TUGAS AKHIR – KI141502

STEGANOGRAFI MENGGUNAKAN RDE DAN *FUZZY LOGIC* UNTUK MENENTUKAN *EMBEDDING LEVEL* : STUDI KASUS PADA CITRA MEDIS

PUTU HARUM BAWA
5111100026

Dosen Pembimbing I
TOHARI AHMAD, SKom.,MIT., Ph.D.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2015

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT – KI141502

STEGANOGRAPHY USING RDE AND FUZZY LOGIC TO DETERMINE EMBEDDING LEVEL : CASE STUDY ON MEDICAL IMAGE

PUTU HARUM BAWA
5111100026

Supervisor
TOHARI AHMAD, SKom.,MIT., Ph.D.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2015

LEMBAR PENGESAHAN

STEGANOGRAFI MENGGUNAKAN RDE DAN *FUZZY LOGIC* UNTUK MENENTUKAN *EMBEDDING LEVEL* : STUDI KASUS PADA CITRA MEDIS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
PUTU HARUM BAWA
NRP: 5111 100 026

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. TOHARI AHMAD, S.Kom, MIT, PhD. - (Pembimbing 1)
   NIP: 19750525 200312 1 002

SURABAYA
JUNI 2015

STEGANOGRAFI MENGGUNAKAN RDE DAN *FUZZY LOGIC* UNTUK MENENTUKAN *EMBEDDING* LEVEL : STUDI KASUS PADA CITRA MEDIS

Nama Mahasiswa : Putu Harum Bawa
NRP : 5111 100 026
Jurusan : Teknik Informatika FTIF ITS
Dosen Pembimbing 1 : Tohari Ahmad, S.Kom., MIT., Ph.D.

Abstrak

Citra medis merupakan salah satu media yang digunakan dalam menganalisis kondisi pasien. Ketika sebuah citra medis dan rekam medis pasien diakses, terdapat beberapa hal yang perlu diperhatikan, diantaranya keamanan dan privasi data pasien. Salah satu solusi yang diterapkan untuk menyelesaikan masalah tersebut yaitu dengan memasukan identitas pasien ke properti citra medis. Namun metode tersebut masih memiliki kelemahan, sehingga digunakan metode lainnya yang lebih aman yaitu dengan menyembunyikan identitas pasien menggunakan metode steganografi.

Pada tugas akhir ini dilakukan pengembangan pada metode penyisipan dengan metode RDE yang dikombinasikan dengan logika fuzzy. Metode RDE dapat merekonstruksi kembali citra medis ke bentuk semula setelah pesan rahasia diterima, dan dengan memanfaatkan logika fuzzy maka penyisipan dapat dilakukan secara adaptif yang berpengaruh pada peningkatan nilai PSNR citra stego.

Hasil uji coba menunjukkan bahwa steganografi mengunakan RDE dan logika fuzzy yang diterapkan secara multi lapis memiliki rata-rata nilai PSNR lebih tinggi dibandingkan hasil metode RDE multi lapis tanpa logika fuzzy.

Kata kunci: steganografi, RDE, logika fuzzy, citra medis

STEGANOGRAPHY USING RDE AND FUZZY LOGIC TO DETERMINE EMBEDDING LEVEL : CASE STUDY ON MEDICAL IMAGE

Student's Name : Putu Harum Bawa
Student's ID : 5111 100 026
Department : Informatics Engineering, FTIF-ITS
First Advisor : Tohari Ahmad, S.Kom., MIT., Ph.D.

Abstract

Medical images is one of the media used in analyzing condition of patient. When a medical images and patient medical records are accessed, there are some points that need to be considered, they are security and privacy of patient data. One of the solution that can be applied to solve the problem is to put the patient's identity into medical image property. However, this methods still have drawbacks, so other safer method is to hide the identity of the patient using steganography.

At this final project, the development is done on combining RDE and fuzzy logic. RDE method can reconstruct medical images into former images after secret message is received, and by using fuzzy logic, the insertion can beadaptive that can increase PSNR value of stego image.

Based on trials, steganography using RDE and fuzzy logic that applied in multi-layer has highter average value of PSNR than the RDE method that applied in multi-layer without fuzzy logic.

Keywords: steganography, RDE, fuzzy logic, medical image

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, Ida Sang Hyang Widhi Wasa, yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Steganografi Menggunakan RDE dan *Fuzzy Logic* untuk Menentukan *Embedding Level* : Studi Kasus pada Citra Medis" dengan tepat waktu.

Pengerjaan Tugas Akhir ini merupakan kesempatan yang sangat berharga bagi penulis, terutama untuk memperdalam dan menerapkan hal-hal yang pernah penulis pelajari selama perkuliahan.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, Ida Sang Hyang Widhi Wasa atas berkah yang tiada habisnya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Kedua orang tua penulis, I Ketut Suwandi dan Ni Ketut Tambun yang selalu memberikan dukungan bagi penulis untuk meraih pengalaman dan pengetahuan di perantauan. Begitu pula kedua saudara penulis Kadek Putri Gyani dan Komang Andhita Gewana yang selalu memberikan semangat serta hiburan pada penulis ketika mengerjakan Tugas Akhir ini.
3. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. selaku dosen pembimbing penulis yang telah memberikan bimbingan dalam penyelesaian Tugas Akhir ini.
4. Ibu Dr. Ir. Siti Rochimah, MT. selaku dosen wali penulis.
5. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Dr. Eng. Radityo Anggoro S.Kom., M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.

6. Administrator Laboratorium Arsitektur dan Jaringan Komputer, Uyung, Vivi, Romen, Dimas, Sam, serta admin lain yang penulis tidak dapat sebut satu persatu yang telah membantu dan berbagi ilmu serta memotivasi penulis untuk menyelesaikan Tugas Akhir ini.
7. Teman-teman kontrakan F7, Arik, Pekak, Togir, Acong yang selalu siap berbagi keceriaan.
8. Teman-teman lubak forever Dia, Rickson, Yandi, Desta, Prana yang selalu berbagi kesulitan Tugas Akhir bersama.
9. Eka Sumarayeni yang selalu memberi semangat dan mengingatkan penulis agar segera menyelesaikan Tugas Akhir ini.
10. Warga lantai 4, Mas Jumali, Ali, Uyung, Yusro, Thiar, Surya, Pak Yudhi, Pak Was, Pak Dwi, Pak Royan, Pak Victor, Pak Bas dan Pak Hudan atas bimbingan ilmunya selama ini.
11. Rekan-rekan Tim Pembina Kerohanian Hindu (TPKH) ITS yang telah memberikan dukungan, dan semangat kepada penulis untuk menyelesaikan Tugas Akhir ini.
12. Teman-teman Teknik Informatika 2011 yang yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis.
13. Serta seluruh pihak-pihak lain yang belum sempat disebutkan satu per satu di sini yang telah membantu terselesaikannya Tugas Akhir ini.

Penulis mengharapkan saran dan kritik yang membangun dari pembaca untuk perbaikan ke depan. Semoga Tugas Akhir ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan bagi semua pihak.

Surabaya, Juni 2015

Putu Harum Bawa

DAFTAR ISI

LEMBAR PENGESAHAN ..... **v**
Abstrak ..... **vii**
Abstract ..... **ix**
KATA PENGANTAR ..... **xi**
DAFTAR ISI ..... **xiii**
DAFTAR GAMBAR ..... **xvii**
DAFTAR TABEL ..... **xxi**
DAFTAR KODE SUMBER ..... **xxiii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Permasalahan 2
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 3
1.5 Metodologi 3
1.6 Sistematika Laporan 4
BAB II TINJAUAN PUSTAKA **5**
2.1. Citra Digital 5
2.1.1. Citra *Grayscale* 6
2.1.2. Piksel Tetangga 6
2.1.3. *Entropy* 7
2.1.4. Tingkat Kecerahan 8
2.2. Citra Medis 8
2.3. Steganografi 9
2.4. *Difference Expansion* 10
2.5. RDE (*Reduced Difference Expansion)* 13
2.5.1. RDE dengan Selisih Piksel Negatif 15
2.5.2. Skema Penyisipan Data Multi Lapis 16
2.6. Logika *Fuzzy* 19
2.6.1. Himpunan *Fuzzy* 19
2.6.2. Fuzzifikasi dan Fungsi Keanggotaan *Fuzzy* 21
2.6.3. Aturan Kontrol *Fuzzy* 23
2.6.3.1. Aturan Pemetaan *Fuzzy* 23
2.6.3.2. Aturan Implikasi *Fuzzy* 24

