

HYBRID GENETIC ALGORITHM DAN ANT COLONY OPTIMIZATION UNTUK OPTIMISASI METODE MULTILEVEL IMAGE THRESHOLDING

Gede Aditra Pradnyana, I Putu Gede Hendra Suputra

Jurusan Ilmu Komputer, Fakultas MIPA, Universitas Udayana

Jalan Kampus Bukit Jimbaran, Badung, Bali

Email: gede.aditra@gmail.com, hendra.suputra@gmail.com

ABSTRAK

Penggunaan *genetic algorithm (GA)* sebagai metode *multilevel image thresholding* dalam segmentasi citra dapat memberikan keuntungan seperti kecepatan proses dan penentuan jumlah *threshold* serta nilai *threshold* yang tepat. Akan tetapi, *genetic algorithm* memiliki beberapa kelemahan dimana salah satunya adalah kemungkinan terjadinya konvergensi yang terlalu dini (*premature convergence*) dan tidak adanya *feedback positive* yang tidak menjamin solusi *global optimal*. Penelitian ini mengajukan metode baru *Hybrid GA-ACO* untuk optimisasi metode *multilevel image thresholding* sehingga dapat mengatasi kelemahan tersebut dengan cara menggabungkan *GA* dan *ant colony optimization (ACO)*. Penggabungan dilakukan dengan menjadikan posisi dan nilai *threshold* yang didapatkan pada *GA* sebagai nilai awal untuk proses algoritma *ACO*. Hasil pengujian dengan citra sintetis dan citra asli menunjukkan nilai *cost function*, *uniformity*, dan *misclassification error* dari metode *hybrid GA-ACO* lebih baik dibandingkan dengan algoritma awal *GA*, yaitu rata-rata 98.87% untuk tingkat *uniformity* dan 97.72% untuk nilai *ME*. Nilai *cost function* metode *hybrid GA-ACO* yang lebih kecil dibandingkan algoritma *GA* menunjukkan bahwa metode *hybrid GA-ACO* dapat mencegah konvergensi dini pada algoritma *GA*. Dari hasil tersebut dapat disimpulkan bahwa metode *hybrid GA-ACO* yang dikembangkan merupakan suatu metode *multilevel image thresholding* yang dapat mencegah konvergensi dini sehingga mencapai konvergensi pada solusi optimal yang bersifat *global optimum*.

Kata Kunci: *Genetic Algorithm, Ant Colony Optimization, Multilevel Image Thresholding*

1. PENDAHULUAN

Segmentasi citra (*image segmentation*) digunakan untuk membedakan suatu objek dengan latar belakangnya (*background*). Dalam beberapa tahun terakhir, banyak metode segmentasi yang diajukan seperti algoritma *fuzzy clustering*, *fuzzy c-means*, dan *mean shift analysis*. Diantara semua teknik segmentasi yang ada, *image thresholding* adalah metode yang paling umum dan populer digunakan karena sifatnya yang sederhana (*simplicity*), handal (*robustness*), dan akurat (*accuracy*) [1], yang melakukan konversi derajat keabuan (*grey-level*) citra ke dalam bentuk citra *binary*.

Metode *thresholding* dapat dibagi dalam dua jenis, yaitu *bi-level thresholding* dan *multilevel thresholding*. *Bi-level* adalah suatu jenis *thresholding* yang mengklasifikasikan piksel citra ke dalam dua kelompok, pertama terdiri dari piksel yang berada di bawah batas *threshold* dan yang kedua adalah sisanya. *Multilevel thresholding* memisahkan piksel ke dalam beberapa kelas atau kelompok. Piksel yang berada dalam kelas yang sama akan memiliki derajat keabuan yang berada dalam kisaran spesifik, yang diperoleh dari beberapa *threshold*. Metode *bi-level*

dan *multilevel thresholding* dapat menggunakan beberapa pendekatan yaitu pendekatan parametrik dan non-parametrik. Sejumlah metode dengan pendekatan parametrik dan non-parametrik diajukan untuk melakukan *bi-level thresholding* [1, 2].

Metode-metode yang diajukan ini jika diperluas juga dapat digunakan dalam melakukan *multilevel thresholding*. Akan tetapi seiring dengan perluasan penggunaan metode ini, jika digunakan dalam *multilevel thresholding* akan meningkatkan proses komputasi. Maka untuk mengatasi masalah ini beberapa teknik diusulkan. Seperti dalam [3], fungsi Otsu dimodifikasi untuk dapat dioptimasi dengan algoritma *fast recursive*. Metode lain dengan *pairwise nearest neighbour* sebelumnya digunakan dalam *hierarchical clustering* yang diadaptasi untuk melakukan *multilevel clustering* [4]. Dari semua *threshold* yang mungkin, salah satu *threshold* yang meningkatkan nilai fungsi objektif digunakan. *Threshold* yang dipilih kemudian dihilangkan dan *mean* dari kelas diperbaharui. Proses seleksi dan penghilangan akan terus dilakukan sampai jumlah dari *threshold* yang diinginkan terpenuhi.

Metode lain yang menggunakan *hierarchical clustering* juga diajukan oleh Arifin [5]. Pada metode ini, dihitung similaritas antar *cluster* satu dengan

cluster lain yang memiliki similaritas paling tinggi akan digabungkan. *Threshold* yang diperkirakan didefinisikan sebagai derajat keabuan dari *cluster* yang paling tinggi, yang diperoleh melalui iterasi metode ini sampai jumlah *cluster* yang diinginkan tercapai.

Metode *fast multilevel thresholding* juga diajukan oleh Yin [6] dimana *threshold* didapatkan dengan mengoptimasi nilai Otsu atau fungsi Kapur menggunakan skema iteratif. Teknik ini dimulai dari nilai *threshold* yang diambil secara acak. Implementasi metode ini sama dengan yang dilakukan oleh Luo dan Tian, dimana fungsi Kapur yang dimaksimalkan dengan *Iterated Conditional Model* (ICM) [7].

