
Multilevel Thresholding Segmentasi Citra Warna Menggunakan Logarithmic Decreasing Inertia Weight Particle Swarm Optimization

Multilevel Thresholding Color Image Segmentation Using Logarithm Decreasing Inertia Weight Particle Swarm Optimization

Murinto^{1*}, Adhi Prahara², Erik Iman Heri Ujianto³

^{1,2} Program Studi Informatika Universitas Ahmad Dahlan Yogyakarta

³ Program Studi Informatika Universitas Teknologi Yogyakarta

*corr_author: murintokusno@tif.uad.ac.id

ABSTRAK

Permasalahan utama dari segmentasi citra warna adalah tidak semua metode segmentasi citra yang ada saat ini dapat digunakan secara langsung seperti halnya pada citra *gray scale*. Maka dari itu diperlukan suatu teknik yang tepat untuk melakukan segmentasi warna. Teknik yang digunakan dalam penelitian ini adalah teknik segmentasi *multilevel thresholding* dengan menggunakan suatu bobot inersia *logarithm decreasing particle swarm optimization* (LogPSO). Bobot inersia Nilai *threshold* optimal diperoleh dengan cara memaksimalkan fungsi objektif Otsu. Teknik yang diusulkan mengurangi waktu perhitungan untuk perhitungan *threshold optimum* didasarkan pada *multilevel thresholding* yang diujikan pada 8 citra warna standar. Suatu analisis perbandingan secara detail dengan bobot inersia lainnya yang didasarkan pada *multilevel thresholding* yakni *constant particle swarm optimization* (CPSO), menunjukkan hasil kinerja yang lebih baik pada metode yang diusulkan. Kinerja segmentasi citra warna dalam penelitian ini didasarkan pada *peak signal to noise ratio* (PSNR), mean, standar deviasi fitness, *structural similarity index measure* (SSIM), *mean square of error* (MSE) serta waktu perhitungannya. Algoritma LogPSO menunjukkan hasil yang lebih baik pada keseluruhan parameter tersebut kecuali pada waktu penghitungan. Algoritma LogPSO lebih lama waktu perhitungannya dibandingkan dengan CPSO.

Kata-kata kunci: Citra Warna, CPSO, LogPSO, Segmentasi

ABSTRACT

The main problem of color image segmentation is that not all existing image segmentation methods can be used directly as in gray scale images. Therefore we need an appropriate technique to perform color segmentation. The technique used in this paper is a multilevel thresholding segmentation technique using an inertial weighting logarithm decreasing particle swarm optimization (LogPSO). Inertia weight Optimal threshold value is obtained by maximizing Otsu's objective function. The proposed technique reduces the calculation time for calculating the optimum threshold based on multilevel thresholding which was tested on 8 standard color images. A detailed comparison analysis with other inertial weights based on multilevel thresholding such as constant particle swarm optimization (CPSO) shows better performance results in the proposed method. Color image segmentation

Meskipun PSO sederhana dan mudah diimplementasikan pada segmentasi citra. Namun, seperti halnya teknik optimasi lainnya, salah satu kelemahan PSO adalah terjebak dalam lokal optimal. Salah satu cara untuk mengatasinya adalah dengan cara menambahkan bobot inersia (*inertia weight*) pada PSO. Bobot inersia mempunyai peran penting pada proses *trade-off* antara kemampuan diversifikasi dan intensifikasi algoritma PSO. Ketika strategi bobot inersia diimplementasikan pada algoritma PSO, maka partikel bergerak disekitar ketetanggannya sambil menyesuaikan kecepatan dan posisi mereka sesuai dengan persamaan PSO di ruang pencarian. Pada tahun 1998, Shi dan Eberhart pertama kali mengusulkan konsep konstanta bobot inersia. Bobot inersia yang kecil membantu dalam mengeksplorasi ruang pencarian sementara bobot inersia yang besar memfasilitasi dalam mengeksploitasi ruang pencarian. Bobot inersia yang diusulkan menggunakan random inertia weight (Shi and Eberhart, 1998). Pelibatan bobot inersia ke dalam PSO juga dilakukan beberapa penelitian lainnya. Strategi linearly decreasing inertia weight dikenalkan untuk meningkatkan kecepatan konvergensi algoritma PSO dalam iterasi awal ruang pencarian (Xin, Chen and Hai, 2009). Bobot inersia dimulai dengan beberapa nilai besar dan kemudian menurun secara linear ke beberapa nilai lebih kecil. Bobot Inersia memberikan hasil yang sangat baik dari 0,9 ke 0,4. Guimin *et al.* (2006) menyajikan dua strategi *natural exponent inertia weight* sebagai e1-PSO dan e2-PSO, yang didasarkan pada penurunan bobot inersia secara eksponensial. Secara eksperimental, strategi ini mengalami konvergensi prematur, meskipun kecepatan konvergensi cepat menuju posisi optimal di tahap awal dari proses pencarian. Malik *et al.* (2007) menyajikan *sigmoid increasing inertia weight (SIIW)* dan *sigmoid decreasing inertia weight (SDIW)*. Strategi-strategi ini memberikan kinerja yang lebih baik dengan kemampuan konvergensi cepat dan penyempitan gerakan agresif menuju area penyelesaian. *Oscillating inertia weight* (Li and Gao, 2009) memberikan keseimbangan antara diversifikasi dan gelombang intensifikasi dan menyimpulkan bahwa strategi ini terlihat kompetitif dan, dalam beberapa kasus, berkinerja lebih baik dalam hal konsistensi. (Gao, An and Liu, 2008) mengusulkan *exponent decreasing inertia weight (EDIW)* dengan stochastic mutation (SM). Stochastic mutation (SM) digunakan untuk meningkatkan keragaman *swarm* sementara EDIW digunakan untuk meningkatkan konvergensi kecepatan individu (Tabel 1). *Linearly decreasing inertia weight (LDIW)* telah diusulkan oleh Shi and Eberhart (1998) dan sangat meningkatkan akurasi dan kecepatan konvergensi. Bobot inersia yang besar memfasilitasi pada fase inklusif dari ruang pencarian lalu kemudian menurun secara linear menjadi bobot inersia kecil.

