

FRAGILE IMAGE WATERMARKING BERBASIS DCT DENGAN OPERATOR EVOLUSI HYBRID OF PARTICLE SWARM OPTIMIZATION

Ronsen Purba¹, Arwin Halim², Apin Ridwan³, Rudy⁴

Program Studi Teknik Informatika, STMIK Mikroskil

Jl. Thamrin No. 122, 124, 140 Medan 20212

ronsen@mikroskil.ac.id¹, arwin@mikroskil.ac.id²,
apinrdw@gmail.com³, lawliet100@yahoo.co.id⁴

Abstrak

Fragile image watermarking berbasis *discrete cosine transform* (DCT) merupakan sebuah teknik *watermarking* dimana gambar sampul ditransformasi ke domain frekuensi. Lalu gambar tanda tangan disisipkan dengan mensubstitusi *least significant bit* (LSB) pada koefisien domain frekuensi. Namun, hasil ekstraksi gambar tanda tangan akan sangat berbeda karena terjadi penyimpangan dalam menerjemahkan bilangan riil ke bilangan bulat dengan hasil yang diperoleh dari proses *inverse DCT* (IDCT) yang mengakibatkan kehilangan gambar tanda tangan yang telah disisipkan ke dalam gambar. Untuk mengatasi masalah tersebut, diterapkan *hybrid of particle swarm optimization* (HPSO) dengan penambahan *cauchy mutation* dan *roulette wheel selection*. Pendekatan ini melakukan evaluasi *fitness* untuk setiap partikel hingga diperoleh satu set solusi terbaik yang menjadi aturan konversi bilangan riil ke bilangan bulat. Hasil pengujian yang telah dilakukan menunjukkan peningkatan yang signifikan dalam meningkatkan akurasi dan kualitas gambar yang lebih baik dibandingkan pembulatan sederhana.

Kata kunci: *fragile watermarking, discrete cosine transform, particle swarm optimization, cauchy mutation, roulette wheel selection*

1. Pendahuluan

Kemudahan penyebaran konten digital berupa gambar melalui media internet, mengakibatkan kebutuhan perlindungan hak cipta serta menjamin keasliannya semakin penting. Oleh karena itu banyak pihak menggunakan teknik *watermarking* untuk melindungi kepemilikan terhadap sebuah citra. *Watermarking* adalah cara untuk menyisipkan sebuah informasi atau data pada sebuah gambar sampul [1]. Salah satu teknik *watermarking* yang sering digunakan adalah *Fragile image watermarking*. *Fragile image watermarking* berbasis DCT merupakan teknik perlindungan dimana gambar sampul ditransformasi ke domain frekuensi dan dilakukan proses penyisipan gambar tanda tangan [2]. Namun, gambar tanda tangan yang diekstraksi sangat berbeda disebabkan oleh penyimpangan dalam menerjemahkan bilangan riil ke bilangan bulat yang dihasilkan melalui IDCT yang mengakibatkan kehilangan gambar tanda tangan yang telah ditanamkan.

Ghargory, S. M. melakukan penerapan HPSO untuk memperbaiki penyimpangan dalam menerjemahkan bilangan riil ke bilangan bulat yang dihasilkan dari perhitungan IDCT yaitu dengan menghasilkan sebuah solusi yang akan digunakan untuk membulatkan bilangan riil tersebut. Hal ini dilakukan agar gambar tanda tangan yang diperoleh memiliki tingkat akurasi dan kualitas yang lebih baik [3].

Tujuan penelitian ini adalah menerapkan HPSO serta membandingkan hasil keakuratan gambar tanda tangan yang diekstraksi dengan pembulatan dan HPSO. Dengan demikian diperoleh metode *watermarking* dengan akurasi dan kualitas yang lebih baik. Dalam penelitian diberikan beberapa batasan seperti: tipe gambar JPEG, BMP, GIF dan PNG, warna gambar *grayscale* untuk sampul dan hitam-putih untuk tanda tangan, ukuran sampul minimum 256x256, maksimum 1024x1024 dan kelipatan 8 piksel untuk menghindari *overlapping*, tanda tangan $\frac{1}{4}$ dari gambar sampul serta parameter HPSO yang digunakan: jumlah partikel (1 – 50), jumlah iterasi (1 – 100), bobot inersia (0.1 – 1), bobot kognitif dan sosial (0.1 – 3).