Masalah utama dari metode-metode yang disebutkan sebelumnya adalah jumlah dari *threshold* yang menentukan seberapa banyak citra akan disegmentasi tidak dapat secara otomatis ditentukan. Untuk mengatasi masalah ini, Yen dkk. [8] mengajukan kriteria baru dalam *multilevel thresholding* yang diberi nama *Automatic Thresholding Criterion* (ATC) dimana kriteria ini digunakan secara bersama-sama dengan teknik *sequential dichotomization* [9].

Konsep dari metode ini adalah histogram yang dikotomisasi menggunakan *bi-level thresholding* dan distribusi dari *variance* terbesar lebih jauh juga mengalami dikotomisasi dalam dua distribusi dengan menggunakan teknik *bi-level thresholding* yang sama. Proses dikotomisasi berlangsung sampai fungsi biaya (*cost function*) ATC mencapai minimum.

Teknik dikotomisasi merupakan algoritma yang memiliki waktu komputasi yang cepat akan tetapi merupakan teknik sub-optimal, tidak memungkinkan memperoleh nilai yang optimal (*optimal threshold value*). Untuk mengatasi masalah ini, Hammouche dkk. [10] menggunakan *genetic algorithm* (GA) yang mengoptimasi ATC yang diajukan Yen, dkk. Metode GA pada penelitian tersebut memiliki peranan yang sangat besar dalam menentukan titik-titik *threshold* pada histogram derajat keabuan, sehingga dapat menentukan jumlah *threshold* yang sesuai dan nilai *threshold* yang memadai.

Mengacu pada kualitas solusi dari GA, pada penelitian yang dilakukan oleh El-Mihoub dkk., (2006) menyarankan untuk mengkombinasikan GA dengan metode *metaheuristic* lainnya untuk meningkatkan kualitas solusi. Salah satu yang dapat digunakan adalah algoritma *ant colony optimization* (ACO). Algoritma *ant colony optimization* memiliki karakteristik *global searching* dan *positive feedback* yang dapat memperbaiki kualitas solusi. Namun kecepatan konvergensi algoritma ini sangat lambat pada awal proses karena hanya ada sedikit perbedaan feromon pada waktu tersebut [11].

Meski memiliki peranan dalam proses optimasi ATC [10], GA yang digunakan sebagai metode

dalam menentukan jumlah, posisi, dan nilai *threshold* pada segmentasi citra mungkin saja terjebak dalam solusi *local optimal*. Mengingat beberapa kelemahan GA yaitu dapat konvergen secara dini (*premature convergence*) dan tanpa *positive feedback*, sehingga tidak ada jaminan bahwa solusi yang diberikan adalah solusi yang terbaik (*global optimal*) yang tentu saja berpengaruh langsung terhadap hasil *thresholding*.

Pada *paper* ini diusulkan suatu metode *hybrid GA-ACO* dengan menggabungkan kedua algoritma tersebut yaitu GA dan ACO untuk optimisasi metode *multilevel image thresholding*. GA berperan untuk mendapatkan perkiraan solusi optimal dimana hasil perkiraan tersebut menjadi acuan bagi algoritma ACO sebagai *initial value* untuk distribusi feromon. Hal tersebut secara otomatis akan membuat algoritma ACO bekerja lebih cepat untuk mencapai solusi optimum yang *global optimal*.

2. METODE HYBRID GA-ACO UNTUK MULTILEVEL IMAGE THRESHOLDING

Metode *thresholding* sederhana tidak mampu menangani citra yang memiliki derajat keabuan yang perbedaannya sangat tipis. Oleh karena itu diperlukan suatu metode *multilevel thresholding* yang dapat membagi derajat keabuan ini kedalam subdaerah yang sesuai. *Multilevel image thresholding* merupakan suatu proses yang memotong derajat keabuan pada suatu citra menjadi beberapa region yang jelas berdasarkan beberapa titik atau nilai *threshold* [10]. Berikut adalah penjelasan mengenai proses algoritma GA serta metode *hybrid GA-ACO* dalam melakukan *multilevel image thresholding*.

2.1 Genetic Algorithm (GA)

GA adalah teknik pencarian yang dikembangkan oleh Holland yang meniru prinsip evolusi alam. Secara sederhana, GA merupakan suatu variabel yang pertama kali dikodekan sebagai binary (0 dan 1). Sebuah model proses kemudian akan menghitung fungsi objektif dari setiap kromosom dan memberikan nilai *fitness* terhadap kromosom tersebut. Nilai *fitness* yang tinggi sangat menentukan suatu kromosom untuk dapat bertahan pada generasi selanjutnya. Proses pencarian hasil dan optimasi dilakukan dengan tiga operator utama yaitu *reproduction*, *crossover*, dan *mutation*. *Crossover* atau pindah silang adalah suatu cara mendapatkan kromosom baru dengan mengkombinasikan kromosom orang tuanya, sehingga dihasilkan suatu variasi genetik. Mutasi diperlukan untuk mengembalikan gen yang hilang akibat proses *crossover*.

Proses dari GA dalam melakukan *multilevel image thresholding* terhadap suatu citra dapat

digambarkan dalam diagram alir pada Gambar 1. Misalkan citra I memiliki N buah piksel dari suatu derajat keabuan $L = \{0, 1, \dots, L-1\}$ akan diklasifikasikan ke dalam k buah kelas (C_1, C_2, \dots, C_k) dengan kumpulan *threshold* $T = \{t_1, t_2, \dots, t_{k-1}\}$. Histogram dari citra direpresentasikan dalam $h(i), i=0, \dots, L-1$, dimana $h(i)$ merupakan jumlah dari piksel dengan derajat keabuan i .