Dalam makalah ini dikenalkan suatu teknik *multilevel thresholding* baru berbasiskan bobot inersia pada *Particle Swarm Optimization* (PSO) untuk memecahkan masalah *multilevel threshold* pada segmentasi citra warna. Validitas teknik yang diusulkan adalah diuji pada 8 citra warna sample dan dibandingkan dengan metode lainnya yaitu *constant particle swarm optimization* (CPSO). Makalah ini disusun dalam empat bagian. Pada bagian satu dijelaskan tentang pendahuluan. Bagian kedua adalah metode penelitian yang digunakan dalam makalah ini. Bagian ketiga adalah hasil dan pembahasan, sedangkan bagian ke empat adalah kesimpulan dan saran penelitian yang bisa dilakukan selanjutnya.

METODE PENELITIAN

A. Multilevel Thresholding

Permasalahan optimisasi mempunyai tujuan untuk menemukan suatu nilai variabel yang bisa mengoptimalkan suatu fungsi objektif atau fungsi Fungsi fitness yang digunakan dalam penelitian ini didasarkan pada fungsi Otsu (Otsu, 1979). Metode Otsu pada dasarnya digunakan untuk segmentasi *bi-level thresholding* dan kurang sesuai jika digunakan pada proses segmentasi citra *multilevel thresholding*. Maka untuk mengatasi kelemahan tersebut

suatu algoritma modifikasi Otsu baru didasarkan pada *particle swarm optimization* diusulkan untuk mengatasi permasalahan *multilevel thresholding* tersebut. Tujuan akhir dari teknik yang diusulkan adalah agar dalam waktu komputasi yang lebih kecil dapat menemukan threshold-threshold yang optimal untuk melakukan segmentasi dengan cara memaksimalkan fungsi objektif Otsu.

Teknik Otsu merupakan suatu teknik *threshold* yang menggunakan variansi untuk mengukur keseragaman dari distribusi *gray level* citra. Dalam kondisi tertentu teknik ini tidak terpengaruh pada variasi kontras dan kecerahan citra dan ini diambil sebagai salah satu teknik yang efektif dari seleksi otomatis *threshold*. Prinsip dasar ini yang digunakan untuk mensegmentasi histogram ke dalam dua kelompok pada nilai tertentu, satu bersesuaian dengan latar sedangkan yang satunya lagi sesuai dengan objek. Ketika variansi diantara dua kelompok tersebut maksimum maka *threshold* optimal segmentasi diperoleh.

Dalam penelitiannya, Otsu (1979) mendefinisikan variansi antar kelas sebagai jumlahan fungsi Sigma dari tiap kelas, yang dituliskan sebagai persamaan (1).

$$\begin{aligned}
 g(t) &= \sigma_0 + \sigma_1 \dots\dots\dots(1) \\
 \sigma_0 &= w_0 + (\mu_0 - \mu_A)^2 \\
 \sigma_1 &= w_1 + (\mu_1 - \mu_A)^2
 \end{aligned}$$

Dimana $\sigma_0 = w_0 + (\mu_0 - \mu_A)^2$ dan $\sigma_1 = w_1 + (\mu_1 - \mu_A)^2$ yakni variansi kelas pertama dan variansi kelas kedua secara berurutan, μ_T adalah rata-rata intensitas dari citra asli. Di mana σ adalah variansi dan μ adalah mean.

Dalam *bilevel thresholding*, rata-rata (*mean*) level dari tiap kelas (μ_i) dituliskan sebagai persamaan (2).

$$\begin{aligned}
 \mu_0 &= \sum_{i=0}^{T-1} \frac{ip_i}{w_0} \dots\dots\dots(2) \\
 \mu_1 &= \sum_{i=T}^{L-1} \frac{ip_i}{w_1}
 \end{aligned}$$

Optimal *threshold* diperoleh dari fungsi maksimisasi antar variansi kelas, yang dituliskan sebagai persamaan (3).

$$T^* = \text{argmax } g(t) \dots\dots\dots(3)$$

Sedangkan untuk permasalahan *multitresholding* metode Otsu dapat dituliskan sebagai persamaan (4).

$$\sigma_i = w_i + (\mu_i - \mu_A)^2 \dots\dots\dots(4)$$

dengan $I = 0, 1, 2, \dots, m$.

