

# Harmony Search berbasis Chaos dan Random Adjustment untuk Perbaikan Kontras Citra

L. M. Rasdi Rere<sup>1</sup> dan Bheta Agus Wardijono<sup>1</sup>

Teknik Komputer<sup>1</sup>, Sistem Komputer<sup>2</sup>

STMIK Jakarta STI&K, Jl. BRI No. 17, Radio Dalam Jakarta 12140

rasdi@jak-stik.ac.id

## Abstrak

*Harmony search* (HS) adalah salah satu metode optimasi metaheuristik yang terbukti handal dalam beberapa tahun terakhir. Performa dari metode ini, seperti umumnya algoritme metaheuristik lainnya, sangat bergantung dari pengaturan nilai parameternya, yang biasanya konstan selama proses perhitungan. Akan tetapi untuk menentukan nilai parameter suatu algoritme metaheuristik tidaklah mudah, karena sangat bergantung dari karakteristik masalah yang dihadapi. Untuk mengatasi hal ini, sejumlah cara telah diusulkan untuk dapat mengatur parameternya secara otomatis, seperti menggunakan logika *fuzzy*, metode *chaos* ataupun teknik *random adjustment*. Selama beberapa tahun terakhir, metode-metode yang diusulkan ini telah dikembangkan secara independen, dan integrasi dari dua atau lebih metode tersebut belum banyak dilakukan. Karena itu dalam penelitian ini, sebuah metode baru yang menggabungkan *chaos* dan teknik *random adjustment* pada algoritme HS diusulkan. Sebagai studi kasus penelitian, metode yang diusulkan diaplikasikan untuk perbaikan kontras pada citra Lena, Rice dan Cameraman. Hasil eksperimen yang diperoleh menunjukkan bahwa metode yang diusulkan lebih baik dari HS asli, HS dengan *chaos*, maupun HS berbasis teknik *random adjustment*.

**Kata Kunci** : metaheuristik, harmony search, *chaos*, teknik *random adjustment*.

## 1. Pendahuluan

Algoritme metaheuristik sebagian besar diinspirasi oleh fenomena alam, yang umumnya didasarkan pada prinsip dalam bidang fisika, biologi dan etologi. Algoritme metaheuristik berdasarkan fenomena fisika misalnya, dapat ditemukan pada *simulated annealing*, *macrocanonical annealing* dan *threshold accepting method*. Berdasarkan fenomena biologi yang didasarkan pada proses evolusi adalah *differential evolution*, *evolution strategy*, dan *genetic algorithm*. Sedangkan berdasarkan pada etologi, yang merupakan bentuk perilaku pada hewan adalah *ant colony optimization*, *particle swarm optimization*, dan *firefly algorithm* [1]. Bentuk lain dari algoritme metaheuristik adalah berdasarkan fenomena musik, seperti *harmony search* [2].

Meskipun algoritme metaheuristik dapat menyelesaikan beragam masalah dengan cepat, fleksibel dan sederhana dalam desain, serta mudah untuk diimplementasikan [3]. Akan tetapi

untuk mengatur nilai parameter yang tepat pada metode ini tidak mudah, karena sangat tergantung pada karakteristik masalah yang dihadapi. Beberapa metode telah direkomendasikan untuk dapat mengatur parameter metaheuristik secara otomatis.

Liu dan Lampinen [4] mengusulkan FADE (*fuzzy adaptive differential evolution*), dimana logika *fuzzy* dipergunakan untuk dapat mengatur parameter mutasi dan *crossover* dalam algoritme DE. Coelho dkk [5] menggunakan *chaos* pada DE dalam perbaikan kontras citra. Norman dkk [6] mengusulkan aDE (*adaptive DE*) berbasis teknik *random adjustment*, dengan strategi membandingkan nilai objektif anak dengan rata-rata nilai objektif generasi saat ini, sedangkan Li dkk [7] menggunakan *chaos* pada *particle swarm optimization* untuk penjadwalan dalam *cloud workflow system*.