Adapun proses dari GA dalam melakukan *multilevel image thresholding* terhadap suatu citra dapat dilihat pada Gambar 1 yang diawali dengan membuat histogram dari citra tersebut. Dari histogram yang telah dihasilkan (L), akan direpresentasikan ke dalam bentuk kromosom yang berupa string biner (A), dimana $A = a_0, a_1, a_2, \dots, a_{L-1}$ dan jumlah bit nol (*zero-bits*) yang ada dalam A menentukan jumlah dari *threshold*. Kromosom dengan nilai *fitness* tertinggi (A^*) akan disimpan, dimana perhitungan nilai *fitness* menggunakan *automatic thresholding criterion* (ATC) yang diajukan oleh Yen [8]. Probabilitas dari kelas C_i , *mean* derajat keabuan dari kelas C_i , dan total *mean* derajat keabuan dari citra secara umum dapat dirumuskan sebagai berikut:

$$P_i = \sum_{j=t_{i-1}}^{t_i-1} p_j \quad (1)$$

$$m_i = \frac{S_i}{P_i} \quad (2)$$

$$\sigma_i^2 = \sum_{j=t_{i-1}}^{t_i-1} j^2 p_j \quad (3)$$

$$m = \sum_{j=0}^{L-1} j p_j \quad (4)$$

dimana $p_j = h(j)/N$ merupakan probabilitas yang dinormalisasi pada *level* j . Berdasarkan rumus diatas, *within-class variance* ($\sigma_{w_i}^2$), *between-class variance* (σ_B^2), dan *total class variance* dapat dicari dengan rumus:

$$\sigma_{w_i}^2(k) = \sum_{j=0}^{t_i-1} \sum_{j=t_{i-1}}^{t_i-1} (j - m_{i+1})^2 p_j \quad (5)$$

$$\sigma_B^2(k) = \sum_{i=1}^k P_i (m_i - m)^2 \quad (6)$$

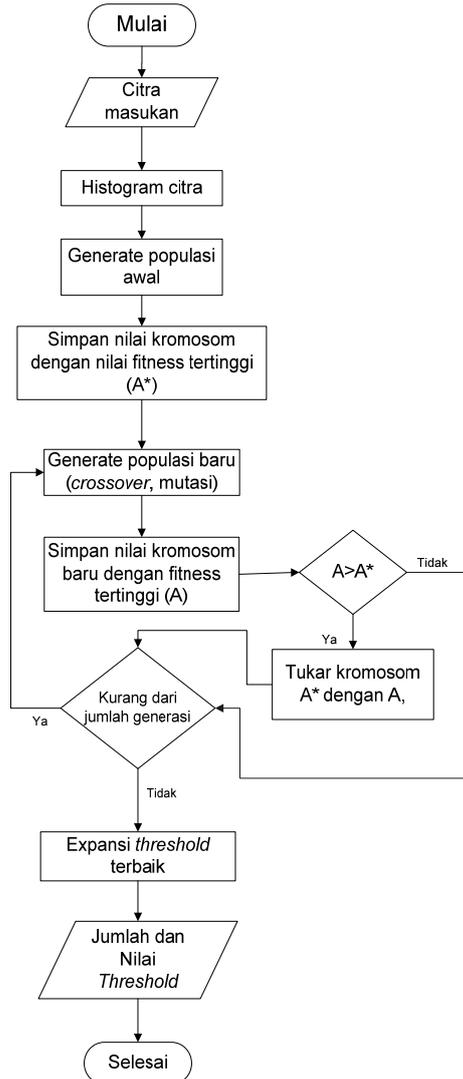
$$\sigma_T^2 = \sum_{j=0}^{L-1} j^2 p_j (j - m)^2 \quad (7)$$

Nilai *fitness* $F(k)$ dari kromosom dapat diperoleh dengan menggunakan rumus :

$$F(k) = \rho * (Disk(k))^{1/2} + (\log_2(k))^2 \quad (8)$$

dimana nilai dari variabel $Disk(k)$ merepresentasikan nilai dari *within-class variance* yang dapat diperoleh dengan

$$\sigma_{w_i}^2(k) = \sigma_T^2 - \sigma_B^2(k) \quad (9)$$



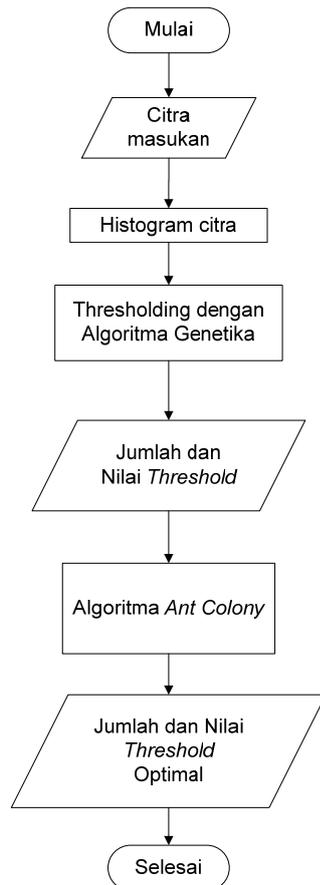
Gambar 1. Diagram Alir Proses GA dalam Melakukan Multilevel Image Thresholding

Nilai optimal dari jumlah kelas (k^*) dan jumlah *threshold* ($k^* - 1$) yang optimal dapat dirumuskan dengan :

$$F(k^*) = \min[F(k)] \quad (10)$$

Setelah menentukan kromosom terbaik, bangkitkan populasi baru dengan melakukan *crossover* dan mutasi. Proses yang dilakukan adalah *single crossover* dengan memperhatikan probabilitas *crossover* P_c . Selanjutnya bandingkan kromosom dengan nilai *fitness* tertinggi dari kromosom baru A dengan kromosom lama yang disimpan A^* . Jika A lebih dari nilai A^* , maka tukar nilai A^* dengan nilai A . Jika belum mencapai maksimum jumlah generasi kembali ke langkah 5 sampai maksimum generasi terpenuhi.

2.2 Metode Hybrid GA-ACO



Gambar 2. Diagram Alir Metode Hybrid GA-ACO

Sama seperti meta-heuristik lainnya yang terinspirasi oleh proses alam, *ant colony optimization* (ACO) merupakan algoritma yang meniru perilaku semut di dunia nyata. Dalam ACO, sebuah koloni agen sederhana, yang disebut semut buatan (*artificial ants*), mencari solusi yang baik pada setiap generasi. Setiap semut buatan pada suatu generasi membangun solusi berdasarkan nilai *state transition probability*. Setelah semua semut menghasilkan solusi, selanjutnya solusi tersebut dievaluasi sesuai dengan metode Otsu [12], kemudian algoritma akan mencatat yang terbaik yang ditemukan sejauh ini. Jejak feromon kemudian diperbarui, dan semut dari generasi berikutnya tertarik dengan feromon sehingga kemungkinan besar mereka akan mencari di dekat daerah-daerah. Prosedur ini akan diulangi sampai kriteria berhenti tercapai.