Dalam *multilevel thresholding*, mean level dari tiap kelas (μ_i) dituliskan sebagai persamaan :

$$\mu_0 = \sum_{i=0}^{T_1-1} \frac{ip_i}{w_0}, \mu_1 = \sum_{i=T_1}^{T_2-1} \frac{ip_i}{w_1}, \mu_j = \sum_{i=T_j}^{T_{j+1}-1} \frac{ip_i}{\mu_j}, \mu_m = \sum_{i=T_n}^{L-1} \frac{ip_i}{\mu_n} \dots\dots\dots(5)$$

Optimal *multilevel threshold* diperoleh dengan cara memaksimalkan variansi antar kelas fungsi, yang dituliskan sebagai persamaan (6).

$$T^* = \text{argmax}(\sum_{i=0}^n \sigma_i) \dots\dots\dots(6)$$

B. Particle Swarm Optimization

Teknik *particle swarm optimization (PSO)* standar pertama kali dikenalkan oleh Kennedy dan Eberhart (1995) merupakan teknik optimisasi stokastik yang sama dengan tingkah laku dari sekawanan burung (*birds flocking*) atau kelakuan secara sosiologi dari sekelompok manusia. Ide dasar dari PSO adalah dengan melibatkan suatu skenario di mana sekawanan burung dalam pencarian sumber makan di dalam suatu area. Semua burung-

burung tidak tahu secara pasti di mana makanan tersebut berada, akan tetapi dengan tiap iterasi mereka akan mengetahui seberapa jauhnya makanan tersebut akan ditemukan. Strategi terbaik akan diikuti burung yang dekat dengan makanan dan juga dari posisi terbaik sebelumnya yang dicapai. PSO dibangun dengan konsep optimisasi melalui suatu partikel swarm. Tiap partikel berada pada suatu posisi dalam ruang pencarian dengan nilai *fitness*, dievaluasi melalui fungsi *fitness* untuk dioptimalkan dari tiap partikel yang merepresentasikan kualitas dari posisi tersebut. Semua partikel terbang melalui ruang pencarian multidimensi dengan menyesuaikan posisinya berdasarkan pengalamannya sendiri dan dari tetangganya.

PSO terdiri dari suatu *swarm* dari partikel yang secara random diinisialisasi sebagai titik-titik dalam ruang dimensi n pada pencarian solusi yang optimal untuk suatu permasalahan optimisasi. Dalam PSO, populasi dinamakan juga sebagai *swarm*, kandidat solusi dikodekan sebagai partikel di ruang pencarian. PSO dimulai dengan inisialisasi random dari populasi partikel-partikel. Partikel bergerak di ruang pencarian untuk mencari solusi optimal dengan memperbarui posisi setiap partikel berdasarkan pengalamannya sendiri dan partikel di sekitarnya. Selama pergerakan, posisi saat ini (*current position*) dari partikel i direpresentasikan melalui suatu vektor $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, di mana D adalah dimensi dari ruang pencarian. Kecepatan dari partikel i direpresentasikan sebagai $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. Sedangkan posisi terbaik sebelumnya (*the best previous position*) dari suatu partikel direkam sebagai terbaik individu (*the personal best*) dan dinamakan sebagai $pbest$ dan posisi terbaik yang diperoleh oleh swarm sepanjang terbaik global dinamakan sebagai $gbest$. PSO mencari solusi optimal melalui pembaruan posisi dan kecepatan tiap partikel melalui suatu persamaan berikut (Kennedy dan Eberhart, 1995).

$$v_{id}^{t+1} = v_{id}^t + c_1 * r_{1i} * (p_{id} - x_{id}^t) + c_2 * r_{2i} * (p_{gd} - x_{id}^t) \quad \dots\dots\dots(7)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad \dots\dots\dots(8)$$

Nilai t merupakan simbol iterasi ke- t dalam proses evolusioner. $d \in D$ menotasikan dimensi ke- d dalam ruang pencarian. c_1 dan c_2 adalah konstantas akselerasi, merepresentasikan pembobotan (*weighting*) dari *term* akselerasi stokastik yang mendorong tiap-tiap partikel menuju ke posisi $pbest$ (p_{id}) dan $gbest$ (p_{gd}). r_{1i} dan r_{2i} adalah nilai random yang mempunyai distribusi uniform dalam $[0,1]$. Sedangkan p_{id} dan p_{gd} merepresentasikan elemen-elemen dari $pbest$ dan $best$ dalam dimensi ke- d . v_{id} dibatasi melalui kecepatan maksimum yang didefinisikan sebelumnya, $v_{max,d}$ sampai $[-v_{max}, v_{max}]$, $pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^d)$ merepresentasikan posisi terbaik sebelumnya (posisi memberikan nilai *fitness* terbaik) dari partikel ke i . $gbest = (gbest^1, gbest^2, \dots, gbest^d)$ merepresentasikan posisi terbaik sebelumnya dari populasi.

C. Metode yang diusulkan

Suatu algoritma *particle swarm optimization* (PSO) didasarkan pada bobot inersia digunakan dalam penelitian ini. Inisialisasi posisi partikel digunakan untuk meningkatkan diversifikasi dan ergodisitas populasi. Dalam PSO suatu bobot inersia yang lebih besar memfasilitasi eksplorasi global yang memungkinkan algoritma mampu mencari area baru lebih luas, sedangkan bobot inersia yang kecil cenderung memfasilitasi eksplorasi lokal. Nilai bobot inersia yang digunakan disini berasal dari pencarian bobot inersia menggunakan *logarithm decreasing inertia weight* (LogDIW) (Gao dkk, 2008). Studi secara empirik memperlihatkan bahwa PSO dengan nilai bobot inersia (w) yang besar mempunyai kemampuan pencarian global yang baik, dibandingkan dengan w yang lebih kecil dengan konvergensi yang cepat. Dalam penelitiannya Gao (2008) memperkenalkan *logarithm decreasing inertia weight* (LogDIW) yang dituliskan sebagai persamaan (9).

$$w = w_{max} + (w_{min} - w_{max}) \times \log_{10}(a + 10t / T_{max}) \quad \dots\dots\dots(9)$$

Di mana a adalah suatu konstanta untuk penyesuaian kecepatan evolusioner, di sini $a=1$.
Persamaan perbaruan kecepatan dan posisi *chaotic* dalam PSO dapat dituliskan sebagai persamaan (10) dan Persamaan (11) secara berurutan.