Pada dasarnya semua metode yang telah diusulkan memiliki kelebihan dalam mengatur parameter metaheuristik secara otomatis, akan tetapi integrasi dari dua atau lebih metode

tersebut belum dilakukan. Karena itu pada penelitian ini diusulkan metode baru yang memberikan keuntungan dari kombinasi *chaos* dan *random adjustment* pada algoritme *harmony search*. Metode *chaos* dapat dipergunakan untuk menghindari minimum lokal, sedangkan *random adjustment* untuk mendapatkan nilai akurasi yang lebih baik.

Dalam beberapa tahun terakhir, algoritme metaheuristik telah banyak dipergunakan untuk perbaikan kontras pada citra. Sejumlah literatur melaporkan [8][9][10], bahwa algoritme ini lebih baik dari metode klasik yang telah ada seperti ekualisasi histogram maupun manipulasi kontras. Karena itu sebagai studi kasus penelitian, metode yang diusulkan diterapkan untuk perbaikan kontras pada citra.

## 2. Perbaikan Kontras Citra

Salah satu isu penting dalam *computer vision*, pemrosesan citra (*image processing*), dan pengenalan pola (*pattern recognition*) adalah perbaikan citra. Secara umum metode perbaikan citra dapat dibedakan dalam *contrast enhancement*, *edge enhancement*, *noise enhancement* dan *edge restoration* [11]. Penelitian ini difokuskan pada perbaikan kontras pada citra (*contrast enhancement*).

Teknik yang dipergunakan dalam perbaikan kontras pada citra, dapat dibedakan menjadi dua kategori, yaitu metode langsung dan metode tidak langsung. Algoritme berdasarkan metode perbaikan kontras pada citra secara tidak langsung, akan memperbaiki citra tanpa adanya pengukuran kontras. Sebaliknya metode langsung, membangun kriteria ukuran kontras dan memperbaiki citra dengan pengukuran kontras secara langsung [5].

Perbaikan citra dengan metode langsung, menggunakan fungsi transformasi dalam domain spasial ke citra level-keabuan (*gray-level*). Untuk menghasilkan citra yang disempurnakan, fungsi transformasi menghasilkan nilai intensitas baru untuk setiap piksel dari citra asli menurut persamaan (1) berikut ini:

$$h(i, j) = T[f(i, j)] \quad (1)$$

Dimana  $f(i, j)$  adalah nilai keabuan piksel ke- $(i, j)$  pada citra masukan,  $h(i, j)$  adalah nilai keabuan piksel  $(i, j)$  pada citra perbaikan, serta  $T$  adalah fungsi transformasi [10].

### 2.1. Fungsi Transformasi Citra

Pengukuran kontras pada citra dapat dilakukan secara global maupun lokal. Jika citra berisi informasi tekstur, menggunakan kontras lokal lebih sesuai. Sebuah fungsi kontras lokal berkaitan dengan perbedaan relatif antara wilayah tengah dan daerah sekitarnya yang lebih besar dari piksel yang diberikan. Nilai kontras diperbaiki dengan fungsi perbaikan kontras, seperti trigonometri, logaritma, eksponensial, dan akar kuadrat [4].

Metode perbaikan lokal menerapkan fungsi transformasi berdasarkan pada distribusi level-keabuan di wilayah setiap piksel dalam citra yang diberikan. Dalam penelitian ini, fungsi transformasi yang dipergunakan untuk setiap piksel yang berada di lokasi  $(x, y)$  diberikan menurut persamaan (2) berikut ini:

$$T[f(i, j)] = \left(s \frac{M}{\sigma + q}\right) (f - rm) + m^p \quad (2)$$

Dimana  $\sigma = \sigma(x, y)$  dan  $m = m(x, y)$  masing-masing adalah standar deviasi dan rata-rata level keabuan, yang dihitung di lingkungan sekitar pusat pada titik  $(x, y)$ . Sedangkan  $M$  adalah global rata-rata dari citra dan  $f = f(x, y)$  adalah nilai intensitas level-keabuan [5].

Sebuah nilai untuk  $q$  dalam persamaan ini, memungkinkan standar deviasi bernilai nol. Sementara  $r$  memungkinkan hanya sebagian kecil dari nilai rata-rata  $m(x, y)$  untuk berkurang dari piksel asli level-keabuan  $f(x, y)$ . Bentuk yang terakhir adalah  $m^p = m(x, y)^p$ , yaitu efek pencerahan dan penghalusan pada citra. Semua parameter  $p$ ,  $q$ ,  $r$ , dan  $s$  didefinisikan bernilai positif, dengan nilai yang sama untuk seluruh citra [9].