Metode penggabungan GA dengan ACO dapat dilihat pada Gambar 2. Dalam metode yang diusulkan, *output* jumlah dan nilai *threshold* yang dihasilkan oleh GA akan menjadi nilai awal dari algoritma ACO. Nilai awal tersebut sudah memiliki beberapa informasi mengenai *between-class variance* sebagai nilai feromon awal, serta jumlah dan nilai *threshold* optimal menurut GA, tugas algoritma ACO selanjutnya adalah melakukan konvergensi ulang dan meyakinkan bahwa

solusi akhir merupakan nilai yang bersifat *global optimal*.

Adapun proses dari algoritma ACO dalam menentukan *threshold* yang optimum diawali dengan menentukan nilai parameter awal (α , ρ , Q , τ_0) dan *initial* feromon. Selanjutnya hitung probabilitas perpindahan *state* dari setiap *ant* dengan menggunakan rumus :

$$P_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha}{\sum (\tau_{ij})^\alpha} & , j \in \{1, 2, \dots, UB_i - LB_i + 1\}, \\ 0 & \text{Lainnya.} \end{cases} \quad (11)$$

dimana i menyatakan *threshold* dalam *multilevel thresholding* (misal $i=1$ untuk *bi-level*, $i=2$ untuk *tri-level*, $i=3$ untuk *four-level*, dst.), sementara j merujuk pada indeks dari derajat keabuan dengan batas dari *pre-specified lower bound* dan *upper bound* dari *threshold* ke- i . Variabel τ_{ij} berisikan nilai feromon yang akan diperbaharui sesuai dengan aturan *update* feromon. *Threshold* dengan tingkat intensitas feromon yang lebih tinggi akan memiliki kesempatan yang lebih besar untuk dipilih. Pilih posisi terbaik, kemudian lakukan *update* feromon dengan rumus :

$$\tau_{ij}^{new} = \rho * \tau_{ij}^{old} + (1 - \rho) \Delta \tau^k, \quad (12)$$

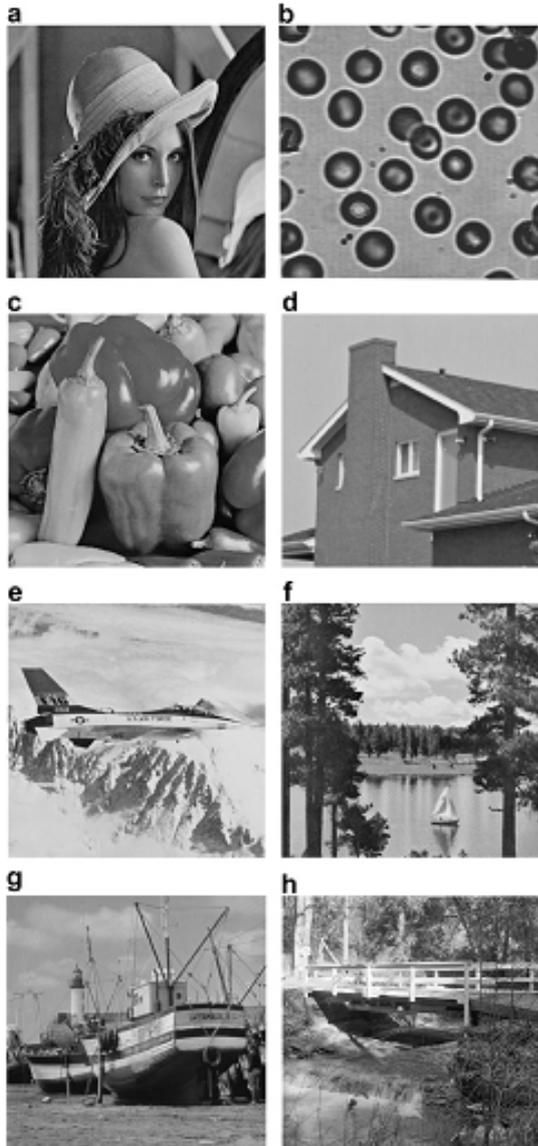
dimana parameter $\rho \in [0,1]$ melakukan *control* terhadap keberadaan feromon dan $1 - \rho$ adalah jumlah proporsi dari penguapan feromon yang terjadi. Variabel $\Delta \tau^k$ menyatakan jumlah dari feromon yang ditambahkan pada τ_{ij} dan dapat dicari nilainya dengan $\Delta \tau^k = Q * \sigma_{i,ac}^k$, dimana konstanta Q mengatur besarnya kontribusi dari feromon dan $\sigma_{i,ac}^k$ merupakan nilai dari *between-class variance*. Lakukan langkah-langkah diatas sampai *stopping criteria* terpenuhi.

3. HASIL UJI COBA DAN PEMBAHASAN

Pada bagian ini, dievaluasi performa metode *hybrid* yang diusulkan pada *multilevel image thresholding*. Percobaan dengan beberapa citra disajikan untuk mengilustrasikan *key features* dari metode yang diajukan dalam menentukan banyaknya *threshold* dan efisiensinya untuk *multilevel image thresholding*.

Metode *hybrid* GA-ACO yang diajukan diimplementasikan dengan bahasa pemrograman Matlab pada sebuah komputer dengan spesifikasi Intel Core i3 2.3 GHz, 2 GB.