- *Persamaan Perbaruan Kecepatan*

$$v_{ij}^{t+1} = w * v_{ij}^t + c_1 \cdot CF_1 * (P_{best,i}^t - x_{ij}^t) + c_2 \cdot CF_2 (G_{best} - x_{ij}^t) \quad \dots(10)$$

- *Persamaan Perbaruan Posisi*

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad \dots\dots\dots(11)$$

dimana $i = 1,2,3, \dots, n$ dan $j = 1,2,3, \dots, d$

Dimana x_{ij}^t dan v_{ij}^t menotasikan posisi dan kecepatan secara berurutan. CF_1 dan CF_2 adalah fungsi dari nilai peta variabel *chaotic*, yang menggantikan nilai-nilai random r_1 dan r_2 dalam PSO standar. Dalam Gambar 1(a) diperlihatkan diagram alir *particle swarm optimization* berdasarkan bobot inersia *logarithm decreasing inertia weight* (LogPSO) yang digunakan dalam penelitian ini. Sedangkan dalam Gambar 1(b) diperlihatkan segmentasi citra warna menggunakan algoritma PSO menggunakan LogDIW.

HASIL DAN PEMBAHASAN

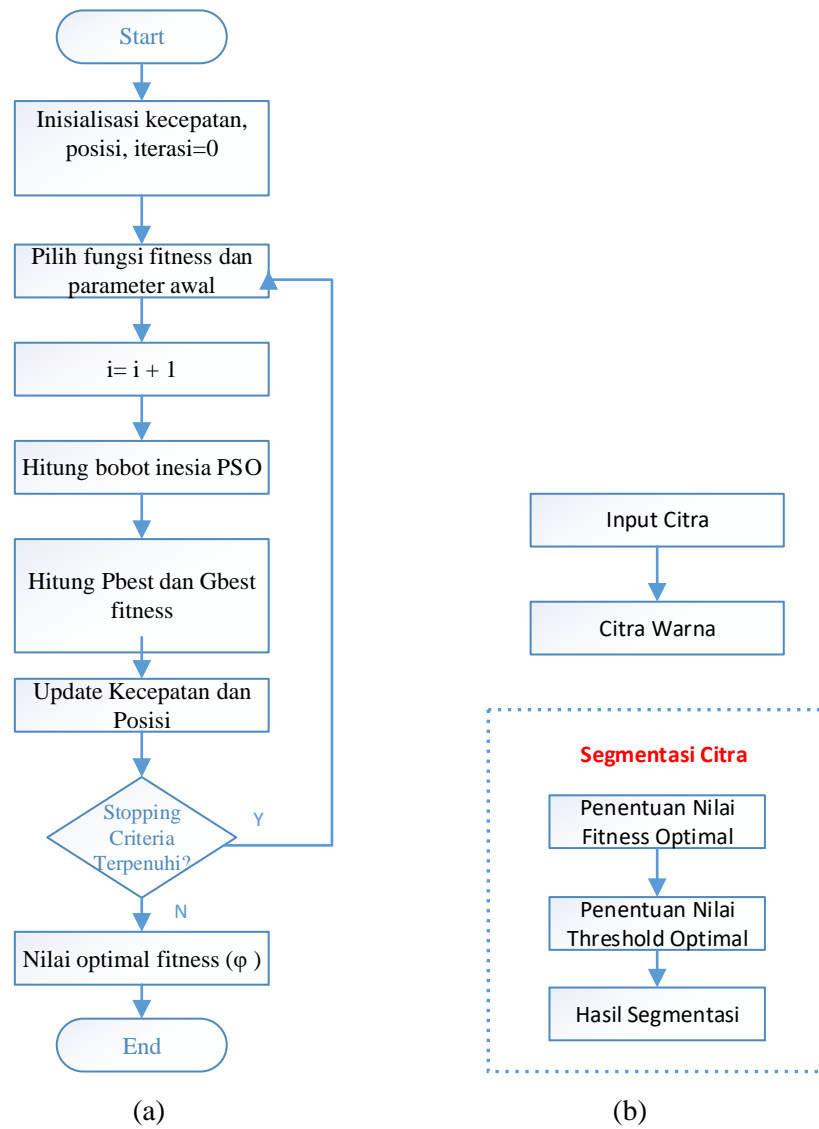
Percobaan dilakukan dengan menggunakan berbagai citra standar dari BSD500. Citra tersebut dapat diakses pada [website https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/](https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/). Algoritma LogPSO yang didasarkan pada *image thresholding* diimplementasikan pada pemrograman MATLAB. Nilai fitness dan CPU time digunakan pada *multilevel thresholding* dibandingkan dengan bio-inspired algorithm yang lain yaitu *Constant PSO* (CPSO) [18] Citra input terdiri dari 8 citra warna meliputi 8068.jpg, 16004.jpg, 16068.jpg, 22090.jpg, 28083.jpg, 29030.jpg, 36046.jpg, 41096.jpg. Citra inputan dan histogram yang bersesuaian diperlihatkan dalam Gambar 2. Ukurannya adalah 256 x 256 piksel.