2. Kajian Pustaka

2.1. Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) adalah suatu teknik mengubah sinyal atau gambar dari domain spasial ke domain frekuensi [4]. Untuk mentransformasi ke domain frekuensi dengan DCT (F), citra dibagi menjadi $N \times N$ piksel. Kemudian dilakukan perhitungan dengan Persamaan 1:

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \alpha(u) \cdot \alpha(v) \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \cos\left[\frac{\pi u (2i+1)}{2N}\right] \cos\left[\frac{\pi v (2j+1)}{2M}\right] \cdot f(i, j) \quad \dots\dots\dots(1)$$

Dimana untuk kembali menjadi domain spasial dilakukan IDCT (f) yaitu dengan Persamaan 2:

$$f(i, j) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \alpha(u) \cdot \alpha(v) \cdot \cos\left[\frac{\pi u (2i+1)}{2N}\right] \cos\left[\frac{\pi v (2j+1)}{2M}\right] \cdot F(u, v) \quad \dots\dots\dots(2)$$

$$\text{dimana, } \alpha(\varepsilon) = \begin{cases} \frac{1}{\sqrt{2}} & \text{untuk } \varepsilon = 0 \\ 1 & \text{untuk } \varepsilon \neq 0 \end{cases} \quad \dots\dots\dots(3)$$

2.2. Particle Swarm Optimization (PSO)

PSO merupakan algoritma metaheuristik yang melakukan pencarian solusi yang disebarakan secara acak dan melakukan perpindahan posisi hingga diperoleh solusi yang optimal. Setiap partikel menyampaikan informasi atau posisi terbaik kepada partikel lain untuk menyesuaikan posisi dan kecepatan berdasarkan informasi yang diperoleh [5]. Setiap perpindahan partikel dapat dihitung dengan Persamaan 4 dan 5:

$$V_j(i) = w_i \cdot V_j(i-1) + c_1 \cdot r_1 \cdot [P_{bestj} - x_j(i-1)] + c_2 \cdot r_2 \cdot [G_{bestj} - x_j(i-1)] \quad \dots\dots\dots(4)$$

$$X_j(i) = X_j(i-1) + V_j(i) \quad \dots\dots\dots(5)$$

Dimana, $V_j(i)$ dan $V_j(i-1)$ merupakan kecepatan partikel ke- j yang sekarang dan sebelumnya. $X_j(i)$ dan $X_j(i-1)$ merupakan posisi partikel ke- j yang sekarang dan sebelumnya. w , c_1 dan c_2 merupakan bobot inersia, kognitif, dan sosial. r_1 dan r_2 bilangan acak, P_{best} menyatakan posisi yang terbaik untuk partikel ke- j dan G_{best} menyatakan partikel yang terbaik dari semua partikel.

2.2.1. Cauchy Mutation

Operasi *cauchy mutation* merupakan algoritma yang digunakan untuk memperbaiki kekurangan PSO yang cenderung terjebak dalam lokal optimum yang menghasilkan *velocity* dan posisi partikel baru dengan Persamaan 6 dan 7 [3].

$$V_j(t) = V_j(t-1) \cdot \exp(\vartheta) \dots\dots\dots(6)$$

$$X_j(t) = X_j(t-1) + V_j \cdot \vartheta_i \dots\dots\dots(7)$$

Dimana: ϑ dan ϑ_i merupakan angka random *cauchy*. *Pseudocode* pembangkit angka *random* sebagai berikut [6]:

```

1. Start
2. RR = 1
3. While (RR >= 1)
4.     V1 = 2 * rand() - 1
5.     V2 = 2 * rand() - 1
6.     RR = V12 + V22
7. EndWhile
8. CauchyRandomNumber = V1 / V2
9. End

```

Gambar 1. Algoritma *cauchy random*.