3.1 Metode Pengukuran Kinerja

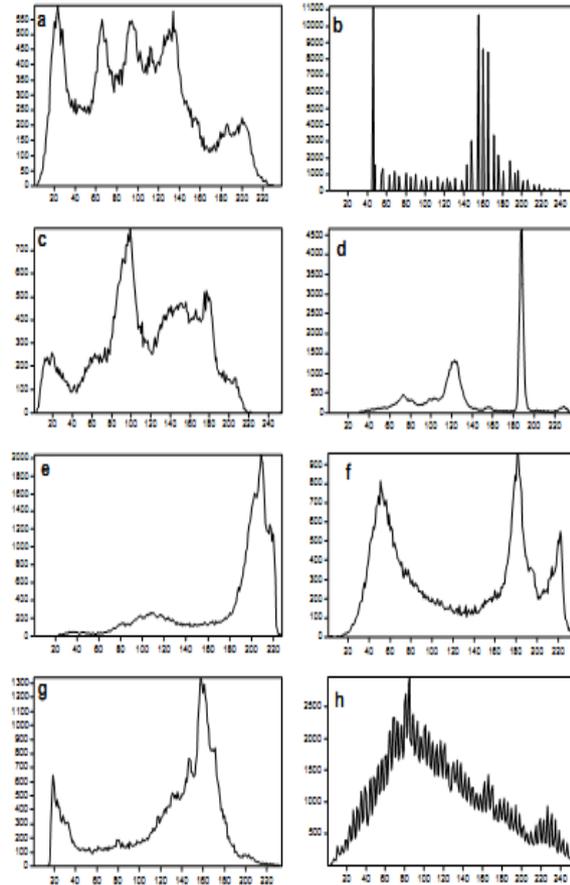


Gambar 3. Citra untuk Pengujian. (a) Lena. (b) Blood. (c) Peppers. (d) House. (e) Airplane. (f) Lac. (g) Boats. (h) Bridge.

Pengujian awal dari metode *hybrid GA-ACO* dilakukan dengan mengujikan tiga buah citra sintetis sederhana. Pengujian ini dilakukan untuk mengetahui akurasi dari metode yang diajukan berdasarkan nilai *misclassification error (ME)*. Berikut formula untuk menghitung nilai *ME* :

$$ME = 1 - \frac{|c_{0,0} - c_{0,1}| + |c_{0,1} - c_{0,2}| + \dots + |c_{0,n-1} - c_{0,n}|}{N} \quad (13)$$

Nilai *ME* menunjukkan tingkat korelasi antara citra dengan observasi manusia. Nilai tersebut mengacu pada perbandingan piksel-piksel pada suatu daerah kelas citra asli yang salah dimasukkan ke dalam suatu daerah kelas citra hasil segmentasi mulai dari kelas pertama sampai dengan kelas ke-*k* yang dibagi dengan



Gambar 4. Histogram Derajat Keabuan dari Citra yang Digunakan untuk Pengujian. (a) Lena. (b) Blood. (c) Peppers. (d) House. (e) Airplane. (f) Lac. (g) Boats. (h) Bridge.

jumlah piksel (*N*). Dimana daerah kelas citra asli dinotasikan dengan *C₀* dan *C_t* untuk daerah kelas citra hasil segmentasi.

Pengujian yang kedua adalah pengujian kinerja berdasarkan nilai *threshold* yang dihasilkan dengan membandingkan nilai *cost function (F)* dan *uniformity (U)* antara *hybrid GA-ACO* yang diajukan dengan metode *GA* yang diusulkan Hammouche dkk. [10]. Terdapat delapan buah citra yang cukup populer digunakan dalam percobaan yaitu: *Lena, Blood, Peppers, House, Airplane, Lac, Boats* dan *Bridge*. Tujuh citra pertama memiliki ukuran 256 x 256 piksel, sedangkan citra *Bridge* memiliki ukuran 512 x 512.

Gambar 3 memperlihatkan delapan citra yang digunakan sebagai citra uji. Gambar 4 memperlihatkan histogram dari delapan citra asli, yang memiliki berbagai variasi bentuk histogram mulai dari unimodal, bimodal, dan multimodal.

Pada pengujian ini, beberapa hasil pengujian dari metode *multilevel image thresholding* pada penelitian yang dilakukan oleh Hammouche dkk. [10] dianalisis dan dibandingkan dengan metode *hybrid GA-ACO* yang diajukan. Terdapat dua fungsi yang digunakan

dalam pengujian dan perbandingan yaitu *cost function* (F) dan *uniformity measure* (U). *Cost function* yang digunakan adalah *cost function* yang telah diformulasikan sesuai dengan persamaan (8). Semakin kecil nilai (F) menunjukkan kesesuaian yang ada antara hasil *threshold* citra asli dengan hasil *thresholding* dari metode yang diusulkan. Fungsi U digunakan sebagai suatu fungsi yang menyatakan kualitas suatu *threshold* citra. Berikut adalah fungsi U :

$$U = 1 - 2(k-1) \frac{\sum_{j=1}^{k-1} \sum_{i \in R_j} (f_i - m_j)^4}{N(f_{max} - f_{min})^4} \quad (14)$$

dimana $(k-1)$ adalah jumlah dari *threshold*, R_j adalah segmentasi wilayah ke- j , f_i adalah derajat keabuan pada piksel ke- i . Adapun m_j adalah rata-rata dari derajat keabuan dari setiap piksel yang tersegmentasi pada wilayah ke- j sedangkan N adalah jumlah dari total piksel pada citra.

Variabel f_{max} dan f_{min} adalah nilai maksimal dan minimal derajat keabuan pada piksel citra. Nilai dari pengukuran *uniformity* antara 0-1, dimana semakin besar nilai yang dihasilkan dari fungsi *uniformity* menunjukkan *threshold* citra yang semakin baik.

3.2. Hasil Pengujian Kinerja

Sebelum melakukan pengujian, terdapat beberapa parameter yang harus disesuaikan. Nilai awal yang mungkin untuk parameter-parameter yang digunakan pada pengujian dapat dilihat pada Tabel 1. Melalui proses *training* dengan citra uji yang terdapat pada Gambar 3 didapatkan nilai parameter paling optimal untuk algoritma GA-ACO adalah ketika $\alpha=1$, $\rho=0.9$, $Q=10^{-4}$, dan $\tau_0=10^{-2}$.