### 2.2. Kriteria Evaluasi Citra

Untuk menerapkan secara otomatis teknik perbaikan citra, tanpa adanya campur tangan manusia dan parameter objektif yang diberikan oleh pengguna. Sebuah standar untuk metode perbaikan harus ditentukan. Suatu fungsi objektif, seperti ditunjukkan pada persamaan (3) berikut ini, dipergunakan untuk standar perbaikan citra yang dipergunakan dalam penelitian ini.

$$F(M) = \log(\log E) \frac{nc}{PH.PV} H \quad (3)$$

Dalam hal ini,  $F(M)$  adalah fungsi objektif dengan transformasi  $T$  pada setiap piksel di lokasi  $(x, y)$  yang diaplikasikan sesuai dengan persamaan ini. Sedangkan parameter  $p, q, r$  dan  $s$  diberikan oleh suatu fungsi  $M(p, q, r, s)$ . Selain itu,  $E = E(I(M))$  adalah intensitas tepi yang dideteksi oleh detektor tepi *Sobel*, yang diaplikasikan untuk perubahan citra  $I(M)$ , dan  $ne = ne(I(M))$  adalah jumlah piksel pada bagian tepi, yang juga dideteksi dengan detektor tepi *Sobel*. Sementara  $PH$  dan  $PV$  adalah jumlah piksel dalam arah horisontal dan vertikal dari citra. Serta yang terakhir adalah entropi citra yang diukur dengan  $H(I(M))$  [5].

### 3. Algoritme Harmony Search

*Harmony search* diusulkan oleh Geem dkk pada tahun 2001 [2]. Algoritme ini diinspirasi oleh hasil pengamatan bahwa tujuan dari musik adalah mencari keadaan harmoni yang sempurna. Suatu *harmony* dalam musik dapat dianalogikan mencari nilai optimal dalam sebuah proses optimisasi. Proses pencarian dalam optimisasi dapat dibandingkan dengan seorang musisi yang melakukan proses improvisasi musik.

Ketika seorang musisi berimprovisasi, ada tiga hal yang mungkin dilakukan: Pertama adalah memainkan musik yang dikenal sesuai dengan yang ada dalam ingatannya, kedua adalah memainkan musik yang dikenal akan tetapi dengan beberapa penyesuaian, serta ketiga adalah memainkan musik yang baru. Dalam algoritme *HS*, tiga aturan ini berkorelasi dengan dua parameter yang bersesuaian yaitu *harmony memory consideration rate (HMCR)* dan *pitch adjusting rate (PAR)*. Secara umum prosedur algoritme *HS* untuk masalah optimisasi adalah sebagai berikut [12]:

1. **Inisialisasi masalah dan parameter:** Dalam algoritme ini, masalah dapat berupa optimisasi minimum atau maksimum, sedangkan parameter yang berkaitan adalah *HMCR*, *PAR*, ukuran *harmony memory (HM)* dan kriteria terminasi.
2. **Inisialisasi harmony memory:** *HM* biasanya dinyatakan sebagai sebuah matrik yang nilainya didapatkan secara acak, sebagai sebuah vektor solusi dan susunannya berdasarkan fungsi objektif.

3. **Membangkitkan harmony baru:** Sebuah vektor dari *harmony* yang baru dihasilkan dari *HM* berdasarkan *HMCR*, *PAR* dan randomisasi. Pemilihan nilai dari vektor harmoni yang baru ini berdasarkan parameter *HMCR* pada kisaran 0 dan 1. Hasilnya kemudian ditentukan apakah diperlukan penyesuaian nada berdasarkan parameter *PAR*.
4. **Memperbaharui HM:** *harmony* yang baru menggantikan *harmony* terburuk berdasarkan nilai fungsi objektifnya, yaitu jika nilai objektif *harmony* baru lebih baik dari *harmony* terburuk.
5. **Kriteria terminasi:** Dalam kasus kriteria terminasi telah memenuhi, maka proses komputasi dihentikan. Jika tidak, maka proses (3) dan (4) akan terus berlangsung. Pada akhirnya, vektor terbaik yang ada dalam *HM* dipilih sebagai solusi terbaik dari masalah yang diberikan.