Parameter algoritma LogPSO untuk *multilevel thresholding* ditentukan melalui analisis secara empiris. Nilai parameter yang digunakan dalam penelitian ini diperlihatkan dalam Tabel 1.

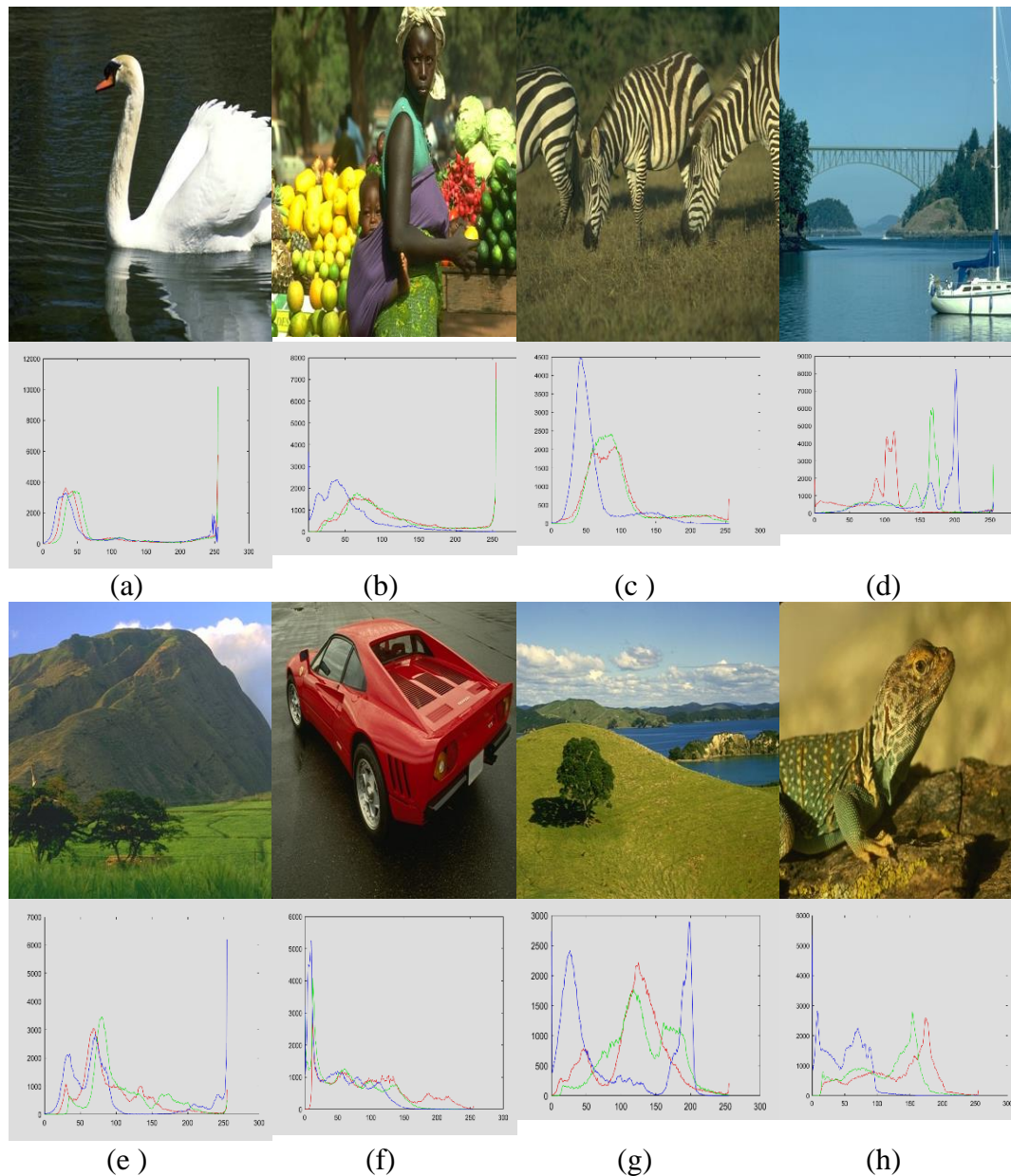
Tabel 1. Nilai Parameter CPSO dan LogPSO

Parameter	Nilai	
	CPSO	LogPSO
Swarm size	30	30
Iteration	500	500
run	20	20
Bobot Inersia	1	$(w_{max}, w_{min}) = (0.9, 0.4)$
c_{max}, w_{min}	1	1

Dalam eksperimen ini fungsi Otsu digunakan sebagai fungsi objektif untuk mengevaluasi kinerja dari algoritma yang diusulkan. Nilai objektif (fitness) dari segmentas citra menggunakan multilevel thresholding yang dihasilkan berdasarkan pada fungsi otsu diperlihatkan dalam Tabel 2.



Gambar 1. Flowchart robot inersia PSO (a) Segmentasi citra warna (b)



Gambar 2. Citra Tes Warna dan Histogramnya 8068.jpg (a), 16004.jpg (b), 16068.jpg (c), 22090.jpg (d), 28083.jpg (e), 29030.jpg (f), 36046.jpg (g), 41096.jpg (h).