Hasil citra dari proses *multilevel thresholding* dengan algoritma GA dan metode *hybrid* GA-ACO dapat dilihat pada Gambar 8 dan 9. Dari dua buah gambar tersebut tidak terlihat jelas perbedaan antara hasil segmentasi citra berdasarkan metode GA dan *hybrid* GA-ACO, sehingga diperlukan pengukuran

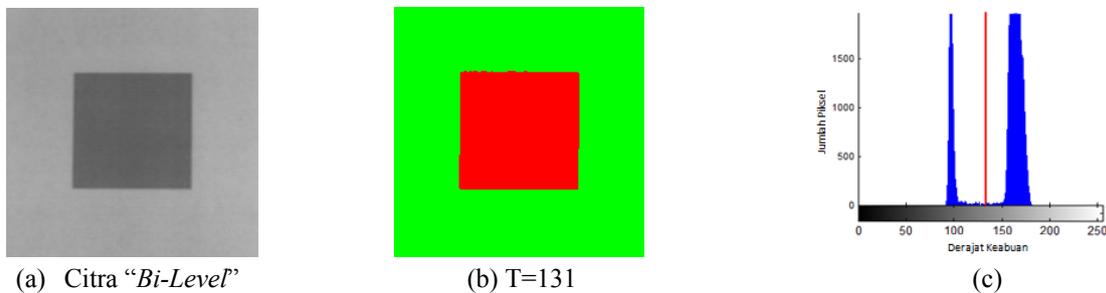
kinerja secara kuantitatif seperti yang dijelaskan pada sub bab 3.1.

Tabel 1. Nilai Awal Parameter ACO Dalam Algoritma Hybrid GA-ACO

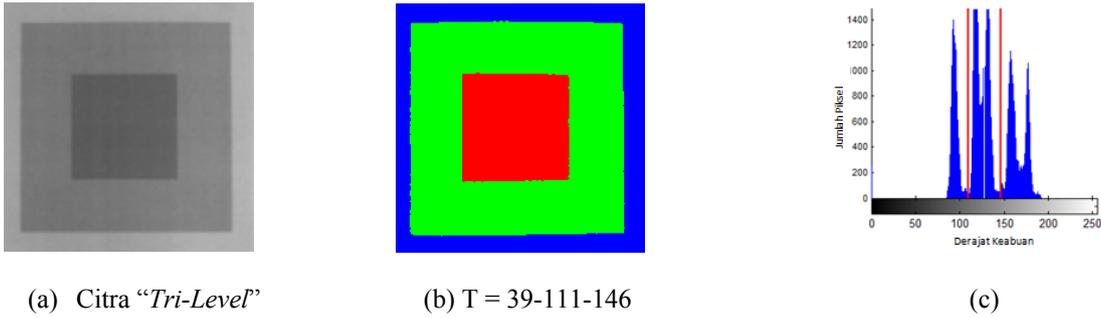
Parameter	Nilai Parameter		
α	0.5	1	2
ρ	0.1	0.5	0.9
Q	10^{-3}	10^{-4}	10^{-5}
τ_0	10^{-1}	10^{-2}	10^{-3}

Pengujian akurasi dilakukan dengan mencari nilai ME dari tiga buah *standard test images* yang merupakan objek persegi dengan tingkat keabuan yang seragam yang ditunjukkan pada Gambar 5-7. Ketiga citra tersebut masing-masing adalah citra *bi-level*, *tri-level* dan *four-level* yang berfungsi untuk memverifikasi bahwa metode yang diusulkan dapat memberikan kinerja yang berkualitas dalam segmentasi citra [13]. Dari hasil pengujian akurasi terhadap citra-citra sintetik yang telah dibuat, nilai ME yang dihasilkan untuk setiap citra uji mulai dari Gambar 5-7 adalah 0.0200, 0.0233, dan 0.0250. Nilai tersebut mengindikasikan tingkat akurasi metode *hybrid* GA-ACO untuk masing-masing citra uji tersebut adalah 98%, 97.67%, dan 97.5%. Tingkat akurasi yang baik tersebut tentu saja dipengaruhi oleh karakteristik citra sintetik yang dijadikan citra uji. Pengujian selanjutnya dengan berbagai karakteristik citra yang memiliki histogram lebih kompleks seperti yang telah dijelaskan pada bab 3 dilakukan untuk mengatasi hal tersebut.

Hasil pengukuran nilai *cost function* dan *uniformity* dari metode GA dan *hybrid* GA-ACO untuk setiap citra uji pada Gambar 3 dapat dilihat pada Tabel 2. Parameter-parameter yang dijadikan sebagai perbandingan antar metode adalah jumlah kelas, nilai *threshold*, *cost function*, *uniformity function*, dan *CPU times* (s) yang dihasilkan.



Gambar 5. Citra Uji untuk Bi-Level Thresholding: (a) Citra Asli, (c) Citra Hasil Metode Hybrid GA-ACO, dan (c) Histogram dan Posisi Threshold Optimal dari (b).

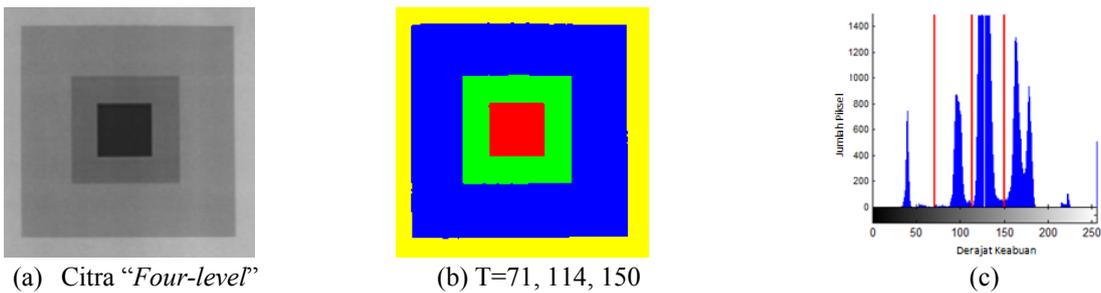


(a) Citra “Tri-Level”

(b) T = 39-111-146

(c)

Gambar 6. Citra Uji untuk Tri-Level Thresholding: (a) Citra Asli, (b) Citra Hasil Metode Hybrid GA-ACO, dan (c) Histogram dan Posisi Threshold Optimal dari (b).