#### 3.1. HS berbasis Chaos

*Chaos* merupakan karakteristik dari sistem tidak linier, meskipun tampaknya stokastik, tetapi yang sebenarnya adalah sistem linier non-deterministik dalam kondisi deterministik [13]. Dalam beberapa tahun terakhir, sejumlah literatur menginformasikan bahwa sistem *chaos* adalah alternatif yang baik dalam prosedur optimisasi stokastik.

Karena memiliki sifat "*ergodicity*", suatu sistem *chaos* bisa digunakan untuk memperbaiki perilaku pencarian dan menghindari terjebak dalam optimum lokal dalam masalah optimisasi [5]. Beberapa contoh sistem *chaos* adalah *lozi map*, *ikeda map* dan *logistic map (chaotic sequence)* [14]. Pada penelitian ini yang akan digunakan adalah *chaotic sequence*, yang diberikan menurut persamaan (4) berikut ini:

$$x(m) = \mu \cdot x(m-1) \cdot (1 - x(m-1)) \quad (4)$$

Di mana  $m$  dan  $\mu$  adalah parameter sampel dan parameter kontrol. Secara substansi, kedua parameter ini akan menentukan apakah  $x$  stabil pada ukuran yang konstan, berosilasi dalam pola yang tidak dapat diprediksi, atau berosilasi diantara ukuran sekuensial yang terbatas. Sebuah perbedaan yang sangat kecil pada nilai awal  $x$ , akan dapat menyebabkan perbedaan

an besar pada perilakunya dalam waktu yang lama [5].

Dalam penelitian ini, variasi dari  $\mu$  adalah  $1 \leq \mu \leq 4$ , dimana  $x$  didistribusikan dalam rentang  $[0, 1]$ , dengan memberikan nilai awal  $x(1) \neq \{0, 0, 25, 0, 50, 0, 75, 1\}$ . Pada algoritme *HS* berbasis *chaos*, parameter *HMCR* dan *PAR* diinisialisasi menggunakan *chaotic sequence*. Nilai dari  $HMCR = y$  dan  $PAR = z$  diperbaharui menurut persamaan (5) dan (6) berikut ini:

$$y_N = \mu \cdot y_{N-1} \cdot (1 - y_{N-1}) \quad (5)$$

$$z_N = \mu \cdot z_{N-1} \cdot (1 - z_{N-1}) \quad (6)$$

Dalam hal ini, inisialisasi dari *HMCR* dan *PAR* dilakukan secara *random* dan didasarkan pada nilai batas yang telah ditentukan.

### 3.2. HS berbasis RA

Teknik *random adjustment (RA)* secara umum dipergunakan untuk membuat parameter *HS* dapat beradaptasi dengan baik pada masalah yang dihadapi. Pada metode ini, tujuan utamanya adalah mencari parameter optimal, sehingga fungsi objektif terbaik akan di dapatkan.

Strategi *HS* berbasis *RA* adalah membandingkan nilai fungsi objektif dari *harmony memory (HM)* terbaru ( $f_{Baru}$ ), dengan rata-rata nilai fungsi objektif pada *HM* saat ini ( $f_{Rata}$ ). Jika  $f_{Baru}$  lebih baik dari  $f_{Rata}$ , maka parameter *HMCR* dan *PAR* terbaru yang dipilih secara *random* akan dipertahankan. Jika tidak, maka *HMCR* dan *PAR* yang lama akan dipertahankan.