2.2.2. Roulette Wheel Selection

Pada PSO, semua *parent* akan diganti dengan *offspring* yang baru, meskipun *offspring* tersebut belum tentu memiliki *fitness* yang lebih baik. Dengan *roulette wheel selection*, *parent* dan *offspring* dikumpulkan dan dihitung besar kemungkinan terpilih berdasarkan *fitness*. Langkah – langkah penerapan *roulette wheel selection* ini adalah sebagai berikut [3]

```

1. Hitung kemungkinan terpilihnya partikel berdasarkan fitness.
2. Hasilkan sebuah angka random.
3. Lakukan pengecekan untuk setiap kemungkinan partikel. Jika angka random lebih besar dari kemungkinan partikel(i) dan angka random lebih kecil dari kemungkinan partikel(i+1), maka partikel(i) terpilih sebagai offspring baru.
4. Ulangi langkah 2, sampai jumlah offspring memenuhi jumlah maks. partikel

```

Gambar 2. *Pseudocode roulette wheel selection*.

2.2.3. Hybrid of Particle Swarm Optimization (HPSO)

Penerapan HPSO [3] merupakan pengabungan PSO dengan penambahan operasi *cauchy mutation* dan *roulette wheel selection* dimana dapat mencegah terjebak pada lokal optimum. *Pseudocode* HPSO adalah sebagai berikut:

```

1. partikel awal dihasilkan secara random baik dari posisi dan velocity
2. Evaluasi fitness untuk setiap partikel
3. Jika fitness partikel(i) lebih baik dari Pbest, update Pbest.
4. Untuk setiap partikel, jika fitness partikel(i) lebih baik dari Gbest, update Gbest.
5. Untuk setiap partikel, lakukan
  a. Bentuk partikel (t) dengan persamaan (4) dan (5)
  b. Bentuk partikel (t') dengan persamaan (6) dan (7)
  c. Bandingkan (t) dan (t'), pilih salah satu dengan fitness yang terbaik sebagai offspring.
6. Hasilkan partikel berikutnya dengan roulette wheel selection
7. Ulangi tahap no.3 sampai jumlah iterasi yang diinginkan atau sampai hasil / fitness yang diperoleh memuaskan.

```

Gambar 3. *Pseudocode HPSO*.

Masing – masing partikel akan dievaluasi dengan fungsi *fitness* dan fungsi *constraint*. Berikut ini merupakan 3 variasi untuk persamaan fungsi *fitness* [3]:

$$fitness_{IAE} = \sum_{i=1}^8 \sum_{j=1}^8 |WB(i,j) - OB(i,j)| \dots\dots\dots(8)$$

$$fitness_{ISE} = \sum_{i=1}^8 \sum_{j=1}^8 (WB(i,j) - OB(i,j))^2 \dots\dots\dots(9)$$

$$fitness_{MSE} = \frac{1}{8 \times 8} \sum_{i=1}^8 \sum_{j=1}^8 (WB(i,j) - OB(i,j))^2 \dots\dots\dots(10)$$