Fungsi fitness digunakan untuk menentukan apakah jumlah threshold yang diberikan telah mencapai nilai optimal atau tidak. Nilai fitness yang dihasilkan dari segmentasi citra mengindikasikan bahwa jika nilai fitness lebih tinggi maka menghasilkan segmentasi yang lebih baik. Hasil segmentasi citra yang dihasilkan dapat dilihat secara visual. Kualitas segmentasi lebih baik atau tidak dalam setiap level *threshold* yang digunakan. Dalam Gambar 2 dapat dilihat bahwa kualitas pada level $n=3,4,5$ maka kualitas segmentasi terlihat lebih baik pada nilai $n=5$ yang dipilih. Deviasi standar dan waktu perhitungan yang didapatkan dari fungsi Otsu didasarkan pada LogPSO yang digunakan dalam penelitian ini ditampilkan dalam Tabel 2. Nilai standar deviasi yang tinggi mengindikasikan bahwa hasil

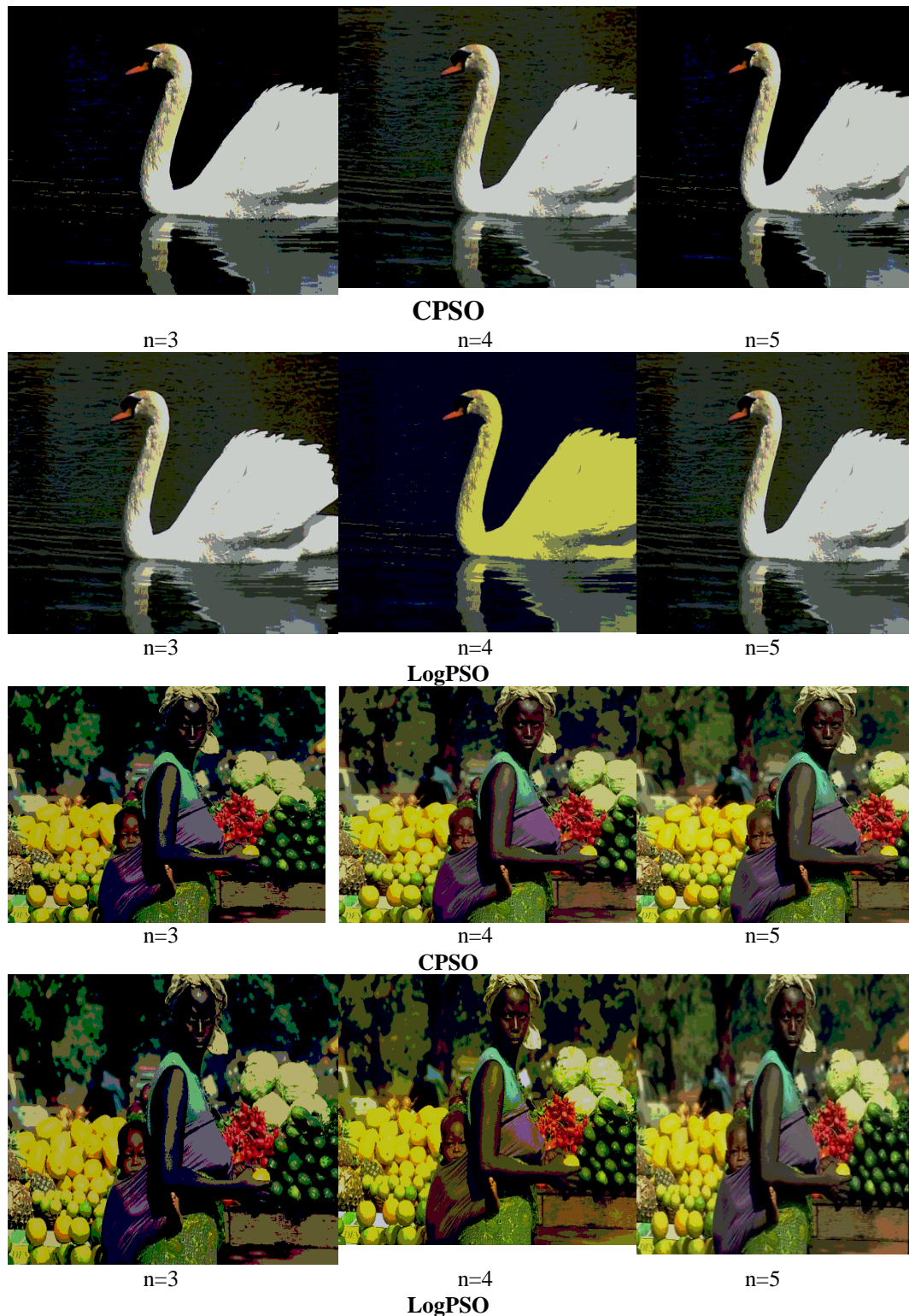
segmentasi yang diperoleh tidak stabil. Kebalikannya jika nilai standar deviasi yang didapatkan lebih kecil maka hasil segmentasi lebih stabil. Dari Tabel 2 terlihat bahwa dengan menggunakan LogPSO maka segmentasi citra yang dihasilkan lebih stabil jika dibandingkan dengan teknik CPSO. Sedangkan dari sisi waktu perhitungan LogPSO lebih tinggi jika dibandingkan CPSO.

Tabel 2. Nilai perbandingan PSNR, MSE, SSIM dan Waktu Penghitungan Menggunakan CPSO dan LogPSO pada Segmentasi Citra Warna.