### 3.3. HS berbasis Chaos dan RA

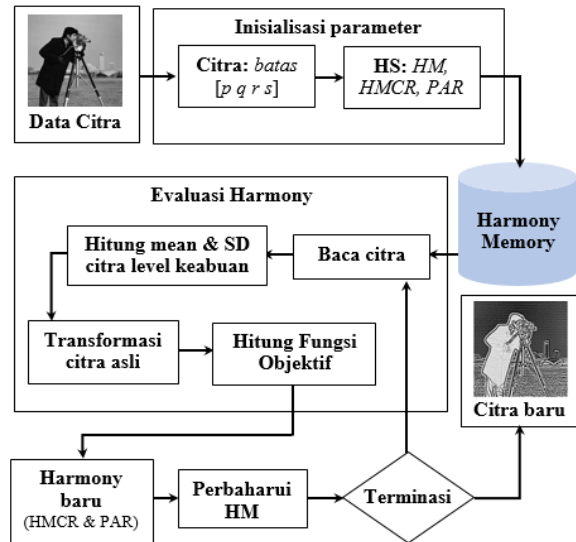
Algoritme *HS* berbasis *chaos* dan *random adjustment* adalah metode yang diusulkan dalam penelitian ini. Strategi yang dipergunakan adalah mengganti parameter *HMCR* dan *PAR* dengan nilai yang dibangkitkan dari *chaotic sequence*.

Pada prinsipnya dalam metode yang diusulkan ini, parameter *HMCR* dan *PAR* akan tetap dipergunakan dalam proses, selama fungsi objektif baru lebih baik dari rata-rata fungsi objektif lama dalam *harmony memory (HM)*. Jika tidak, maka parameter *HMCR* dan *PAR* akan digantikan nilainya yang berasal dari *chaotic sequence*.

## 4. Skenario Eksperimen

Secara umum skenario desain dari metode yang diusulkan, untuk perbaikan citra, diperlihatkan pada Gambar 1. Pada awalnya diberikan sebuah citra, kemudian dilakukan inisialisasi parameter pada citra dan algoritme *HS*.

Suatu *HM* awal ditentukan secara acak, berdasarkan batasan nilai yang ditentukan. Setelah *HM* awal terbentuk, evaluasi setiap *harmony* dilakukan melalui tahapan membaca citra, menghitung *mean* dan standar deviasi (*SD*) citra level keabuan, transformasi pada citra, serta kemudian menghitung nilai fungsi objektif. Tahap berikutnya adalah menghasilkan *harmony* baru, sampai dengan memperbaharui *HM*.



Gambar 1: Skenario desain harmony search berbasis chaos dan random adjustment

Evaluasi pada *HM* baru dilakukan kembali, demikian juga dengan proses lainnya, sampai kriteria terminasi terpenuhi. Hasil akhir dari yang didapatkan adalah citra baru, dengan parameter ( $p, q, r, s$ ), nilai fungsi objektif akhir dan waktu komputasi.

Pada penelitian ini, ada empat skenario yang dilakukan seperti diperlihatkan pada Tabel 1. Skenario pertama adalah tiga eksperimen pada *HS* asli, yaitu  $HS_1, HS_2$ , dan  $HS_3$ . Skenario ke dua adalah eksperimen *HS* berbasis *chaos (HSC)*. Skenario ke tiga adalah eksperimen *HS* berbasis *random adjustment (HSRA)*, serta skenario ke empat adalah eksperimen dari metode yang diusulkan, yaitu *HS* berbasis kombinasi *chaos* dan *Random adjustment (HSCRA)*.

Tabel 1: Parameter pada skenario HS, HSC, HSRA, dan HSCRA

Metode	HMCR	PAR
$HS_1$	0,2	0,2
$HS_2$	0,5	0,5
$HS_3$	0,8	0,8
$HSC$	Chaos	Chaos
$HSRA$	RA	RA
$HSCRA$	Chaos&RA	Chaos&RA

Dalam rangka untuk mengevaluasi hasil perbaikan yang dilakukan, tiga citra digunakan yaitu citra Lena, Boat dan Cameraman. Eksperimen dilakukan menggunakan Matlab, dengan 30 kali dijalankan untuk setiap eksperimen yang dilakukan. Pada prinsipnya ke empat skenario eksperimen mencari nilai dari ke empat parameter yang ada pada  $Z$ , yaitu  $p, q, r, s$ , sehingga nilai fungsi objektif pada persamaan (3) menjadi optimal.

Karena tidak mudah untuk memastikan parameter pada algoritme  $HS$  ini, untuk semua eksperimen yang dilakukan, banyaknya *harmony memory* ditentukan 20, dengan kriteria pemberhentian 40 untuk semua algoritme yang dipergunakan. Parameter yang akan dicari, yaitu  $Z = (pqrs)$  juga ditentukan nilai batasnya:  $p = (0 - 1, 5)$ ,  $q = (0 - 2)$ ,  $r = (0, 5 - 5)$ , dan  $s = (0, 5 - 30)$ .