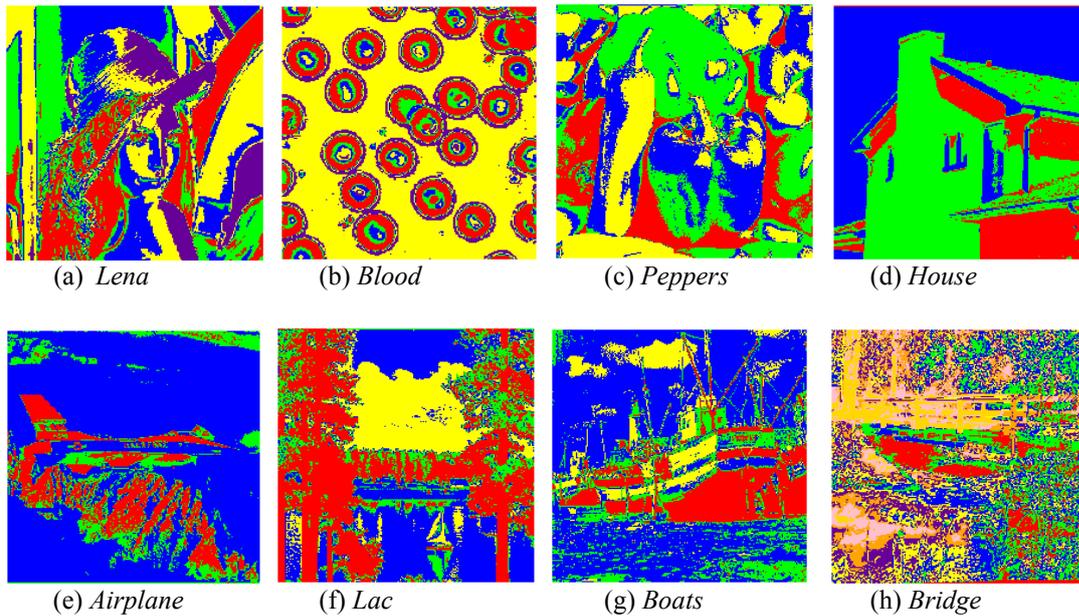


(a) Citra “Four-level”

(b) T=71, 114, 150

(c)

Gambar 7. Citra Uji untuk Four-Level Thresholding: (a) Citra Asli, (b) Citra Hasil Metode Hybrid GA-ACO, dan (c) Histogram dan Posisi Threshold Optimal dari (b).



(a) Lena

(b) Blood

(c) Peppers

(d) House

(e) Airplane

(f) Lac

(g) Boats

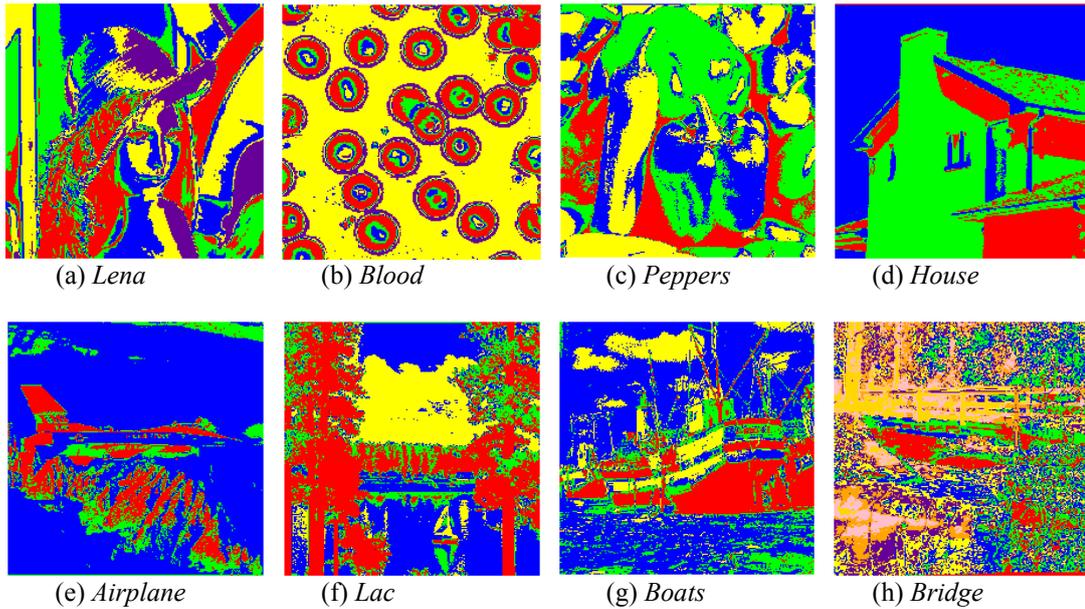
(h) Bridge

Gambar8. Citra Hasil Multilevel Image Thresholding dengan Algoritma GA

Dari hasil pengujian pada Tabel 2, dapat dilihat bahwa nilai *cost function* dan *uniformity function* metode *hybrid* GA-ACO selalu lebih optimal dari algoritma GA sebagai pembandingan pada setiap citra yang diujikan. Pada Tabel 2 dapat dilihat nilai *cost function* metode *hybrid* GA-ACO lebih kecil dibandingkan dengan algoritma GA menunjukkan citra hasil segmentasi dari metode *hybrid* GA-ACO lebih baik. Nilai *cost function* ini sangat dipengaruhi oleh

nilai *within class-variance* yang dihasilkan dari kedua metode tersebut. Pemilihan nilai *threshold* yang tepat sangat berpengaruh terhadap nilai *within class-variance*, semakin kecil nilai *within class-variance* menunjukkan tingkat keberagaman yang semakin kecil antar anggota tiap kelas yang dihasilkan dari nilai *threshold*.