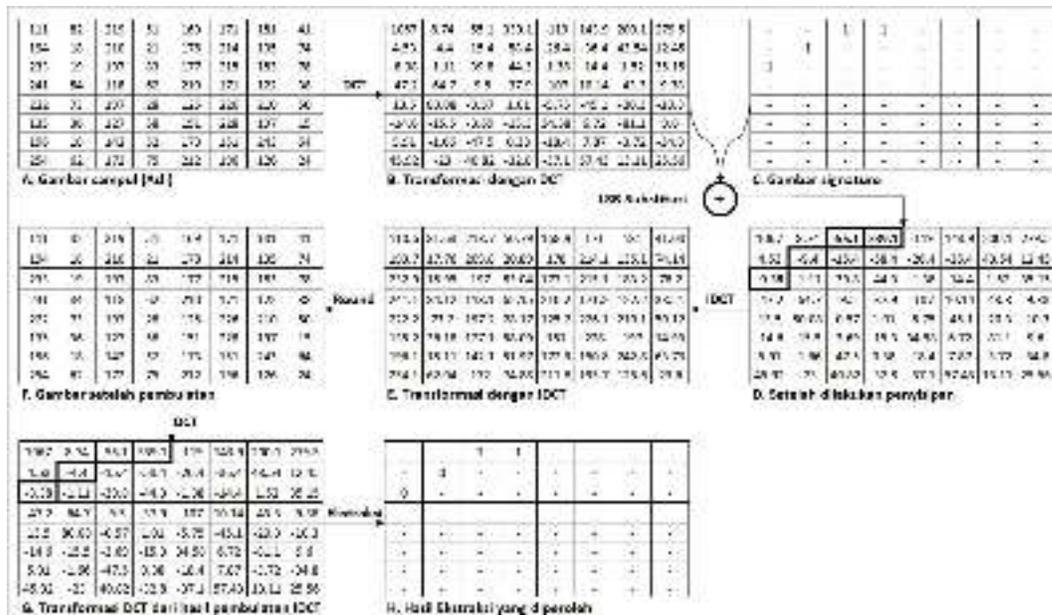
Dimana, *watermarked block* (*WB*) dan *original block* (*OB*). Dan fungsi *constraint*, dimana *WMBlock^o* merupakan *original watermark* dan *WMBlock^w* merupakan *watermark* yang diekstraksi. Persamaannya adalah sebagai berikut [3]:

$$C = \sum_{i=1}^4 |WMBlock_i^o - WMBlock_i^w| \dots\dots\dots(11)$$

3. Metode Penelitian

3.1. Analisis Masalah

Permasalahan utama yang ditimbulkan dengan pembulatan sederhana ditunjukkan pada Gambar 4. Gambar 4A adalah satu blok gambar sampel yang berukuran 8x8 piksel dalam domain spasial dan Gambar 4B adalah gambar sampel yang telah ditransformasi ke domain frekuensi dengan DCT. Gambar 4C merupakan blok tanda tangan dengan format biner "0" dan "1" menyatakan nilai yang akan ditanam pada lokasi tersebut. Proses penyisipan dilakukan dengan substitusi LSB dengan nilai bulat (*truncate*) absolut dari posisi gambar sampel (Contohnya: "-55.13" menjadi "55") sehingga diperoleh Gambar 4D. Lalu dilakukan transformasi ke domain spasial dengan IDCT, sehingga diperoleh hasil yang ditunjukkan pada Gambar 4E dimana dihasilkan satu blok bilangan riil yang harus dibulatkan. Namun hasil pembulatan tersebut mengalami penyimpangan dalam menerjemahkan bilangan riil ke bilangan bulat, membuat hasil ekstraksi (yaitu: "0011") tidak sesuai dengan tanda tangan yang ditanamkan sebelumnya (yaitu: "1111").



Gambar 4. Ilustrasi penyimpangan dalam pembulatan IDCT.

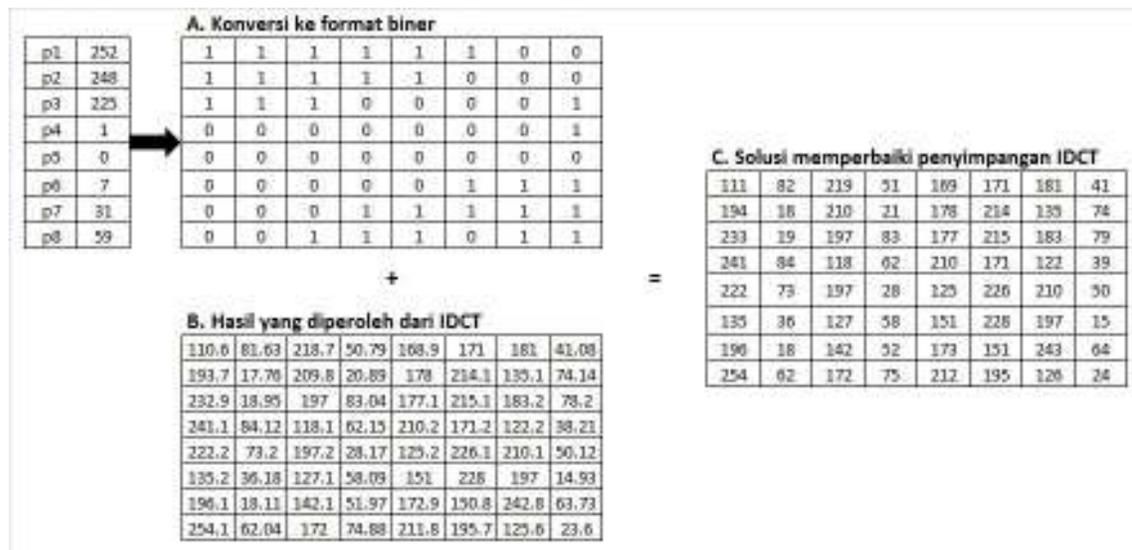
Gambar 4D menunjukkan lokasi penyisipan dilakukan terhadap gambar sampul dengan substitusi LSB [3]. Pada Tabel 1 ditunjukkan kemungkinan penyimpangan yang terjadi.