Pada awal program, semua citra dikonversi ke presisi ganda untuk perhitungan numerik, serta mengubah ukurannya menjadi 256x256 piksel. Dalam hal perbaikan kontras untuk citra berwarna, hal yang dilakukan adalah mengkonversi citra dari ruang warna  $RGB$  (*red, green, blue*) ke ruang warna  $YIQ$  (*luminance, hue, saturation*). Kemudian menggunakan perhitungan dengan metaheuristik hanya untuk komponen saturasi, dan setelah selesai mengkonversi kembali ke bentuk  $RGB$ .

## 5. Hasil Eksperimen

Dalam penelitian ini, perhatian utama dari eksperimen yang dilakukan adalah memperbaiki kontras citra dengan menggunakan algoritme *harmony Search* berbasis *chaos* dan *random adjustment*. Tujuannya adalah memaksimalkan jumlah piksel pada bagian tepi citra, serta meningkatkan intensitas bagian tepi dan pengukuran entropi.

Eksperimen pada citra Lena, Rice dan Ca-

meraman untuk fungsi objektif diperlihatkan seperti pada Tabel 2. Dengan nilai fungsi objektif citra asli adalah 0,0812 untuk citra Lena, 0,2019 untuk citra Rice, dan 0,1013 untuk citra Cameraman. Hasil ini memperlihatkan bahwa untuk semua citra, nilai fungsi objektif terbaik adalah pada metode yang diusulkan, yaitu 0,1460 untuk citra Lena, 0,2424 untuk citra Rice, dan 0,1416 untuk citra Cameraman.

Tabel 2: Rata-rata fungsi objektif

Metode	Lena	Rice	Cameraman
$HS_1$	0,1383	0,2418	0,1398
$HS_2$	0,1428	0,2359	0,1391
$HS_3$	0,1409	0,2354	0,1397
$HSC$	0,1440	0,2419	0,1403
$HSRA$	0,1441	0,2407	0,1405
$HSCRA$	0,1460	0,2424	0,1416

Dalam hal waktu komputasi, seperti diperlihatkan Tabel 3, hasil terbaik citra Lena adalah 7,7 detik untuk metode  $HS_1$  dan  $HS_2$ . Pada citra Rice adalah 6,7 detik untuk metode  $HS_3$ , serta pada citra Cameraman adalah 6,9 detik untuk metode yang diusulkan.

Tabel 3: Rata-rata waktu komputasi (detik)

Metode	Lena	Rice	Cameraman
$HS_1$	7,7	14,5	9,0
$HS_2$	7,7	8,7	8,9
$HS_3$	7,8	6,7	8,9
$HSC$	9,8	6,8	8,9
$HSRA$	10,0	8,9	8,9
$HSCRA$	7,8	6,9	6,9

Perbandingan untuk citra asli, citra dengan fungsi objektif terbaik, dan fungsi objektif terendah untuk semua metode yang dipergunakan dalam penelitian ini diperlihatkan seperti pada Gambar 2. Secara umum berdasarkan hasil eksperimen yang dilakukan,  $HS$  berbasis kombinasi *chaos* dan *random adjustment* yang diusulkan, lebih baik dari algoritme  $HS$  asli,  $HS$  dengan *chaos*, maupun  $HS$  menggunakan teknik *random adjustment*. Hal ini karena metode yang diusulkan, dapat mengambil manfaat dari kelebihan *chaos* dan teknik *random adjustment*, sehingga dapat terhindar dari jebakan lokal optima, serta mendapatkan akurasi yang lebih baik, karena mendapatkan parameter yang tepat.