Citra	n	PSNR(db)		SSIM		MSE		Waktu (detik)	
		CPSO	LogPSO	CPSO	LogPSO	CPSO	LogPSO	CPSO	LogPSO
8068.jpg	3	19,4589	19,4589	0,2112	0,2441	741,6033	741,6033	0,4937	0,5138
	4	23,6042	18,8816	0,3651	0,3724	950,4325	954,3060	0,7405	0,7333
	5	19,4753	23,6167	0,4569	0,5088	284,5526	284,1036	0,9612	0,9528
16004.jpg	3	19,4753	19,4753	0,3557	0,3557	839,9915	839,9915	0,2859	0,4976
	4	20,8479	20,6484	0,4301	0,4745	540,0803	561,2888	0,8102	0,5974
	5	24,2338	24,2272	0,6217	0,6223	269,7052	270,1932	0,7252	0,7483
16068.jpg	3	19,1743	20,1658	0,3378	0,4180	488,8016	655,2868	0,4603	0,3216
	4	21,4648	20,0283	0,4876	0,3814	480,4874	653,8145	0,6849	0,4419
	5	21,6404	20,0283	0,4717	0,6400	810,4654	278,1871	0,7734	0,4349
22090.jpg	3	19,2000	19,2000	0,6454	0,6454	380,0291	810,4654	0,4793	0,3126
	4	22,6068	22,6068	0,7250	0,7250	267,1331	380,0291	0,6833	0,3344
	5	24,0002	24,0698	0,7632	0,7641	1022,7268	262,0071	0,7657	0,4226
28083.jpg	3	18,1644	18,1644	0,3615	0,3615	1022,7268	1022,7268	0,6377	0,3026
	4	22,3718	22,3623	0,5365	0,5375	397,3509	397,9359	0,7590	0,3345
	5	24,6684	24,6636	0,6451	0,6449	223,4990	223,7916	0,6640	0,3971
29030.jpg	3	20,2193	23,0916	0,2836	0,4189	754,3148	370,4695	0,5495	0,2727
	4	23,1819	25,5954	0,5168	0,5135	354,8181	199,5862	0,6365	0,3405
	5	27,4231	27,4332	0,5857	0,5849	130,5468	130,3808	0,8906	0,3962
36046.jpg	3	20,6730	20,6730	0,4569	0,4569	558,4380	558,4380	0,5914	0,2689
	4	22,3403	22,4992	0,5629	0,5086	382,3179	366,1130	0,6913	0,3311
	5	25,0125	25,1054	0,6289	0,6326	208,6333	204,1506	0,6781	0,4069
41096.jpg	3	23,2131	23,2131	0,6400	0,6400	362,4404	362,4404	0,3060	0,2761
	4	25,6123	22,5180	0,7116	0,5768	207,4024	401,2930	0,3551	0,3583
	5	26,8797	25,3356	0,7453	0,6654	150,9903	192,8202	0,4369	0,3935

Tabel 3. Nilai perbandingan Standar Deviasi dan Mean Fitness Menggunakan CPSO dan LogPSO pada Segmentasi Citra Warna

Citra	n	CPSO			LogPSO	
		Band	Standar Deviasi	Mean	Standar Deviasi	Mean
8068.jpg	3	R	7,0711E+01	1,3350E+02	7,0711E+01	1,3350E+02
		G	1,2700E+02	7,0004E+01	1,2700E+02	7,0004E+01
		B	6,8589E+01	1,2650E+02	6,8589E+01	1,2650E+02
	4	R	6,6124E+01	1,3833E+02	6,6124E+01	1,3633E+02
		G	1,3067E+02	6,7022E+01	1,3067E+02	3,5921E+01
		B	6,4065E+01	1,3100E+02	6,3066E+01	3,6333E+01
	5	R	7,0656E+01	1,1850E+02	4,9210E+01	1,1750E+02
		G	1,1075E+02	7,1931E+01	1,1075E+02	7,2450E+01
		B	6,9745E+01	1,1300E+02	6,9304E+01	1,1250E+02
16004.jpg	3	R	6,1518E+01	1,3300E+02	6,1518E+01	1,3300E+02
		G	1,3950E+02	4,3841E+01	1,3950E+02	4,3841E+01
		B	6,0811E+01	7,8000E+01	6,0811E+01	7,8000E+01
	4	R	6,2453E+01	1,3167E+02	6,2453E+01	1,3067E+02
		G	1,2767E+02	6,7486E+01	1,2767E+02	9,6837E+01
		B	6,0863E+01	1,2967E+02	6,0863E+01	1,2333E+02
	5	R	6,2876E+01	1,2675E+02	6,2876E+01	1,2625E+02
		G	1,2900E+02	5,4573E+01	1,2900E+02	5,4653E+01
		B	6,3945E+01	8,4750E+01	6,3683E+01	8,4250E+01
16068.jpg	3	R	5,2326E+01	1,1300E+02	5,2326E+01	1,1300E+02
		G	1,1600E+02	7,0004E+01	1,1600E+02	3,5355E+01
		B	4,6669E+01	1,3950E+02	4,6669E+01	7,6000E+01
	4	R	5,0935E+01	1,2667E+02	5,0935E+01	1,2567E+02
		G	1,1967E+02	4,2143E+01	1,1967E+02	5,2205E+01
		B	5,2157E+01	8,8000E+01	5,1160E+01	1,5933E+02
	5	R	5,6612E+01	1,1950E+02	5,5675E+01	1,2050E+02
		G	1,2125E+02	5,5931E+01	1,2050E+02	4,1845E+01
		B	5,2969E+01	1,6675E+02	5,2969E+01	7,9500E+01
22090.jpg	3	R	6,7175E+01	1,6000E+02	6,7175E+01	1,6000E+02
		G	1,1250E+02	4,2426E+01	1,1250E+02	4,2426E+01
		B	5,7983E+01	1,4800E+02	5,7983E+01	1,4800E+02
	4	R	5,6959E+01	1,5200E+02	5,6959E+01	1,5200E+02
		G	1,0333E+02	4,8501E+01	1,0333E+02	4,8501E+01
		B	5,3028E+01	1,3167E+02	5,3028E+01	1,3167E+02
	5	R	6,4859E+01	1,3800E+02	6,4356E+01	1,3750E+02
		G	1,2100E+02	5,6622E+01	1,2050E+02	5,8657E+01
		B	5,6184E+01	1,5200E+02	5,6152E+01	1,5300E+02