Tabel 1. Total Kemungkinan Penyimpangan dengan Pembulatan.

Ditanam	Diekstraksi	Bits Error	Ditanam	Diesktraksi	Bits Error
0000	0011	2	1000	0011	3
0001	0011	1	1001	0011	2
0010	0011	1	1010	0011	2
0011	0011	0	1011	0011	1
0100	0000	1	1100	0000	2
0101	0011	2	1101	0011	3
0110	0011	2	1110	0010	2
0111	0011	1	1111	0011	2

3.2. Usulan Metode Peningkatan Kualitas

Untuk memperbaiki penyimpangan tersebut, HPSO menghasilkan sebuah set solusi yang terdiri dari 8 buah bilangan bulat yang akan menjadi formula untuk menghindari kesalahan dalam pembulatan bilangan riil. Gambar 5 menunjukkan set solusi yang diperoleh dari proses HPSO.



Gambar 5. Set solusi dan formula memperbaiki pembulatan.

Gambar 5A merupakan satu set solusi yang akan dijumlahkan dengan nilai bulat (*truncate*) untuk setiap elemen pada hasil yang diperoleh dengan IDCT pada Gambar 5B. Sebagai contohnya, nilai dari “110.61” adalah “110” dan diperoleh “111” sehingga diperoleh Gambar 5C. Untuk menentukan kualitas dari gambar sampul digunakan persamaan *mean squared error* (MSE) dan *peak signal to noise ratio* (PSNR). MSE dari gambar sampul yang asli (Y) dengan gambar sampul telah diberikan tanda tangan (Y^W) dapat dihitung dengan menggunakan Persamaan 12:

$$MSE = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [Y(m,n) - Y^W(m,n)]^2 \dots\dots\dots(12)$$

MSE sangat tergantung pada intensitas skala gambar, semakin besar perbedaannya mengakibatkan hasil MSE semakin besar. PSNR memperbaiki masalah tersebut dengan mengukur skala rentang gambar dapat dihitung dengan Persamaan 13:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) (dB) \dots\dots\dots(13)$$

Dalam kompresi gambar, nilai PSNR yang ditoleransi terhadap penglihatan mata antara 30 – 50 dB, dimana lebih tinggi akan lebih baik. *Normalized cross correlation* (NC) secara umum digunakan untuk mengukur kesamaan antara 2 buah gambar, jika NC = 1 maka kedua gambar dinyatakan sama. NC dapat dihitung dengan Persamaan 14:

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N [w(i,j) \times w'(i,j)]}{\sum_{i=1}^M \sum_{j=1}^N [w(i,j)]^2} \dots\dots\dots(14)$$

3.3. Skenario Evaluasi Usulan Metode Peningkatan Kualitas

Dalam memperbaiki penyimpangan dalam pembulatan, proses penerapan *fragile image watermarking* dengan HPSO berbasis DCT. Dimana metode yang digunakan bisa berupa pembulatan sederhana dan HPSO yang dijelaskan sebagai berikut:

1. Y = Gambar sampul dibagi menjadi 8x8 piksel per blok
2. W = Gambar signature dibagi menjadi 2x2 piksel per blok
3. P = Kata sandi yang dikonversi ke format biner
4. Untuk setiap blok Y, lakukan:
 - 4.a. Hitung DCT, $Y^{DCT}_{[i]} = DCT(Y_{[i]})$
 - 4.b. Sisipkan blok $W_{[i]}$ dengan substitusi LSB diperoleh $Y^{WDCT}_{[i]}$
 - 4.c. Hitung IDCT, $Y^{WIDCT}_{[i]} = IDCT(Y^{WDCT}_{[i]})$
 - 4.d. Jika metode pembulatan?
 - i. $Y^*_{[i]} = Round(Y^{WIDCT}_{[i]})$
 - 4.e. Jika HPSO
 - i. $Y^W_{[i]} = HPSO(Y^{WIDCT}_{[i]})$
 - 4.f. Sisipkan 1 bit Kata Sandi ($P_{[i]}$) pada $Y^*_{[i]}$
 - 4.g. Ulangi langkah 4, sampai seluruh W telah disisipkan
5. Output Y^W , gambar sampul yang telah disisipkan tanda tangan.

Gambar 6. Psuedocode penyisipan gambar tanda tangan.

3.4. Skenario Ekstraksi Tanda Tangan

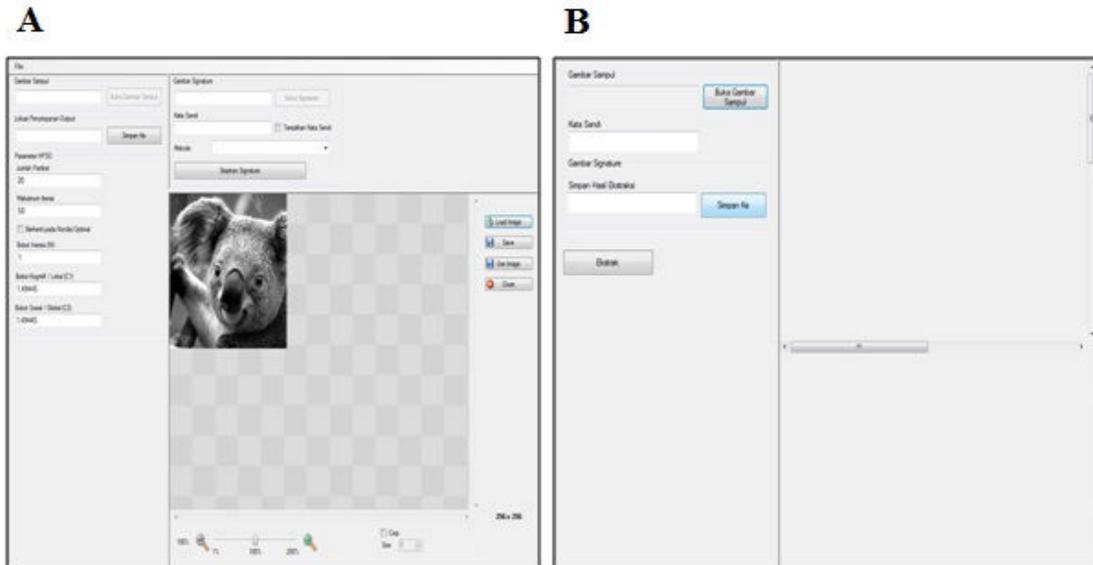
Skenario ini digunakan untuk proses ekstraksi gambar tanda tangan, dimana pengguna diharuskan memasukkan kata sandi yang sesuai untuk dapat melanjutkan proses ekstraksi. Proses untuk mengekstraksi tanda tangan pada gambar sampul dideskripsikan sebagai berikut:

1. Y^w = Gambar sampul yang telah disisipkan tanda tangan dibagi menjadi 8×8 piksel per blok.
2. P = Kata sandi untuk mengekstraksi tanda tangan
3. P^E = Ekstraksi kata sandi pada gambar Y^w
4. Jika $P \neq P^E$
 - 4.a. Pesan kesalahan: "Tidak dapat melakukan proses ekstraksi"
5. Jika $P = P^E$
 - 5.a. Untuk setiap blok Y^w , dilakukan:
 - i. Hitung koefisien DCT, $Y^{wDCT}_{[i]} = DCT(Y^w_{[i]})$
 - ii. X = Nilai bulat (*truncate*) dari absolut pada lokasi penyisipan diperoleh dan dikonversi ke format biner
 - iii. $W_{[i]}$ = LSB dari X , mengembalikan blok gambar tanda tangan
 - 5.b. Output W , gambar tanda tangan yang diekstraksi
6. Selesai.