2.6.4. Defuzzifikasi 24
2.6.5. Logika *Fuzzy* Mamdani 24
2.7. Skema Steganografi Adaptif berdasarkan *Fuzzy Inference System* (FIS).....26
2.8. MATLAB.....28
2.8.1. Entropyfilt 28
2.8.2. *Rangefilt* 28
2.8.3. *Stdfilt* 28
2.8.4. *Fuzzy Logic Toolbox* 29
BAB III PERANCANGAN.....31
3.1. Deskripsi Umum Metode 31
3.2. Perancangan Skema Metode Keseluruhan 31
3.3. Perancangan Penyisipan dan Ekstraksi Bit 33
3.4. Perancangan Mekanisme Penyisipan Multi Lapis.....33
3.4.1. Penentuan Blok 33
3.4.2. Struktur Level *Map*, Daftar Blok *Expandable*, dan Daftar Blok Penyisipan.....34
3.4.3. Penyisipan Bit dan Metode Urutan Pasangan Piksel.....35
3.4.4. Struktur Location Map 39
3.4.5. Ekstraksi Bit Multi Lapis 39
3.5. Perancangan Mekanisme Penentuan Level Penyisipan.....39
3.5.1. Perancangan Variabel Fuzzy.....39
3.5.2. Perancangan Fuzzifikasi dan Fungsi Keanggotaan *Fuzzy*.46
3.5.3. Perancangan Aturan Kontrol Fuzzy 50
3.5.4. Perancangan Defuzzyfikasi 52
3.6. Rancangan Antarmuka 52
BAB IV IMPLEMENTASI.....55
4.1. Lingkungan Implementasi.....55
4.1.1. Perangkat Keras.....55
4.1.2. Perangkat Lunak.....55
4.2. Implementasi RDE (*Reduced Difference Expansion*) 55
4.3. Implementasi Penyisipan Pesan Rahasia Multi Lapis 58
4.4. Implementasi Ekstraksi Pesan Rahasia Multi Lapis.....60
4.5. Implementasi Sistem Inferensi *Fuzzy*.....61
4.3.1. Implementasi Fungsi Keanggotaan 61

4.3.2. Implementasi Aturan Kontrol 63
4.6. Implementasi Antarmuka 67
BAB V UJI COBA DAN EVALUASI 69
5.1. Lingkungan Uji Coba 69
5.2. Data Uji Coba 70
5.3. Skenario Uji Coba 73
5.3.1. Skenaro Uji Coba Fungsionalitas 73
5.3.2. Skenaro Uji Coba Performa 75
5.4. Uji Coba 76
5.4.1. Uji Coba Fungsionalitas 76
5.4.2. Uji Coba Performa 77
5.5. Evaluasi Uji Coba 83
5.5.1. Evaluasi Uji Coba Fungsionalitas 83
5.5.2. Evaluasi Uji Coba Performa 83
BAB VI KESIMPULAN DAN SARAN 93
6.1. Kesimpulan 93
6.2. Saran 94
LAMPIRAN 97
Lampiran 1. Kumpulan Kode Sumber yang Digunakan 97
Lampiran 2. Perbandingan Metode yang Diusulkan 110
BIODATA PENULIS 113

DAFTAR GAMBAR

Gambar 2-1 Koordinat Citra Digital 5
Gambar 2-2 Radiograf Dada 9
Gambar 2-3 Lapis Pertama Dengan Penyisiran Secara Horizontal 17
Gambar 2-4 Lapis Kedua dengan Penyisiran Secara Vertikal 17
Gambar 2-5 Urutan Penyisiran Secara Horizontal 18
Gambar 2-6 Urutan Penyisiran Secara Vertikal 18
Gambar 2-7 Himpunan Tegas [12].....20
Gambar 2-8 Himpunan *Fuzzy* [12].....21
Gambar 2-9 Fungsi Keanggotaan Masukan [12].....22
Gambar 2-10 Fungsi Keanggotaan Hasil [12].....23
Gambar 2-11 Diagram Prosedur Penyisipan Secara Adaptif [5].27
Gambar 2-12 Fungsi Keanggotaan dan Kelas Semantik; (A) *Input* Karakteristik T_k, (B) *Input* Karakteristik E_k, (C) *Input* Karakteristik B_k, (D) *Output* [5].....27
Gambar 3-1 Blok Diagram Prosedur Penyisipan RDE Multi Lapis Menggunakan Logika *Fuzzy* 32
Gambar 3-2 Blok Piksel 33
Gambar 3-3 Urutan Penyisiran Blok 34
Gambar 3-4 Contoh Level *Map*.....34
Gambar 3-5 Contoh Daftar Blok *Expandable* 35
Gambar 3-6 Urutan Pasangan Piksel Horizontal.....36
Gambar 3-7 Urutan Pasangan Piksel Horizontal Vertikal pada Lapis Genap.....36
Gambar 3-8 Urutan Pasangan Piksel Horizontal Vertikal pada Lapis Ganjil 37
Gambar 3-9 Urutan Pasangan Piksel Terdefinisi pada Lapis 1 ..37
Gambar 3-10 Urutan Pasangan Piksel Terdefinisi pada Lapis 2 37
Gambar 3-11 Urutan Pasangan Piksel Terdefinisi pada Lapis 3 37
Gambar 3-12 Urutan Pasangan Piksel Terdefinisi pada Lapis 4 38
Gambar 3-13 Urutan Pasangan Piksel Terdefinisi pada Lapis 5 38
Gambar 3-14 Urutan Pasangan Piksel Terdefinisi pada Lapis 6 38
Gambar 3-15 Urutan Pasangan Piksel Terdefinisi pada Lapis 7 38

Gambar 3-16 Urutan Pasangan Piksel Terdefinisi pada Lapis 8 38
Gambar 3-17 Contoh Hasil Penyisipan Bit 1 40
Gambar 3-18 *Pseudocode* Pembuatan Sampel Blok Skenario 1 .41
Gambar 3-19 *Pseudocode* Pembuatan Sampel Blok Skenario 2 .41
Gambar 3-20 *Pseudocode* Pembuatan Sampel Blok Skenario 3 .41
Gambar 3-21 *Pseudocode* Pembuatan Sampel Blok Skenario 4 .41
Gambar 3-22 Grafik PSNR Sampel Blok Level Penyisipan 1 42
Gambar 3-23 Grafik PSNR Sampel Blok Level Penyisipan 2 43
Gambar 3-24 Grafik PSNR Sampel Blok Level Penyisipan 3 43
Gambar 3-25 Grafik PSNR Sampel Blok Level Penyisipan 4 44
Gambar 3-26 Grafik PSNR Sampel Blok Level Penyisipan 5 44
Gambar 3-27 Grafik PSNR Sampel Blok Level Penyisipan 6 45
Gambar 3-28 Grafik PSNR Sampel Blok Level Penyisipan 7 45
Gambar 3-29 Grafik PSNR Sampel Blok Level Penyisipan 8 46
Gambar 3-30 Rancangan jendela utama 53
Gambar 4-1 *Pseudocode* Penyisipan Menggunkan RDE 56
Gambar 4-2 *Pseudocode* Ekstraksi Menggunakan RDE 58
Gambar 4-3 *Pseudocode* Penyisipan Multi Lapis 60
Gambar 4-4 *Pseudocode* Ekstraksi Multi Lapis 61
Gambar 4-5 Bagan Keseluruhan Implementasi Fungsi Keanggotaan *Fuzzy*.....62
Gambar 4-6 Grafik Fungsi Keanggotaan *Entropy* Lokal 62
Gambar 4-7 Grafik Fungsi Keanggotaan Jarak Lokal 62
Gambar 4-8 Grafik Fungsi Keanggotaan Standar Deviasi Lokal 62
Gambar 4-9 Implementasi Aturan *Entropy* Lokal 63
Gambar 4-10 Implementasi Aturan Jarak Lokal 63
Gambar 4-11 Implementasi Aturan Standar Deviasi Lokal 63
Gambar 4-12 Implementasi Kombinasi Aturan *Entropy* Lokal Dan Jarak Lokal 64
Gambar 4-13 Implementasi Kombinasi Aturan *Entropy* Lokal Dan Standar Deviasi Lokal 64
Gambar 4-14 Implementasi Kombinasi Aturan Jarak Lokal Dan Standar Deviasi Lokal 65
Gambar 4-15 Implementasi Kombinasi Aturan Entropy Lokal, Jarak Lokal Dan Standar Deviasi Lokal 66

Gambar 4-16 Tampilan Jendela Utama 67
Gambar 5-1 Radiograf dada 70
Gambar 5-2 Radiograf tangan 71
Gambar 5-3 Radiograf kaki 71
Gambar 5-4 Radiograf paru-paru 72
Gambar 5-5 Radiograf leher 72
Gambar 5-6 Pesan rahasia pada uji coba fungsionalitas 75
Gambar 5-7 Grafik PSNR Penyisipan 50kbit pada Urutan Pasangan Piksel Horizontal Vertikal 84
Gambar 5-8 Grafik PSNR Penyisipan 250 Kbit pada Urutan Pasangan Piksel Horizontal Vertikal 85
Gambar 5-9 Grafik PSNR Penyisipan 500 Kbit pada Urutan Pasangan Piksel Horizontal Vertikal 85
Gambar 5-10 Grafik PSNR Penyisipan 50 Kbit pada Urutan Pasangan Piksel Terdefinisi 86
Gambar 5-11 Grafik PSNR Penyisipan 250 Kbit pada Urutan Pasangan Piksel Terdefinisi 86
Gambar 5-12 Grafik PSNR Penyisipan 500 Kbit pada Urutan Pasangan Piksel Terdefinisi 87
Gambar 5-13 Perbandingan PSNR Penyisipan 50 kbit 88
Gambar 5-14 Perbandingan PSNR Penyisipan 250 kbit 88
Gambar 5-15 Perbandingan PSNR Penyisipan 500 kbit 89
Gambar 5-16 Grafik PSNR Penyisipan Maksimal pada Urutan Pasangan Piksel Horizontal Vertikal 90
Gambar 5-17 Grafik PSNR Penyisipan Maksimal pada Urutan Pasangan Piksel Terdefinisi 90
Gambar 5-18 Grafik Perbandingan PSNR Penyisipan Maksimal 91
Gambar 5-19 Kapasitas Penyisipan Urutan Pasangan Piksel Horizontal Vertikal 91
Gambar 5-20 Kapasitas Penyisipan Urutan Pasangan Piksel Terdefinisi 92
Gambar 5-21 Perbandingan Kapasitas Penyisipan 92