Berbanding terbalik dengan nilai *cost function*, nilai *uniformity function* metode *hybrid* GA-ACO lebih



Gambar 9. Citra Hasil *Multilevel Image Thresholding* dengan Metode *Hybrid GA-ACO*

Tabel 2. Hasil Nilai *Cost Function*, *Uniformity Function*, dan *CPU Time* untuk Setiap Citra Uji

Citra	Metode	Jumlah Kelas K*	Nilai Threshold T*	Cost Function F	Uniformity Function U	CPU Time (s)
Lena	GA	5	77-115-147-182	12.3621	0.985953	3.56
	Hybrid GA-ACO	5	76-115-147-182	12.3597	0.985961	10.35
Blood	GA	5	76-131-178	12.4233	0.992149	3.62
	Hybrid GA-ACO	5	76-130-178	12.4226	0.992150	10.29
Peppers	GA	4	64-120-167	16.2638	0.984387	3.57
	Hybrid GA-ACO	4	64-120-168	16.2600	0.984390	10.42
House	GA	3	98-155	14.0209	0.993332	4.57
	Hybrid GA-ACO	3	98-157	14.0051	0.993346	11.07
Airplane	GA	3	110-172	16.4216	0.991619	3.14
	Hybrid GA-ACO	3	111-173	16.4173	0.991622	9.76
Lac	GA	4	79-141-196	15.7424	0.986892	3.52
	Hybrid GA-ACO	4	79-142-196	15.7421	0.986892	10.33
Boats	GA	4	76-130-171	15.9979	0.988183	4.13
	Hybrid GA-ACO	4	74-128-169	15.9822	0.988210	10.88
Bridge	GA	8	49-75-99-124-150-178-212	11.7589	0.987368	3.18
	Hybrid GA-ACO	8	49-76-99-123-148-176-210	11.7584	0.987370	9.99

besar dibandingkan dengan algoritma GA menunjukkan citra hasil segmentasi dari metode *hybrid GA-ACO* lebih baik. Nilai *uniformity function* menunjukkan tingkat keseragaman antar anggota tiap kelas.

Nilai *cost function* yang lebih kecil dan *uniformity function* yang lebih besar dibandingkan dengan algoritma sebelumnya yaitu segmentasi dengan GA saja menunjukkan bahwa hasil segmentasi dari GA saja belum mencapai konvergensi yang bersifat *global optimal* atau dengan kata lain mengalami konvergensi dini. Algoritma *GA-ACO* mengatasinya dengan mencari solusi yang

lebih baik yang ditunjukkan dengan meningkatnya nilai *uniformity* dan menurunnya nilai *cost function* dari solusi yang dihasilkan. Jika dilihat dari *CPU time* yang dibutuhkan, metode *GA-ACO* membutuhkan *CPU time* yang cenderung lebih lama dibandingkan dengan algoritma GA dalam menghasilkan nilai *threshold* yang optimal. Hal ini disebabkan karena pada metode *hybrid GA-ACO* terdiri dari dua algoritma utama yaitu GA dan ACO yang berjalan secara sekuensial. GA dan ACO merupakan algoritma optimasi yang memang memerlukan waktu komputasi yang cukup lama untuk menghasilkan suatu solusi. *CPU time* dari metode *hybrid GA-ACO*

tidak dipengaruhi oleh jumlah *threshold* atau jumlah kelas yang ditentukan untuk setiap citra.

4. KESIMPULAN

Pada *paper* ini, telah diusulkan sebuah metode *hybrid* GA-ACO pada *multilevel imagethresholding* yaitu dengan mengkombinasikan algoritma GA dan ACO. *Thresholding* dilakukan pada histogram dengan menjadikan GA sebagai algoritma utama yang melakukan *automatic thresholding* yang dioptimasi dengan algoritma ACO untuk optimasi kualitas solusi dari hasil *thresholding* oleh GA.

Pengujian dengan citra sintetis dan citra asli telah membuktikan bahwa metode *hybrid* GA-ACO memiliki hasil segmentasi citra yang lebih baik dibandingkan dengan algoritma awal yaitu GA, yang dievaluasi dari sisi akurasi dengan nilai *misclassification error*, *cost function*, dan *uniformity function*. Dari hasil tersebut dapat disimpulkan juga bahwa metode *hybrid* GA-ACO dapat menghasilkan suatu metode *multilevel image thresholding* yang dapat mencegah konvergensi dini yang terjadi pada algoritma GA sehingga dapat mencapai konvergensi pada solusi optimal yang bersifat global optimum.

5. DAFTAR PUSTAKA

- [1] Sezgin, M., Sankur, B., 2004. "Survey over image thresholding techniques and quantitative performance evaluation". *J. Electron. Imaging* 13 (1), 146–156.
- [2] Lievers, W.B., Pilkey, A.K., 2004. "An evaluation of global thresholding techniques for automatic image segmentation of automotive alumi-num sheet alloys". *Mater. Sci. Eng. A381*, 134–142.
- [3] Liao, P.-S., T.-S. Chen, P.-C. Chung., 2001. "A fast algorithm for multilevel thresholding". *J. Inf. Sci. Eng.* 17 713–727.
- [4] Virmajoki, O., P. Franti., 2003. "Fast pairwise nearest neighbour based algorithm for multilevel thresholding". *J. Electron. Imaging* 12 (4) 648–659.
- [5] Arifin, A.Z., Asano, A., 2006. "Image segmentation by histogram threshold-ing using hierarchical cluster analysis". *Pattern Recognit. Lett.* 27, 1515–1521.
- [6] Yin, P.Y., L.-H. Chen., 1997. "A fast iterative scheme for multilevel thresholding methods". *Signal Process.* 60, 305–313.
- [7] Luo, X., J. Tian., 2000. "Multi-level thresholding: maximum entropy approach using ICM". *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 3, pp. 778–781.
- [8] Yen, J.C., F.J. Chang, S. Chang., 1995. "A new criterion for automatic multilevel thresholding". *IEEE Trans. Image Process.* IP-4 370 –378.
- [9] Sezgin, M., Tasaltin, R., 2000. "A new dichotomization technique to multilevel thresholding devoted to inspection applications". *Pattern Recognit. Lett.* 21, 151–161.
- [10] Hammouche, K., Diaf, M., Siarry, P., 2008. "A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation". *Computer Vision Image Understanding* 109 (2), 163–175.
- [11] Dorigo, M., Stutzle, T., 2006. "The ant colony optimization metaheuristic: algorithms, applications and advances". *Technical Report IRIDIA-2000-32*.
- [12] Otsu, N., 1979. "A threshold selection method for grey level histograms". *IEEE Trans. Syst. Man Cybern.* SMC-9 (1) 62–66.
- [13] Liang, Y-C., Chen, H-L., Chyu, C-C., 2006. *Application of Hybrid Ant Colony Optimization for The Multilevel in Image Processing*. Germany:Springer Link.