Gambar 2: Perbandingan citra asli (kiri), citra fungsi objektif terbaik (tengah) dan citra fungsi objektif terendah (kanan)

## 6. Penutup

Algoritme *harmony search* yang dipergunakan dalam penelitian ini terbukti dapat meningkatkan kontras pada citra. Hal ini dapat dilihat dari nilai fungsi objektif semua metode *HS* yang dipergunakan, lebih besar dari citra asli. Pada citra Lena dengan fungsi objektif 0,0812 misalnya, rata-rata nilai fungsi objektif *HS* asli ( $HS_1$ ) adalah 0,1383, *HS* dengan *chaos* (*HSC*) adalah 0,1440, *HS* berbasis teknik *random adjustment* (*HSRA*) adalah 0,1441, serta metode yang diusulkan (*HSCRA*) adalah 0,1460.

Berdasarkan nilai rata-rata fungsi objektifnya, metode yang diusulkan lebih baik dari *HS* asli, *HS* dengan *chaos*, maupun *HS* berbasis *RA* untuk semua citra yang dipergunakan. Akan tetapi untuk waktu komputasi, *HS* asli secara umum lebih baik dari lainnya.

Penelitian lanjutan diperlukan untuk membuktikan bahwa metode yang diusulkan untuk mengintegrasikan *chaos* dan teknik *random adjustment* sangat menjanjikan. Hal ini dapat dilakukan dengan menambahkan sampel dataset citra lainnya seperti citra Boat, Elaine, House dan lainnya, ataupun strategi yang diusulkan diaplikasikan pada algoritme Metaheuristiknya lainnya seperti *simulated annealing*, *particle swarm optimization*, *differential evolution* dan *ant colony optimization*.

## Daftar Pustaka

- [1] Ilhem Boussaid, Julien Lepagnot, dan Patric Siarry, "A survey on optimization metaheuristics Information Science 237: 82–117, 2013.
- [2] Z.W. Geem, J. H. Kim, and G. V. Logathan, "A new heuristic optimization algorithm: harmony search" Simulation 76 (2): 60 – 68, 2001.
- [3] El-Ghazali Talbi, *Metaheuristics From Design to Implementation*, John Wiley & Sons, Hoboken, New Jersey, 2009.
- [4] J. Liu, dan J. Lampinen, "A fuzzy adaptive differential evolution algorithm" Soft Computing - A fusion of Foundations, Methodologies and Applications 9 (6): 448–462, 2013.
- [5] Coelho L. D.S, G. Sauer, dan M. Rudek, "Differential evolution optimization combined with chaotic sequences for image contrast enhancement" Chaos, Solitons and Fractals 42: 522–529, 2009.
- [6] N. Noman, D. Bollgala, dan H. Iba, "An adaptive differential evolution algorithm" IEEE Evolutionary Computation, 2229–2236, 2011.
- [7] X. Li, J. Xu, dan Y. Yang, " A chaotic particle swarm optimization - based heuristic for market-oriented task-level scheduling in cloud work ow systems" Computational Intelligenece and Neuroscience, 1–11, 2015 .
- [8] C. Munteanu dan A. Rosa, "Towards automatic image enhancement using genetic algorithms" IEEE Congress on Evolutionary Computation, 1535–1542, 2000.
- [9] C. Munteanu dan A. Rosa, "Gray-scale enhancement as an automatic process driven by evolution" IEEE Transaction on systems, man, and cybernetics Part B: Cybernetics 34 (2): 1292–1298, 2004.
- [10] A. Gorai dan A. Ghosh, "Gray-level image enhancement by particle swarm optimization" Congress on Nature Biologically Inspired Computing, 72–77, 2009.
- [11] H.D. Cheng dan H. Xu," A novel fuzzy logic approach to contrast enhancement," Pattern Recognition 33: 809 – 819, 2000.

- [12] Kang Seok Lee dan ZongWoo Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice" *Comput. Methods Appl. Mech. Engrg* 194: 3902–3933, 2005.
- [13] M. F. El-Santawy, A.N. Ahmed, dan R.A.Z. El-Dean, "Chaotic differential evolution optimization" *Computing and Information System Journal* 16(2): 1–4, 2012.
- [14] M. R. Senkerik, Pluhacek, D. Davendra, I. Zelinka, dan Z. Oplatkova, "Chaos driven evolutionary algorithm: a new approach for evolutionary optimization" *International Conference on Systems, Control and Informatics*, 117–122, 2013.
- [15] H.D. Cheng dan H. Xu, "A novel fuzzy logic approach to contrast enhancement," *Pattern Recognition* 33: 809 – 819, 2000.