DAFTAR TABEL

Tabel 2-1 Contoh Aturan-Aturan Fuzzy [12] 23
Tabel 3-1 Fungsi Keanggotaan *Entropy* Lokal (Urutan Pasangan Piksel Horizontal Vertikal) 47
Tabel 3-2 Fungsi Keanggotaan Jarak Lokal (Urutan Pasangan Piksel Horizontal Vertikal) 48
Tabel 3-3 Fungsi Keanggotaan Standar Deviasi Lokal (Urutan Pasangan Piksel Horizontal Vertikal) 48
Tabel 3-4 Fungsi Keanggotaan Entropy Lokal (Urutan Pasangan Piksel Terdefinisi) 49
Tabel 3-5 Fungsi Keanggotaan Jarak Lokal (Urutan Pasangan Piksel Terdefinisi) 49
Tabel 3-6 Fungsi Keanggotaan Standar Deviasi Lokal (Urutan Pasangan Piksel Terdefinisi) 49
Tabel 3-7 Fungsi Keanggotaan *Output* Level Penyisipan 50
Tabel 3-8 Aturan Kontrol *Entropy* Lokal 51
Tabel 3-9 Aturan Kontrol Jarak Lokal 51
Tabel 3-10 Aturan Kontrol Standar Deviasi Lokal 51
Tabel 3-11 Kombinasi Aturan Kontrol Jarak Lokal dan Standar Deviasi Lokal 52
Tabel 5-1 Spesifikasi Komputer Uji Coba 69
Tabel 5-2 Daftar Ukuran Data Rahasia 73
Tabel 5-3 Skenario Uji Coba Fungsionalitas 74
Tabel 5-4 Hasil Uji Coba Fungsionalitas 76
Tabel 5-5 Hasil Uji Coba Berupa Nilai PSNR pada Penyisipan 50 Kbit, 250 Kbit dan 500 Kbit Data Rahasia 77
Tabel 5-6 Hasil Uji Coba Penyisipan Maksimal 80
Tabel 7-1 Perbandingan Metode yang Diusulkan dengan yang Sudah Ada Sebelumnya 110

DAFTAR KODE SUMBER

Kode Sumber 7-1 Implementasi Penyisipan Bit dengan Metode RDE 98
Kode Sumber 7-3 Implementasi Ekstraksi Bit dengan Metode RDE 100
Kode Sumber 7-4 Implementasi Penyisipan Pesan Rahasia ke Dalam Blok-Blok pada Citra 105
Kode Sumber 7-5 Implementasi Ekstraksi Pesan Rahasia ke Dalam Blok-Blok pada Citra 109

BAB I
PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari Tugas Akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan Tugas Akhir.

1.1 Latar Belakang

Citra medis merupakan salah satu media yang digunakan dalam menganalisis kondisi pasien. Dengan menggunakan citra medis, dokter dapat menentukan penanganan yang tepat bagi pasien. Citra medis tidak hanya digunakan oleh dokter yang menangani pasien bersangkutan saja, namun juga dapat dibagikan kepada dokter lain untuk mendapatkan penanganan yang lebih komprehensif.

Ketika sebuah citra medis dan rekam medis pasien diakses ataupun dikirim melalui jaringan komputer, terdapat beberapa hal yang perlu diperhatikan, diantaranya keamanan dan privasi data pasien. Salah satu solusi yang diterapkan untuk menyelesaikan masalah tersebut yaitu dengan memasukan identitas pasien ke *property* citra medis. Namun, metode ini masih memiliki kekurangan. Orang-orang yang tidak bertanggung jawab dapat membaca atau bahkan mengubah informasi pasien di dalamnya. Solusi lainnya yaitu dengan menerapkan steganografi, dengan cara menyisipkan identitas pasien ke dalam citra medis pasien bersangkutan [1].

Dalam steganografi, terdapat beberapa aspek penting yang perlu diperhatikan, diantaranya *imperceptibility*, *robustness* dan *capacity*. Dalam konteks steganografi pada citra medis terdapat aspek tambahan yang harus diperhatikan, yaitu citra medis harus dapat direkonstruksi kembali setelah pengambilan data rahasia. Jika citra medis tidak dapat direkonstruksi kembali akan berakibat pada kesalahan analisis oleh dokter.

Agar tidak terjadi kesalahan analisis akibat kualitas citra medis yang berubah, maka perlu pemilihan metode yang dapat merekonstruksi kembali citra medis ke bentuk semula setelah pesan rahasia diterima. Salah satu metode tersebut yaitu metode *reversible data hiding* seperti *Difference Expation* (DE) [2]. Terdapat beberapa pengembagnan metode DE seperti pada [3] dan [1] yang dapat meningkatkan kapasitas pesan yang disisipkan.

Akhir-akhir ini, *intelegent algorithm* berbasis *soft computing* mulai banyak digunakan dalam memecahkan permasalahan di bidang komputer. Algoritma tersebut antara lain, *Fuzzy Logic* (FL), *Adaptive Neural Network*s (ANNs), *Genetic Algorithms* (GA), dan *Particle Swarm Optimizer* (PSO). Algoritma tersebut mulai digunakan pada steganografi untuk mencapai hasil yang masksimal dan solusi yang adaptif. Beberapa pengembangan steganografi yang memanfaatklan soft computing yaitu seperti [4], [5] yang menggunakan *Fuzzy Logic* (FL). Namun kedua metode tersebut [4], [5] masih menggunakan LSB dalam proses penyisipan data, sehingga gambar awal yang digunakan dalam steganografi tidak dapat direkonstruksi kembali.

Pada Tugas Akhir ini, penulis berusaha melakukan kombinasi beberapa metode untuk meningkatkan *imperceptibility* pada hasil steganografi. Penulis mengusulkan sebuah metode yang mengkombinasikan metode steganografi dengan *Fuzzy Logic* (FL).

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Faktor apa saja yang dapat digunakan untuk membentuk keanggotaan *Fuzzy Logic* yang akan dirancang?
2. Bagaimana merancang *Fuzzy Inference System* yang dapat meningkatkan *imperceptibility* pada steganografi?

1.3 Batasan Masalah

Batasan masalah pada Tugas Akhir ini yaitu sebagai berikut:

1. Pengembangan terletak pada cara memanfaatkan *Fuzzy Logic* dalam membantu proses penyisipan data rahasia pada steganografi.
2. Citra medis yang digunakan dalam uji coba yaitu citra medis *grayscale* 8 *bit*.

1.4 Tujuan dan Manfaat

Tujuan Tugas Akhir ini adalah mengembangkan metode steganografi dengan memanfaatkan *Fuzzy Logic* untuk meningkatkan tingkat kemiripan citra medis awal dengan citra medis hasil steganografi.

Manfaat yang didapatkan dari pembuatan Tugas Akhir ini adalah terciptanya suatu pengembangan metode steganografi yang dapat diterapkan pada citra medis.

1.5 Metodologi

Metodologi yang digunakan pada pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Studi Literatur
   Pada proses ini dilakukan studi lebih lanjut terhadap konsep-konsep pada jurnal, buku, artikel, dan literatur yang terkait dengan Tugas Akhir. Studi dilakukan untuk mendalami konsep mengenai metode *Difference Expansion* yang sudah dikembangkan sebelumnya, metode *Multiple Layer Data Hiding*, serta pemanfaatan *Fuzzy Logic* pada Steganografi.
2. Implementasi Algoritma
   Implementasi merupakan tahap untuk membangun metode yang dikembangkan. Metode akan diimplementasikan menggunakan perangkan lunak MATLAB R2014a.
3. Uji Coba dan Evaluasi
   Uji coba dan evaluasi dilakukan dengan tahap:

1. Uji kualitas visual citra medis yang telah disisipkan data dengan jumlah bit tertentu. Kualitas visual citra diukur dengan menghitung *peak signal to noise ratio* (PSNR) citra tersebut.
2. Uji kapasitas data yang dapat disisipkan pada citra medis berdasarkan metode yang diusulkan.

4. Penyusunan Laporan Tugas Akhir
   Pada tahap ini, disusun buku berisi metode, dasar teori, hasil uji coba, dan hasil evaluasi dari Tugas Akhir.