Gambar 3. Hasil Segmentasi Citra Menggunakan CPSO dan LogPSO Pada Level 3,4,5 untuk citra warna 8068.jpg (a), 16004.jpg(b), 16068.jpg (c), 22090.jpg (d), 28083.jpg (e), 29030.jpg (f), 36046.jpg (g), 41096.jpg (h).

KESIMPULAN

Dalam penelitian ini telah usulkan teknik segmentasi untuk *multilevel thresholding* menggunakan bobot inersia *logarithm decreasing particle swarm optimization (LogPSO)*. Efisiensi dan efektivitas dari teknik yang diusulkan diujikan pada delapan citra standar. Kinerja dari metode LogPSO yang diusulkan kemudian dibandingkan dengan algoritma CPSO. Dari percobaan yang dilakukan didapatkan bahwa LogPSO mempunyai kinerja yang lebih baik jika dibandingkan lainnya dalam hal nilai fitness, konvergensi dan *robustness*. Dari sini dapat disimpulkan bahwa LogPSO merupakan suatu teknik yang baik dalam mencari nilai threshold yang optimal. Penelitian selanjutnya dapat menggunakan bobot inersia yang lain agar hasil segmentasi citra lebih baik.

UCAPAN TERIMA KASIH

Terima kasih kepada Ristekdikti yang telah mendanai penelitian ini melalui skema penelitian dasar unggulan perguruan tinggi (PDUPT) tahun 2022.

DAFTAR PUSTAKA

- Dhieb, M. and Frikha, M. (2016) 'A multilevel thresholding algorithm for image segmentation based on particle swarm optimization', *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 0(1), pp. 0–3.
- Dilpreet, K. and Yadwinder, K. (2014) 'Various Image Segmentation Techniques: A Review', *International Journal of Computer Science and Mobile Computing*, 3(5), pp. 809–814.
- Dorigo, M., Stutzle, T. and Birattari, M. (2006) 'Ant Colony Optimization', *IEEE Computational Intelligence Magazine*, (November 2006), pp. 28–39.
- Duraisamy, P. S. and Kayalvizhi, R. (2010) 'A New Multilevel Thresholding Method Using Swarm Intelligence Algorithm for Image Segmentation', *Journal of Intelligent Learning Systems and Applications*, 02(03), pp. 126–138.
- Eberhart, R. and Kennedy, J. (1995) 'New optimizer using particle swarm theory', *Proceedings of the International Symposium on Micro Machine and Human Science*, pp. 39–43.
- Gao, Y. L., An, X. H. and Liu, J. M. (2008) 'A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation', *Proceedings - 2008 International Conference on Computational Intelligence and Security, CIS 2008*, 1, pp. 61–65.
- Guimin, C. *et al.* (2006) 'Self-active inertia weight strategy in particle swarm optimization algorithm', *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 1(2002), pp. 3686–3689.
- Guruprasad, P. (2020) 'Overview Of Different Thresholding Methods In Image Processing', *TEQIP Sponsored 3rd National Conference on ETACC*, (June).
- Li, H. R. and Gao, Y. L. (2009) 'Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation', *2009 2nd International Conference on Information and Computing Science, ICIC 2009*, 1(3), pp. 66–69.
- Malik, R. F. *et al.* (2007) 'New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight', *International Journal of Computer Science and Security. IJCSS*, 1(1), pp.

35–44.

- Murinto and Ismi, D. P. (2019) ‘Hyperspectral image segmentation using discriminant independent component analysis and swarm optimization approaches’, in *IOP Conference Series: Materials Science and Engineering*.
- Otsu, N. (1979) ‘A Threshold Selection Method from Gray-Level Histograms’, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, SMC-9(1), pp. 62–66.
- Pare, S. *et al.* (2017) ‘An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix’, *Expert Systems with Applications*, 87, pp. 335–362.
- Sahoo, P. K., Soltani, S. and Wong, A. K. C. (1988) ‘A survey of thresholding techniques’, *Computer Vision, Graphics and Image Processing*, 41(2), pp. 233–260. d
- Sathya, P. D. and Kayalvizhi, R. (2010) ‘Development of a New Optimal Multilevel Thresholding Using Improved Particle Swarm Optimization Algorithm for’, *International Journal of Electronics*, 2(1), pp. 63–67.
- Shi, Y. and Eberhart, R. (1998) ‘Modified particle swarm optimizer’, *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, (February), pp. 69–73.
- Storn, R. and Price, K. (1995) ‘Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces’, *Technical Report*, (TR-95-012), pp. 1–12.
- Xin, J., Chen, G. and Hai, Y. (2009) ‘A particle swarm optimizer with multi-stage linearly-decreasing inertia weight’, *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization, CSO 2009*, 1, pp. 505–508.