Gambar 7. Psuedocode ekstraksi gambar tanda tangan.

4. Hasil dan Pembahasan

Hasil implementasi pada aplikasi berupa penyisipan tanda tangan, ekstraksi tanda tangan, perbandingan gambar terhadap gambar asli dan perbandingan antara pembulatan sederhana dengan HPSO dalam hal MSR, PSNR dan NC.



Gambar 8. Gambar tampilan aplikasi penyisipan (A) dan ekstraksi (B).

Gambar 8A merupakan tampilan untuk penyisipan tanda tangan dimana terdapat tombol buka gambar sampul untuk membuka gambar sampul, tombol buka *signature* untuk membuka gambar tanda tangan, tombol *load image* untuk memilih gambar dari komputer yang akan digunakan, tombol *save* untuk menyimpan gambar, tombol *use image* untuk menggunakan gambar yang telah dipilih, tombol *close* untuk keluar, tombol *resize* untuk memperbesar atau memperkecil gambar, *checkbox* *crop* untuk memotong gambar sesuai dengan ukuran pada *textbox*, tombol *simpan ke* untuk memilih lokasi penyimpanan gambar sampul yang telah

disisipkan, *textbox* kata sandi untuk memasukkan kata sandi yang akan digunakan pada ekstraksi tanda tangan, *dropdown* metode untuk pemilihan metode yang digunakan yaitu pembulatan sederhana dan HPSO, *textbox* parameter HPSO yaitu jumlah partikel untuk memasukkan jumlah partikel, maksimum iterasi untuk memasukkan jumlah maksimum iterasi, bobot inersia untuk memasukkan nilai bobot inersia (W), bobot kognitif untuk memasukkan nilai bobot kognitif ($C1$), bobot sosial untuk memasukkan nilai bobot sosial ($C2$), *checkbox* untuk berhenti pada kondisi optimal dan tombol sisipkan *signature* untuk menjalankan proses penyisipan.

Gambar 8B merupakan tampilan untuk ekstraksi tanda tangan dimana terdapat tombol buka gambar sampul untuk membuka gambar sampul yang akan diekstraksi, *textbox* kata sandi untuk memasukkan kata sandi sesuai dengan kata sandi yang telah dimasukkan pada proses penyisipan, tombol simpan ke untuk memilih lokasi penyimpanan hasil ekstraksi tanda tangan dan tombol ekstrak untuk menjalankan proses ekstraksi.



Gambar 9. Gambar Tampilan Aplikasi Perbandingan

Gambar 9 merupakan tampilan untuk melakukan perbandingan dari gambar yang satu dengan gambar yang lainnya dimana terdapat 3 tombol buka gambar untuk membuka gambar yaitu tombol buka gambar bagian kiri untuk membuka gambar A, tombol buka gambar bagian tengah untuk membuka gambar sampul atau tanda tangan yang asli, dan tombol buka gambar bagian kanan untuk membuka gambar B, tombol bandingkan gambar untuk menjalankan proses perbandingan yaitu gambar A (kiri) dan gambar B (kanan) akan dibandingkan dengan gambar di bagian tengah.

Untuk mendemonstrasikan hasil ekstraksi gambar tanda tangan, dilakukan 3 buah pengujian dengan ukuran gambar sampul yang berbeda yang ditunjukkan pada Gambar 10. Untuk gambar tanda tangan yang digunakan berukuran 64x64 piksel seperti yang ditunjukkan pada Gambar 11.



A. Gambar Pantai
(1024x1024)



B. Gambar Cameraman
(512x512)



C. Gambar Panda
(256x256)

Gambar 10. Gambar sampul yang akan disisipkan tanda tangan.