1.6 Sistematika Laporan

Sistematika buku Tugas Akhir ini yaitu:

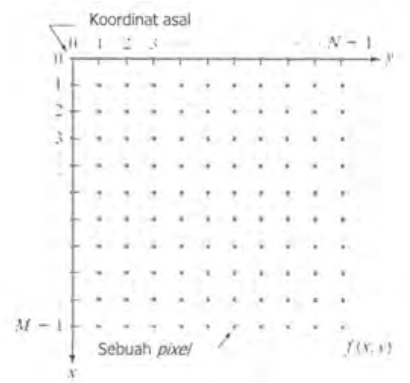
1. Bab I Pendahuluan
   Pada bab ini dibahas latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika laporan Tugas Akhir.
2. Bab II Tinjauan Pustaka
   Pada bab ini dibahas dasar teori yang berkaitan dengan topik Tugas Akhir.
3. Bab III Perancangan
   Pada bab ini dibahas perancangan pengembangan metode yang akan diimplementasikan.
4. Bab IV Implementasi
   Pada bab ini dibahas implementasi rancangan metode yang dibuat pada tahap perancangan.
5. Bab V Uji Coba dan Evaluasi
   Pada bab ini dibahas tahap-tahap uji coba. Setelah hasil uji coba didapatkan, maka akan dilakukan evaluasi pada nilai PSNR dan kapasitas dari metode steganografi yang diusulkan.
6. Bab VI Kesimpulan dan Saran
   Pada bab ini ditulis kesimpulan hasil uji coba yang telah dilakukan. Kesimpulan yang diambil berdasar pada rumusan masalah yang telah ditentukan. Pada bagian ini juga ditulis mengenai saran pengembangan yang dapat dilakukan.

BAB II
TINJAUAN PUSTAKA

Pada bab ini dijelaskan dasar teori yang digunakan dalam Tugas Akhir. Dasar teori meliputi metode terkait yang telah dikembangkan sebelumnya, alat yang digunakan, dan definisi data dan isitilah yang digunakan dalam buku ini.

2.1. Citra Digital

Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun komplek. Nilai-nilai tersebut direpresentasikan dengan deretan bit tertentu. Sebuah citra dapat direpresentasikan dengan ukuran *M* baris dan *N* kolom. Citra didefinisikan sebagai fungsi *f(x,y)*, dimana *x* dan *y* merupakan koordinat spasial, dan amplitudo *f* di titik koordinat (*x,y*) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Sebuah citra dapat dikatakan sebagai sebuah citra digital apabila nilai *x*, *y*, dan nilai tingkat keabuan citra secara keseluruhan berhingga (*finite*) dan bernilai diskrit. Gambar 2-1 merupakan penjelasan tentang koordinat citra digital [6].



Gambar 2-1 Koordinat Citra Digital

Sebuah citra digital dapat direpresentasikan ke dalam bentuk matrik. Matrik tersebut dapat ditulis seperti rumus 2.1.

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{matrix} \quad (2.1)$$

Nilai pada posisi *x,y* yang merupakan irisan antara baris dan kolom disebut dengan *picture elements*, *image elements*, *pels*, atau piksel.

2.1.1. Citra *Grayscale*

Citra *grayscale* merupakan citra dimana nilai bagian RED = GREEN = BLUE. Dengan kata lain citra *grayscale* merupakan citra digital yang hanya memiliki satu kanal pada setiap pikselnya. Warna yang dimiliki antara lain warna hitam, keabuan, dan putih. Warna keabuan yang dimaksud dalam citra *grayscale* merupakan warna diantara warna putih menuju warna hitam [6]. Salah satu jenis citra *grayscale* yaitu citra *grayscale* dengan kedalaman warna 8 bit dimana terdapat 256 kombinasi warna keabuan, mulai dari 0 hingga 255. Nilai piksel 0 menunjukkan warna putih sedangkan nilai piksel 255 menunjukkan warna hitam.

2.1.2. Piksel Tetangga

Sebuah piksel *p* pada koordinat (*x,y*) memiliki empat tentangga secara horizontal dan vertikal, dimana koordinat masing-masing tetangganya yaitu

$$(x+1,y), (x-1,y), (x,y+1), (x,y-1),$$

empat piksel tersebut merupakan 4-*neighbors* dari *p*, yang dinotasikan dengan N_4(p). Setiap piksel tetangga memiliki jarak 1 piksel dari koordinat (*x,y*). Lokasi piksel tetangga dari *p* berada di luar citra digital jika (*x,y*) berada pada tepi citra [7].

Empat buah tetangga diagonal dari *p* memiliki koordinat

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

dan dinotasikan dengan $N_D(p)$. Titik-titik ini, bersama dengan 4-*neighbors* disebut dengan 8-*neighbors* dari *p*, dinotasikan dengan $N_8(p)$. Seperti telah dijelaskan sebelumnya, titik dari $N_D(p)$ dan $N_8(p)$ berada di luar citra jika *(x,y)* berada pada tepi citra [7].

2.1.3. *Entropy*

Rata – rata informasi yang didapatkan dari seluruh piksel disebut dengan *entropy* dan didefinisikan seperti pada rumus (2.2).

$$Entropy,\ H = -\sum_{k=0}^{L-1} P_k log_2 P_k\ bit/pesan$$
$$= \sum_{k=0}^{L-1} \frac{P_k}{log_2 P_k} \qquad (2.2)$$

Pada sebuah citra digital, *entropy* dari citra tersebut dapat dilihat dari histrogramnya. *Entropy* dari sebuah citra bernilai maksimum ketika piksel terdistribusi secara rata pada citra tersebut [8].

2.1.4. Tingkat Kecerahan

Setiap piksel pada citra *grayscale* memiliki format 8 bit dengan nilai antara 0 hingga 255. Nilai 0 menunjukkan bahwa piksel tersebut memiliki tingkat kegelapan maksimum. Sedangkan nilai 255 menunjukkan nilai tingkat kecerahan maksimum. Nilai-nilai yang lain diantara 0 dan 255 menunjukan tingkat keabuan dari piksel tersebut.

Untuk mengestimasi tingkat kecerahan atau *brightness* dari suatu blok pada citra digital, dalam contoh ini yaitu blok berukuran 3×3 piksel, maka dapat dilakukan dengan menghitung rata-rata nilai piksel, *Bm*, seperti pada rumus (2.3) [5] :

$$Bm = \frac{\sum_i pi}{9} \text{ , } 0 \leq i \leq 8 \qquad (2.3)$$

2.2. Citra Medis

Citra medis merupakan gambar bagian tubuh manusia yang digunakan dalam bidang kedokteran. Selain digunakan dalam ilmu kedokteran, citra medis juga digunakan untuk tujuan klinis. Teknik yang digunakan dalam mengambil citra medis pun bermacam-macam seiring dengan berkembangnya teknologi kedokteran. Citra medis digunakan sebagai alat bantu untuk menganalisis kondisi pasien. Hasil analisis citra medis tersebut kemudian dapat digunakan sebagai acuan dalam mendiagnosis dan menangani penyakit pasien.

Salah satu contoh citra medis dapat dilihat pada Gambar 2-2. Citra medis yang ditampilkan pada Gambar 2-2 merupakan sebuah citra medis yang didapatkan dengan teknologi radiografi. Radiografi sendiri merupakan penggunaan sinar pengionan (sinar X, sinar gama) untuk membentuk bayangan benda yang dikaji pada film.

Gambar 2-2 Radiograf Dada

2.3. Steganografi

Steganografi adalah seni dalam melakukan komunikasi secara tersembunyi. Dalam steganografi, biasanya dipilih sebuah media yang tidak terlalu mencolok sebagai tempat untuk menyembunyikan informasi. Di masa lalu, orang-orang menggunakan tato tersembunyi atau tinta tak terlihat untuk mengirimkan konten steganografi. Saat ini, teknologi jaringan komputer menyediakan jalur komunikasi yang mudah digunakan untuk menerapkan steganografi [9].

Pada dasarnya, proses penyembunyian informasi di dalam steganografi memanfaatkan keberadaan bit redundan yang terletak pada medium steganografi. Bit-bit redundan tersebut kemudian diganti dengan informasi yang akan disembunyikan.

Steganografi modern lebih menekankan bagaimana cara agar penyembunyian data tidak terdeteksi. Salah satu kelemahan steganografi yaitu masih meninggalkan jejak perubahan data pada medium yang digunakan. Bahkan jika pesan rahasia tidak terungkap, perubahan dapat dilihat pada sifat statistik medium

yang digunakan, sehingga penyadap dapat mendeteksi distorsi dalam sifat statistik dari media steganografi yang dihasilkan. Proses menemukan distorsi ini disebut statistik *steganalysis* [9].

