimana nilai *cophenet* 0.7574 pada citra *masking* lebih tinggi dari nilai 0.6946 citra *non-masking*.

Untuk mengetahui perbedaan nilai *cophenet* antar metode *linkage* pada kedua citra maka nilai rata-rata *cophenet* dari masing-masing metode pada kedua citra dihitung berikut adalah hasilnya.



**Gambar 13.** perbandingan *cophenet* berdasarkan metode *linkage*

Pada gambar rata-rata *cophenet* kedua citra, dapat dilihat metode *ward* memiliki nilai *cophenet* terendah dengan nilai 0.7237, pada urutan ketiga metode *single* dengan nilai *cophenet* 0.7260, pada urutan kedua metode *complete* dengan nilai *cophenet* 0.7871 dan metode *average* memiliki nilai



*cophenet* terbaik dengan nilai 0.8489.

#### 4. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Dari hasil analisis dan pembahasan dapat di tarik kesimpulan bahwa citra masking dapat meningkatkan nilai purity pada semua offset yang ditunjukkan melalui grafik perbandingan *purity* antara citra non-masking dan masking. Yang terjadi adalah pada citra masking objek-objek lebih tersebar pada setiap cluster. Sedangkan nilai *cophenet* pada citra masking memiliki nilai yang lebih rendah pada semua offset dibandingkan nilai *cophenet* pada citra non-masking yang ditunjukkan melalui grafik perbandingan *cophenet* antara citra non-masking dan citra masking. Yang terjadi adalah pada citra masking jarak pada dendogram bertambah besar sehingga perbandingan antara jarak asli antar objek dengan jarak setelah dikelompokkan menjadi lebih besar Hal ini juga disebabkan karena purity dan *cophenet* mennghitung dua hal yang berbeda. Purity mengukur tingkat mayoritas objek yang dominan pada setiap cluster berbanding seluruh objek yang digunakan, sedangkan *cophenet* mengukur keakuratan *hierarcical clustering tree* yang dihasilkan terhadap jarak asli antar objek atau *dissimilarity* yang digunakan untuk membangun pohon itu sendiri. Pada percobaan menggunakan keempat metode linkage dapat disimpulkan metode *complete* dan metode *ward* memiliki nilai purity dan *cophenet* paling baik. Hasil pengelompokan ini dipengaruhi oleh faktor faktor antara lain resolusi pixel yang berbeda-beda, sudut pandang pengambilan citra kupu-kupu, tingkat kecerahan citra yang berbeda-beda.

##### 4.2 Saran

Berdasarkan hasil cluster citra kupu-kupu ini, dapat diberikan beberapa saran sebagai berikut, untuk meningkatkan akurasi menjadi lebih baik, sudut pandang pengambilan citra di usahakan sama, pada *dataset* harus melalui proses penyamaan resolusi piksel terlebih dahulu. Usahakan dataset memiliki tingkat pencahayaan yang sama, usahakan gambar terang namun juga tak terlalu terang. *Dataset* dapat melalui proses *cropping* (memotong) disesuaikan dengan objek yang diobservasi. Dalam penghilangan background lebih baik menggunakan segmentasi dengan background benar-benar dihilangkan dari citra.

#### DAFTAR PUSTAKA

- [1] C. Gillott, Entomology Third Edition, Springer, 2005.
- [2] P. Gullan and P. Caston, The Insects: an Outline of Entomology, 2010.
- [3] R. S. Estalita, "Kelimpahan dan Keanekaragaman Spesies Kupu-Kupu (Lepidoptera; Rhopalocera) Pada Berbagai Tipe Habitat di Hutan Kota Muhammad Sabki Kota Jambi," *Biospecies*, vol. 5, pp. 40-48, 2012.
- [4] H. Soekardi, Kupu-kupu di Kampus Unila, Lampung: Universitas Lampung, 2007.
- [5] J. Eakins and M. Graham, "Content-based Umage

- Retrieval".
- [6] E. Celebi and A. Alpkocak, "Clustering of Texture Features for Content-Based Image Retrieval," in *Advances in Information Systems*, T. Yakhno, Ed., Springer Berlin Heidelberg, 2000, pp. 216-225.
- [7] M. Hall Beyer, 21 January 2007. [Online]. Available: <http://www.fp.ucalgary.ca/mhallbey/tutorial.htm>. [Accessed 26 11 2014].
- [8] Ch.Kavitha, D. Rao and Dr.A.Govardhan, "Image Retrieval Based On Color and Texture Featuresof the Image Sub-blocks," 2011.
- [9] R. M. Haralick, K. Shanmugam and I. Dinstein, Textural Features of Image Classification, IEEE, 1973.
- [10] S. Mal-Sarkar, "Uncertainty Management Of Intelligent Feature Selection In Wireless Sensor Networks," 2009.
- [11] J. Wang, K. Markert and M. Everingham, "Learning Models for Object Recognition from Natural Language Description," Leeds, 2009.
- [12] E. Hartini, "Metode Clustering Hirarki," in *BATAN*, 2004.
- [13] V. K. Singh, N. Tiwari and S. Garg, "Document Clustering using K-means, Heuristic K-means and Fuzzy C-means," in *International Conference on Computational Intelligence and Communication Networks (CICN)*, Gwalior, 2011.