A. Gambar Wajah



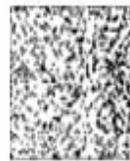
B. Gambar Daun



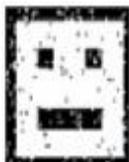
C. Gambar Fu

Gambar 11. Gambar tanda tangan.

Setelah dilakukan proses penyisipan ke dalam gambar sampul, lalu dilakukan proses ekstraksi gambar tanda tangan untuk dilakukan perbandingan terhadap metode pembulatan sederhana dan HPSO. Hasil ini dapat dilihat pada Gambar 12 dan 13, dimana penerapan HPSO dapat dengan mudah diverifikasi hanya dengan penglihatan mata.



Gambar 12. Hasil ekstraksi dengan pembulatan sederhana.



Gambar 13. Hasil ekstraksi dengan penerapan HPSO.

Pada Tabel 2, ditunjukkan nilai MSE dan PSNR untuk gambar sampul yang telah disisipkan gambar tanda tangan terhadap gambar sampul asli.

Tabel 2. Hasil MSE, PSNR gambar sampul yang diperoleh dari proses pengujian

Gambar	Pembulatan Sederhana		HPSO	
	MSE	PSNR	MSE	PSNR
Gambar Pantai	0.01	68.09	0.42	51.95
Gambar Cameraman	0.01	67.48	0.43	51.79
Gambar Panda	0.02	66.48	0.45	51.63

Pada Tabel 3, ditunjukkan nilai NC untuk gambar tanda tangan yang telah diekstraksi dari gambar sampul terhadap gambar tanda tangan asli.

Tabel 3. Hasil NC gambar tanda tangan yang diperoleh dari proses pengujian

Gambar	Pembulatan Sederhana	HPSO
	NC	NC
Gambar Wajah	0.93	0.98
Gambar Daun	0.73	0.93
Gambar Fu	0.56	0.89

Dari pengujian yang telah didemonstrasikan terbukti bahwa dengan penerapan HPSO hasil ekstraksi gambar tanda tangan yang diperoleh memiliki kualitas dan tingkat keakuratan yang lebih baik dibandingkan hanya dengan pembulatan sederhana. Dapat dilihat pada tanda tangan yang menggunakan metode HPSO memiliki nilai NC yang mendekati 1 daripada tanda tangan yang menggunakan metode pembulatan sederhana walaupun nilai MSE dan PSNR yang dimiliki gambar sampul dengan metode pembulatan sederhana lebih baik daripada metode HPSO karena banyaknya perubahan yang terjadi pada gambar sampul dari proses HPSO.

5. Kesimpulan

Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa hasil gambar sampul yang disisipkan gambar tanda tangan, memiliki rata – rata PSNR dengan pembulatan sederhana (67.39 dB) lebih baik dibandingkan HPSO (51.88 dB). Hal ini diakibatkan karena banyaknya perubahan piksel yang terjadi dengan penerapan HPSO untuk memperbaiki penyimpangan dalam proses pembulatan. Disisi lain, hasil ekstraksi gambar tanda tangan berdasarkan nilai rata – rata NC yang diperoleh HPSO (0.94) dan pembulatan sederhana (0.75) menunjukkan peningkatan yang signifikan dalam memperbaiki akurasi dan kualitas gambar tanda tangan.

Referensi

- [1] Cox, I. J., Miller, M. L., Bloom, J. A., 2002, *Digital Watermarking*, Morgan Kaufmann Publishers.
- [2] He, X., 2012, *Signal Processing, Perceptual Coding and Watermarking of Digital Audio Advanced Technologies and Models*, IGI Global.
- [3] Ghargory, S. M., 2011, *Hybrid of Particle Swarm Optimization with Evolutionary Operators to Fragile Image Watermarking Based DCT*. Computers and Systems Department, Electronics Research Institute, Cairo, Egypt.
- [4] Khayam, S. A., 2003, *The Discrete Cosine Transform (DCT) : Theory and Application*, Department of Electrical & Computer Engineering Michigan State University.
- [5] Kennedy, J., Eberhart, R., 1995, *Particle Swarm Optimization*, Purdue School of Engineering and Technology. Indianapolis.