2.4. *Difference Expansion*

Metode DE (*difference expansion*) merupakan metode steganografi yang berdasarkan pada transformasi *reversible* pada bilangan asli. Pertama hitung nilai rata-rata dan selisih dari sepasang piksel. Untuk sepasang piksel citra *grayscale* 8 bit (*x,y)*, dimana $x, y \in Z$, $0 \leq x, y \leq 255$, maka nilai rata-rata *m* dan selisihnya *d* dapat dihitung dengan rumus 2.4 [2]

$$m = \left\lfloor \frac{x+y}{2} \right\rfloor, \qquad d = x - y, \tag{2.4}$$

dimana $\lfloor z \rfloor$ merupakan operasi pembulatan ke bawah. Rumus yang digunakan untuk mendapatkan kembali nilai *x* dan *y* ketika proses rekonstruksi dijelaskan pada rumus 2.5 [2]:

$$x = m + \left\lfloor \frac{d+1}{2} \right\rfloor, \qquad y = m - \left\lfloor \frac{d}{2} \right\rfloor. \tag{2.5}$$

Penyisipan dengan metode *difference expansion* didefinisakan pada rumus 2.6 :

$$d' = 2 \times d + b, \tag{2.6}$$

Dimana d' merupakan hasil penyisipan dan b merupakan bit yang disisipkan. Nilai b antara 1 dan 0. Dalam proses penyisipan bit, terkadang nilai baru dari x dan y melewati batas nilai normal. Nilai x atau y disebut *overflow* jika bernilai lebih dari 255, dan disebut *underflow* jika nilai x atau y dibawah 0. Untuk mencegah *overvlow* dan *underflow*, maka d' harus memenuhi persamaan 2.7 [3]:

$$\begin{cases} |d'| \leq 2 \times (255 - m) & jika\ 128 \leq m \leq 255 \\ |d'| \leq m + 1 & jika\ 0 \leq m \leq 127 \end{cases} \quad (2.7)$$

Jika nilai d' memenuhi persamaan 2.7, maka d dapat diekspansi, namun jika tidak maka d tidak dapat diekspansi. Selisih yang dapat diekspansi mampu disisipkan sebuah bit [2].

Sebagai contoh, terdapat sepasang piksel x dan y. Nilai x dan y secara berurutan yaitu $x = 205$ dan $y = 201$. Berikut merupakan langkah-langkah menyisipkan sebuah bit bernilai 1 pada sepasang piksel.

1. Hitung nilai rata-rata m dan selisih d

$$m = \left\lfloor \frac{x + y}{2} \right\rfloor = \left\lfloor \frac{205 + 201}{2} \right\rfloor = \left\lfloor \frac{406}{2} \right\rfloor = 203, \quad (2.8)$$

dan

$$d = x - y = 205 - 201 = 4 \quad (2.9)$$

2. Sisipkan nilai bit ke LSB dari nilai selisih dan buat nilai selisih baru *d'*: *d'* = $100b_2 = 1001_2 = 9$,.

$$d' = 2 \times d + b = 2 \times 4 + 1 = 9 \quad (2.10)$$

3. Hitung nilai piksel baru (x', y') dengan menggunakan nilai selisih yang baru d' dan nilai rata-rata awal m seperti persamaan 2.11 dan 2.12.

$$x' = m + \left\lfloor \frac{d' + 1}{2} \right\rfloor = 203 + \left\lfloor \frac{9 + 1}{2} \right\rfloor \quad (2.11)$$
$$x' = 203 + 5 = 208$$

$$y' = m - \left\lfloor \frac{d}{2} \right\rfloor = 203 - \left\lfloor \frac{9}{2} \right\rfloor \quad (2.12)$$
$$y' = 203 - 4 = 199$$

Proses ekstraksi data tersembunyi d hampir sama dengan proses penyisipannya. Langkah-langkahnya yaitu sebagai berikut.

1. Hitung nilai rata-rata m' dan nilai selisih d' dari nilai piksel yang baru.

$$m' = \left\lfloor \frac{x' + y'}{2} \right\rfloor = \left\lfloor \frac{208 + 199}{2} \right\rfloor,$$
$$m' = \left\lfloor \frac{407}{2} \right\rfloor = 203, \tag{2.13}$$

dan

$$d' = x' - y' = 208 - 199 = 9. \tag{2.14}$$

2. Lakukan ekstraksi pada LSB dari d' untuk mendapatkan b ($b = 1$) dan h (h=1001_2=9) dengan persamaan berikut:

$$b = LSB(d') = LSB(9) = 1, \tag{2.15}$$

$$d = \left\lfloor \frac{d'}{2} \right\rfloor = \left\lfloor \frac{9}{2} \right\rfloor = 4 \tag{2.16}$$

dimana $LSB(.)$ menunjukkan fungsi ekstraksi LSB.

3. Hitung nilai asli pasangan piksel (x, y) dengan melakukan perhitungan pada nilai rata-rata m' dan nilai selisih piksel.

$$x = m' + \left\lfloor \frac{d + 1}{2} \right\rfloor = 203 + \left\lfloor \frac{4 + 1}{2} \right\rfloor$$
$$x = 203 + 2 = 205 \tag{2.17}$$

$$y = m' - \left\lfloor \frac{d}{2} \right\rfloor = 203 - \left\lfloor \frac{4}{2} \right\rfloor \tag{2.18}$$

$$y' = 203 - 2 = 201$$

2.5. RDE (*Reduced Difference Expansion)*

Metode steganografi yang diajukan Lou dkk. [3] menggunakan sebuah fungsi transformasi untuk mereduksi nilai selisih. Metode yang diajukan yaitu *Reduced Difference Expansion* (RDE) dimana selisih d' merupakan nilai selisih yang dapat direpresentasikan pada rumus [3]:

$$d' = \begin{cases} d, & \text{jika } d' < 2 \\ d - 2^{\lfloor log_2 d \rfloor - 1}, & \text{jika } d \geq 2 \end{cases} \quad (2.19)$$

Untuk mendapatkan kembali nilai selisih awal, maka dibuat sebuah *location map* ketika melakukan reduksi terhadap nilai selisih awal. Jumlah *location map* sama dengan jumlah pasanagan piksel yang ada. Ketika nilai selisih d 0 atau 1, nilai piksel tidak dirubah dan *location map* bernilai 0. Ketika $d = 2$ dan $d' = 1$, *location map* bernilai 1. Nilai dari *location map* ditentukan menggunakan rumus [3]:

$$location\ map = \begin{cases} 0, & \text{jika } 2^{\lfloor log_2 d' \rfloor} = 2^{\lfloor log_2 d \rfloor} \text{ atau } d' = d \\ 1, & \text{jika } 2^{\lfloor log_2 d' \rfloor} \neq 2^{\lfloor log_2 d \rfloor} \end{cases} \quad (2.20)$$

Sebagai contoh, terdapat sepasang piksel dengan nilai $x = 205$ dan $y = 201$. Nilai selisih piksel $d = 4$, dan nilai rata-rata $m = 203$. Nilai d' dapat dihitung dengan rumus 2.21.

$$d' = d - 2^{\lfloor log_2 d \rfloor - 1} = 4 - 2^{\lfloor log_2 4 \rfloor - 1} \quad (2.21)$$

$$d' = 4 - 2^{2-1} = 4 - 2 = 2$$

Kemudian kita lakukan penyisipan bit $b = 1$ sehingga $d'' = 2 \times d' + b = 2 \times 2 + 1 = 5$ dan $location\ map = 1$.

$$x' = m + \left\lfloor \frac{d'' + 1}{2} \right\rfloor = 203 + \left\lfloor \frac{5 + 1}{2} \right\rfloor \quad (2.22)$$
$$x' = 203 + 3 = 206$$

$$y' = m - \left\lfloor \frac{d''}{2} \right\rfloor = 203 - \left\lfloor \frac{5}{2} \right\rfloor \quad (2.23)$$
$$y' = 203 - 2 = 201$$

Untuk mendapatkan kembali data yang disisipkan, digunakan rumus [3] :

$$d = \begin{cases} d' + 2^{\lfloor log_2 d' \rfloor - 1}, & \text{jika } location\ map = 0 \\ d' + 2^{\lfloor log_2 d' \rfloor}, & \text{jika } location\ map = 1 \end{cases} \quad (2.24)$$

Jika terdapat pasangan piksel (206,201) dan nilai *location map* 1, maka untuk mendapatkan nilai piksel semula terdapat beberapa langkah yang harus dilakukan. Pertama, kita hitung rata-rata m dan selisih d''.

$$m = \left\lfloor \frac{206 + 201}{2} \right\rfloor = \left\lfloor \frac{407}{2} \right\rfloor = 203, \quad (2.25)$$

dan

$$d'' = x - y = 206 - 201 = 5. \quad (2.26)$$

Kemudian kita ekstrak bit tersembunyi $b = 1$ dan menghitung d' dengan persamaan sebagai berikut :

$$b = LSB(d'') = LSB(5) = 1, \quad (2.27)$$

$$d' = \left\lfloor \frac{d''}{2} \right\rfloor = \left\lfloor \frac{5}{2} \right\rfloor = 2 \tag{2.28}$$

Untuk mendapatkan nilai selisih awal d maka digunakan persamaan 2.29,

$$\begin{aligned} d &= d' + 2^{\lfloor log_2 d' \rfloor} = 2 + 2^{\lfloor log_2 2 \rfloor} \\ d &= 2 + 2^1 = 2 + 2 = 4 \end{aligned} \tag{2.29}$$

Namun jika $d' = [0,1]$ dan nilai $location\ map = 1$, maka nilai $d = d'$. Terakhir, nilai rata-rata m dan nilai selisih d kita gunakan untuk mendapatkan nilai awal piksel.

$$\begin{aligned} x &= m + \left\lfloor \frac{d+1}{2} \right\rfloor = 203 + \left\lfloor \frac{4+1}{2} \right\rfloor \\ x &= 203 + 2 = 205 \end{aligned} \tag{2.30}$$

$$\begin{aligned} y &= m' - \left\lfloor \frac{d}{2} \right\rfloor = 203 - \left\lfloor \frac{4}{2} \right\rfloor \\ y' &= 203 - 2 = 201 \end{aligned} \tag{2.31}$$

2.5.1. RDE dengan Selisih Piksel Negatif

Lou dkk. [3] mengajukan metode reduksi selisih yang hanya dijalankan ketika nilai selisih piksel lebih dari 2. Itu berarti, nilai selisih selain 2 tidak akan direduksi. Untuk mengatasi hal tersebut, perlu dibuat modifikasi pada metode *Reduced Difference Expansion* agar tetap bisa mereduksi nilai selisih dibawah 2.

Proses reduksi dilakukan baik pada selisih piksel yang bernilai positif maupun negatif kecuali $-1 \leq v_n \leq 1$. Pengecualian ini dilakukan karena jika dilakukan reduksi pada rentang tersebut maka akan terjadi kesalahan rekonstruksi baik pada gambar maupun pesan rahasia yang dihasilkan. Proses

reduksi menggunakan dua metode. Metode pertama diterapkan untuk $v_n > 1$, dimana metode ini sama dengan metode oleh Lou dkk. [3] seperti yang ditunjukkan pada (2.32). Metode kedua digunakan untuk untuk $v_n < -1$ seperti terlihat pada rumus (2.33) [1].

$$\overline{v_n} = v_n - 2^{\lfloor log_2(v_n) \rfloor} \quad (2.32)$$

$$\overline{v_n} = v_n + 2^{\lfloor log_2(|v_n|) \rfloor} \quad (2.33)$$

Pada kondisi $\overline{v_n} \leq -1$ dan *location map* 0, maka metode yang digunakan yaitu (2.34), sedangkan pada kondisi $\overline{v_n} \leq -1$ dan *location map* 1, maka metode yang digunakan yaitu (2.35)

$$v_n = \overline{v_n} - 2^{\lfloor log_2 |\overline{v_n}| \rfloor - 1} \quad (2.34)$$

$$v_n = \overline{v_n} - 2^{\lfloor log_2 |\overline{v_n}| \rfloor} \quad (2.35)$$

2.5.2. Skema Penyisipan Data Multi Lapis

Baik metode *difference expansion* ataupun *reduced difference expansion* dapat diaplikasikan lebih dari sekali pada sebuah citra dalam skema penyisipan multi lapis. Oleh karena operasi *difference expansion* meningkatkan nilai selisih dari piksel-piksel yang bertetangga, maka ketika penyisipan multi lapis dilakukan, kualitas dari citra hasil penyisipan akan turun drastis. Masalah lain yang muncul ketika menggunakan metode *difference expansion* ialah semakin besar kemungkinan terjadinya *underflow* dan *overvlow* setelah proses penyisipan [3]. Berbeda halnya dengan penyisipan multi lapis menggunakan RDE. Kualitas citra hasil penyisipan dapat dipertahankan.

Berdasarkan perbandingan metode DE dan RDE, metode RDE dapat mengurangi nilai *difference expansion* dan pada akhirnya dapat meningkatkan kualitas citra. RDE dapat mengontrol jumlah pasangan piksel yang mengalami *underflow* ataupun *overflow* menjadi lebih sedikit.

Mekanise penyisipan multi lapis dengan RDE yaitu sebagai berikut [3]. Pada lapis pertama, penyisiran dilakukan secara horizontal untuk mendapatkan pasangan piksel yang ada. Ketika penyisiran secara horizontal telah selesai dilakukan, maka dilanjutkan pada lapis kedua, namun penyisiran dilakukan secara vertikal seperti ditunjukkan pada Gambar 2-3 dan Gambar 2-4. Kemudian dilanjutkan dengan penyisiran secara horizontal pada lapis ketiga, penyisiran vertikal pada lapis keempat dan seterusnya hingga mencapai *k* lapis. Urutan penyisiran dapat dilihat pada Gambar 2-5 dan Gambar 2-6.

324	324	317	324
315	324	334	315
312	318	318	327
330	322	322	325

Gambar 2-3 Lapis Pertama Dengan Penyisiran Secara Horizontal

324	324	317	324
315	324	334	315
312	318	318	327
330	322	322	325

Gambar 2-4 Lapis Kedua dengan Penyisiran Secara Vertikal

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Gambar 2-5 Urutan Penyisiran Secara Horizontal

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Gambar 2-6 Urutan Penyisiran Secara Vertikal

Proses ekstraksi pada penyisipan multi lapis memanfaatkan sifat *reversible* dari metode RDE untuk mendapatkan data rahasia dan mengembalikan nilai piksel ke nilai semula. Sebagai contoh, ketika data pada layer kelima berhasil diekstrak, maka data pada layer keempat juga dapat diekstrak dengan bantuan *location map*. Begitu pula pada layer ketiga, kedua dan layer terakhir. Jika proses ekstraksi pada layer terakhir berhasil dilakukan, maka citra awal dapat dikembalikan. Begitu pula pesan rahasia berhasil didapatkan dengan mengkobinasikan pesan rahasia yang didapat pada tiap lapis [3].

2.6. Logika *Fuzzy*

Fuzzy secara bahasa diartikan sebagai kabur atau samar samar. Berbeda dengan himpunan tegas yang hanya mengenal nilai 1 atau 0 (ya atau tidak), *fuzzy* mengenal derajat keanggotaan yang memiliki rentang nilai 0 (nol) hingga 1 (satu). Dalam teori logika *fuzzy*, suatu nilai dapat bernilai benar atau salah sekaligus. Namun seberapa besar kebenaran dan kesalahannya tergantung pada bobot keanggotaan yang dimiliki [10].

Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (linguistik), misalkan tingkat intensitas cahaya yang direpresentasikan dengan gelap, agak gelap, terang dan sangat terang. Logika *fuzzy* menunjukan seberapa tingkat kebenaran dan tingkat kesalahan suatu nilai. Tidak seperti logika *crisp*, suatu nilai hanya mempunyai 2 kemungkinan yaitu merupakan suatu anggota himpunan atau tidak [11].

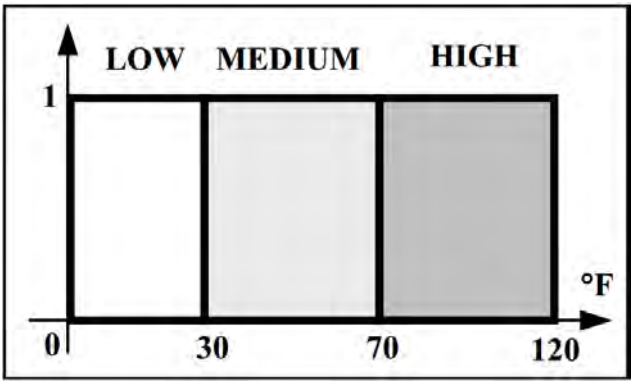
Fuzzy dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh karena itu, dalam *fuzzy* suatu hal dapat dikatan benar dan salah secara bersamaan [11]. Logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, seperti tingkat keabuan antara hitam dan putih, dan dalam bentuk konsep yang tidak pasti seperti "sedikit", "sedang" dan "sangat" [10]. Dengan menggunakan teori logika *fuzzy* dapat dilakukan penalaran secara bahasa (*linguistic reasoning*). Sehingga dalam perancangannya tidak melibatkan persamaan matematik dari objek yang akan dikendalikan.

2.6.1. Himpunan *Fuzzy*

Konsep himpunan *fuzzy* merupakan perluasan dari konsep himpunan tegas. Himpunan tegas hanya menerjemahkan derajat keanggotaan yang terbatas seperti ‘0’ atau ‘1’ atau sebuah rentang data dengan derajat keanggotaan yang terbatas. Misal, jika suhu yang tinggi didefinisikan sebagai suhu yang lebih dari 80 ^{o}F, maka suhu 70 ^{o}F tidak termasuk di dalamnya. Tetapi himpunan *fuzzy* mengatur rentang yang lebih luas untuk mendefinisikan suhu yang

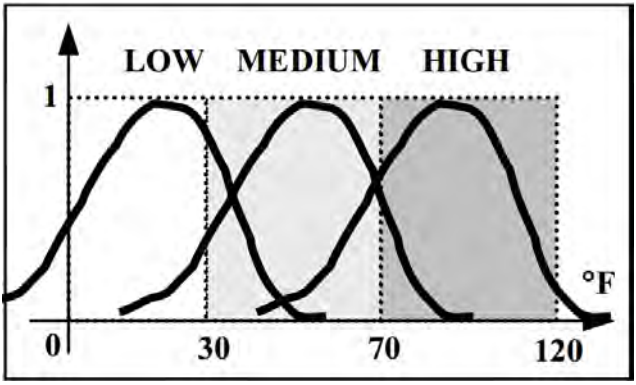
tinggi. Derajat keanggotaan yang tepat dimana 0 ^{o}F dapat termasuk dalam suhu yang tinggi tergantung pada fungsi keanggotaan *fuzzy*.

Dibanding dengan himpunan tegas, sebuah himpunan *fuzzy* memiliki batasan yang lebih kabur. Sebuah himpunan *fuzzy* memperbolehkan sebuah anggota himpunannnya termasuk ke dalam sebuah himpunan dengan tingkat derajat tertentu. Sebagai contoh, suhu dapat dibagi menjadi 3 kategori, rendah (0 ~ 30 ^{o}F), sedang (30 ~ 70 ^{o}F) dan tinggi (70 ~ 120 ^{o}F). Ketiga kategori tersebut dibagi berdasarkan sudut pandang himpunan tegas, seperti yang terlihat pada Gambar 2-7 [12].



Gambar 2-7 Himpunan Tegas [12]

Pada himpunan tegas, setiap suhu hanya dapat dikategorikan pada satu himpunan bagian saja, antara rendah, sedang ataupun tinggi dengan batas yang jelas. Namun berbeda halnya dengan himpunan *fuzzy*. Himpunan *fuzzy* seperti pada Gambar 2-8**,** batas antara himpunan bagian samar-samar atau kabur. Suatu nilai suhu dapat dikategorikan ke dalam 2 himpunan bagian, bahkan 3 himpunan bagian.



Gambar 2-8 Himpunan *Fuzzy* [12]

Ada beberapa hal yang perlu diketahui dalam memahami system *fuzzy* [11], yaitu:

1. Variabel *Fuzzy*
   Variabel *fuzzy* merupakan variable yang dibahas dalam sistem *fuzzy*. Contoh : umur, temperature.
2. Himpunan *Fuzzy*
   Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi tertentu pada suatu variable *fuzzy*.
3. Semesta Pembicaraan
   Merupakan keseluruhan nilai yang boleh digunakan pada operasi variable *fuzzy*.
4. Domain
   Domain merupakan keseluruhan nilai yang diperbolehkan pada semesta pembicaraan dan yang diperbolehkan untuk dioperasikan.

2.6.2. Fuzzifikasi dan Fungsi Keanggotaan *Fuzzy*

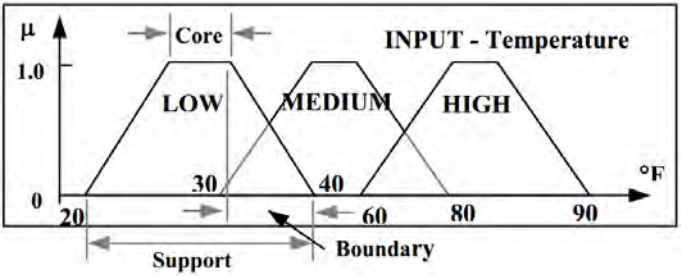
Himpunan *fuzzy* tidak akan berguna jika hanya sebatas berupa himpunan saja. Namun perlu adanya proses pengambilan keputusan pada *fuzzy*. Untuk menerapkan proses pengambilan keputusan pada *fuzzy* diperlukan beberapa tahap yaitu: fuzzifikasi, inferensi *fuzzy* dan *defuzzifikasi* [12].

Fuzzifikasi merupakan tahap pertama untuk menerapkan sebuah sistem inferensi *fuzzy*. Kebanyakan variabel di dunia nyata merupakan data yang bersifat *crisp*. Oleh karena itu, variabel yang bersifat *crisp* tersebut (baik variabel *input* dan *output*) dirubah menjadi variabel *fuzzy*. Kemudian, pada akhir proses, hasil *fuzzy* harus dikonversi lagi ke dalam variabel *crisp*.

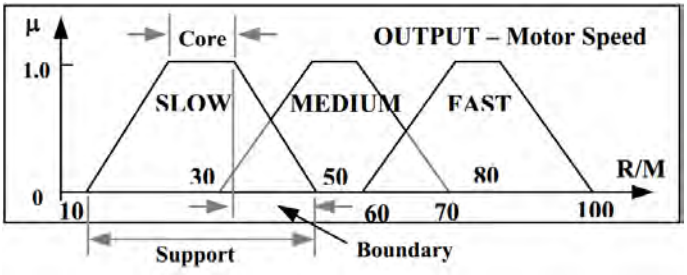
Secara umum, fuzzifikasi melibatkan dua proses: membuat fungsi keanggotaan untuk variabel *input* dan *output*, kemudian merepresentasikannya dalam bahasa lunguistik.

Pada penerapannya, fungsi keanggotaan *fuzzy* memiliki banyak jenis, seperti kurva segitiga, trapesium, *Gaussian*, *bell-shaped*, *sigmoidal* dan *S-curve*. Untuk sistem-sistem yang membutuhkan variasi dinamis signifikan dalam periode waktu yang singkat, maka dapat menggunakan kurva segitiga ataupun trapesium. Namun bagi sistem yang membutuhkan kontrol akurasi yang sangat tinggi dapat menggunakan kurva *Gaussian* atau *S-curve* [12].

Gambar 2-9 menunjukkan fungsi keanggotaan *input* yang telah dikonversi ke dalam bahasa linguistik. Begitu pula pada Gambar 2-10 merupakan fungsi keanggotaan *output* yang sudah dikonversi ke dalam bahasa linguistik yaitu *slow*, *medium* dan *fast*.



Gambar 2-9 Fungsi Keanggotaan Masukan [12]



Gambar 2-10 Fungsi Keanggotaan Hasil [12]

2.6.3. Aturan Kontrol *Fuzzy*

Sebuah aturan *fuzzy* merepresentasikan sebuah urutan dalam bentuk IF-THEN, menjelaskan tindakan atau hasil apa yang seharusnya diambil berdasarkan pada informasi yang telah diamati.

2.6.3.1. Aturan Pemetaan *Fuzzy*

Aturan pemetaan *fuzzy* merupakan pemetaan fungsional antara *input* dan *output* menggunakan variabel linguistik. Dasar dari sebuah aturan pemetaan *fuzzy* adalah sebuah tabel atau grafik *fuzzy* yang menjelaskan hubungan antara *input* dan output.

Dalam aplikasi nyata, variabel *input* biasanya lebih dari satu. Sebagai contoh, sebuah sistem pengatur suhu ruangan, variabel *input*-nya meliputi suhu saat ini dan laju perubahan suhu. Tabel 2-1 menunjukkan contoh aturan *fuzzy* yang diterapkan pada pengatur suhu rangan [12].

Tabel 2-1 Contoh Aturan-Aturan Fuzzy [12]

Ṫ \ T	RENDAH	SEDANG	TINGGI
RENDAH	CEPAT	SEDANG	SDANG
SEDANG	CEPAT	LAMBAT	LAMBAT
TINGGI	SEDANG	LAMBAT	LAMBAT

Baris dan kolom menunjukkan dua buah *input*, suhu dan laju perubahan suhu. Kedua *input* tersebut berhubungan dengan bagian IF dalam aturan IF-THEN. Kesimpulan atau kontrol *output* terletak pada titik persilangan dari tiap baris (suhu) dan tiap kolom (laju perubahan suhu), dan kesimpulan tersebut berhubngan dengan bagian THEN pada aturan IF-THEN.

2.6.3.2. Aturan Implikasi *Fuzzy*

Sebuah aturan implikasi *fuzzy* menjelaskan sebuah logika hubungan implikasi antara *input* dan *output*. Sebagai contoh

IF suhu RENDAH, THEN penghangat ruangan seharusnya CEPAT.

2.6.4. Defuzzifikasi

Kesimpulan atau kontrol *output* didapatkan dari kombinasi *input*, fungsi keanggotaan *output* dan aturan *fuzzy*. Kesimpulan atau kontrol *output* masih merupakan elemen *fuzzy* yang samar. Untuk membuat kesimpulan atau hasil *fuzzy* yang dapat digunakan pada aplikasi nyata, maka perlu adanya proses defuzzifikasi. Proses defuzzifikasi merupakan proses untuk merubah hasil *fuzzy* ke dalam hasil *crisp*. Tiga teknik defuzzifikasi yang sering digunakan antara lain : metode *Mean of Maximum*, *Center of Gravity* dan metode *Height* [12].

2.6.5. Logika *Fuzzy* Mamdani

Mamdani merupakan salah satu teknik inferensi *fuzzy* yang umum digunakan. Metode ini dikembangkan oleh Mamdani dan Assilian pada tahun 1975. Penerapan metode ini pertama kali dilakukan untuk mengontrol mesin uap. Aturan yang diterapkan saat itu didapat dari karyawan operator yang sudah berpengalaman. Di dalam metode Mamdani, proses implikasi *fuzzy* menggunakan operator *min* atau *conjuction* yang menandakan nilai minimum,

sedangkan aturan agregasi menggunakan operator *max* yang merupakan nilai maksimumnya [13].

Terdapat empat buah tahap pemprosesan variabel *input* pada metode Mamdani. Tahapan-tahapan tersebut antara lain [13]:

1. Fuzzifikasi
   Tahap pertama dalam metode Mamdani yaitu mengubah *crisp input* menjadi angka *fuzzy*, dan meletakkan pada *fuzzy set* yang benar.
2. Evaluasi Aturan
   Setelah dilakukan fuzzifikasi, maka aturan yang berlaku akan dievaluasi bersamaan dengan *fuzzy set* yang sudah didapat.
   Jika kondisinya menggunakan AND maka rumusnya :
   $$\mu_{A\cap B} = min(\mu_A[X], \mu_B[X])$$
   Dimana
   μ= *fuzzy set*
   $\mu_A[X]$ = kumpulan *fuzzy set* A dari x
   $\mu_B[X]$ = kumpulan *fuzzy set* B dari x

   Jika kondisinya menggunakan OR maka rumusnya:
   $$\mu_{A\cap B} = max(\mu_A[X], \mu_B[X])$$

   Dimana
   μ= *fuzzy set*
   $\mu_A[X]$ = kumpulan *fuzzy set* A dari x
   $\mu_B[X]$ = kumpulan *fuzzy set* B dari x

3. Agregasi Aturan
   Pada tahan ini akan dilakukan agregasi pada *output set* yang mengambil nilai maksimumnya dengan menggunakan fungsi *max*.
   $$\mu_{sf}[X_i] \leftarrow max([X_i], \mu_{kf}[X_i])$$

   Dengan μ_{sf} merupakan *fuzzy set*.

4. *Defuzzifikasi*
   Setelah dilakukan agregasi, maka selanjutnya adalah mengubah *fuzzy output* menjadi *crisp output*. Metode *defuzzifikasi* yang digunakan adalah metode *centroid* atau sering disebut dengan *Center of Gravity* (COG).

2.7. Skema Steganografi Adaptif berdasarkan *Fuzzy Inference System* (FIS)

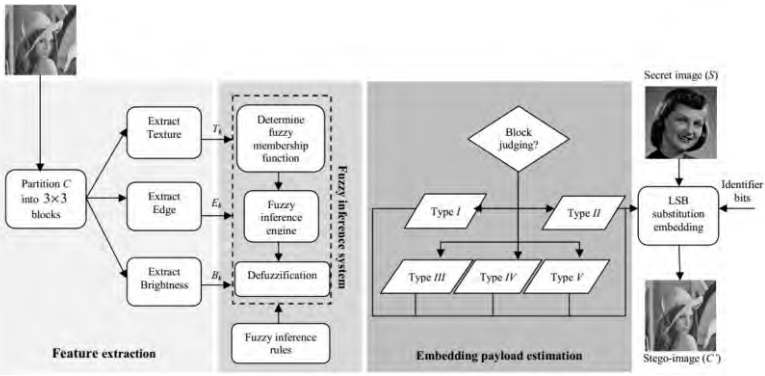
Skema steganografi adaptif [5] seperti yang diusulkan oleh Sara Sajasi dan Amir Masoud memanfaatkan *Fuzzy inference System* dalam menentukan jumlah bit yang disipkan pada sebuah blok. Sebuah blok berukuran 3×3 piksel.

Terdapat tiga tahapan dalam skema ini antara lain : ekstraksi karakteristik citra, *Fuzzy Inference System*, dan estimasi kapasitas penyisipan.

Tahap pertama yaitu menentukan karakteristik citra. Terdapat lebih dari satu karakteristik citra yang digunakan. Karakteristik citra yang digunakan antara lain tekstur, sensitivitas tepi dan tingkat kecerahan citra. Karakteristik setiap blok pada citra akan dihitung untuk digunakan sebagai masukan pada *Fuzzy Inference System* [5].

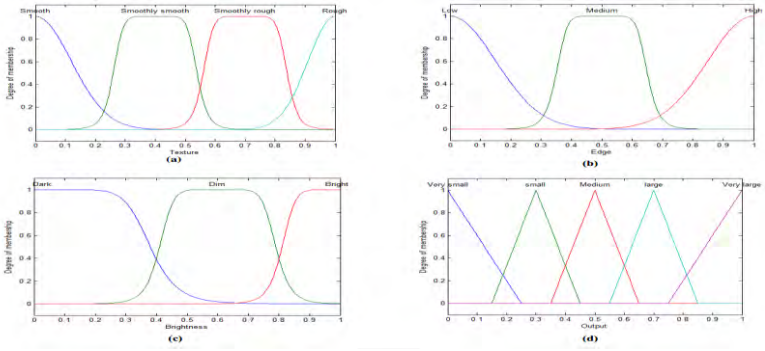
Tahap kedua yaitu merancang FIS yang akan digunakan. Pada tahap ini dibuat fungsi keanggotan *fuzzy* untuk masing-masing karakteristik citra yang digunakan. Begitu pula fungsi keanggotaan *fuzzy* untuk variabel *output* juga didefinisikan pada tahap ini [5].

Fuzzy Inference System yang dibuat akan menentukan nilai kategori. Terdapat lima nilai kategori yang digunakan. Pada tahap akhir, masing-masing kategori akan dipetakan ke jumlah bit yang akan disisipkan. Skema steganografi adaptif berdasarkan FIS secara keseluruhan dapat dilihat pada Gambar 2-11. Fungsi keanggotaan masukan dan hasil yang digunakan dapat dilihat pada Gambar 2-12.



Gambar 2-11 Diagram Prosedur Penyisipan Secara Adaptif [5]

Metode penyisipan yang digunakan oleh Sara Sajasi dan Amir Masoud adalah penyisipan LSB. Ketika suatu blok memiliki tekstur yang semakin kasar, dan nilai *entropy* yang makin besar maka jumlah bit yang dapat disisipkan per pikselnya makin banyak. Begitu pula ketika tingkat kecerahan suatu blok making besar, maka jumlah bit yang disisipkan juga semakin banyak.



Gambar 2-12 Fungsi Keanggotaan dan Kelas Semantik; (A) *Input* Karakteristik T_k, (B) *Input* Karakteristik E_k, (C) *Input* Karakteristik B_k, (D) *Output* [5]

2.8. MATLAB

MATLAB merupakan singkatan dari Matrix Laboratoris. MATLAB merupakan bahasa pemrograman komputer generasi keempat yang dikembangkan oleh The MathWorks. MATLAB memungkinkan pengguna untuk manipulasi matriks, membuat plot fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, serta sebuah antarmuka dengan program dalam bahasa lainnya [14].

2.8.1. Entropyfilt

Entropyfilt merupakan fungsi dalam MATLAB untuk menghitung *entropy* lokal dari sebuah citra *grayscale*. $J = entropyfilt(I)$ mengembalikan nilai berupa *array* J, dimana setiap *output* piksel berisi nilai *entropy* terhadap 9×9 pixel tetangga di sekitar piksel yang sedang dihitung. Untuk piksel-piksel yang berada pada tepi citra, fungsi *entropyfilt* menggunakan mekanisme *symmetric padding*. Dalam *symmetric padding*, nilai dari *padding* piksel merupakan sebuah refleksi cermin dari piksel tepi [14].

2.8.2. *Rangefilt*

Rangefilt merupakan fungsi dalam MATLAB untuk menghitung jarak lokal dari sebuah citra. $J = rangefilt(I)$ mengembalikan nilai berupa *array* J, dimana setiap *output* piksel berisi nilai jarak (nilai maksimum – nilai minimum) dari 3×3 piksel tetangga di sekitar piksel yang sedang dihitung [14].

2.8.3. *Stdfilt*

Stdfilt merupakan fungsi dalam MATLAB untuk menghitung standar deviasi lokal dari sebuah citra *grayscale*. $J = stdfilt(I)$ mengembalikan nilai berupa *array* J, dimana setiap *output pixel* berisi nilai standar deviasi terhadap 3×3 pixel tetangga di sekitar *pixel* yang sedang dihitung. Untuk piksel – piksel yang

berada pada tepi citra, fungsi *stdfilt* menggunakan mekanisme *symmetric padding*. Dalam *symmetric padding*, nilai dari *padding* piksel merupakan sebuah refleksi cermin dari piksel tepi [14].

2.8.4. *Fuzzy Logic Toolbox*

Fuzzy logic toolbox merupakan *toolbox* yang disediakan oleh MATLAB untuk merancang dan mensimulasikan sistem logika *fuzzy*. *Fuzzy logic toolbox* mengarahkan pengguna melalui langkah-langkah dalam merancang *fuzzy inference system* [14].

BAB III
PERANCANGAN

Pada bab ini dibahas mengenai deskripsi umum metode, perancangan penyisipan dan ekstraksi bit, serta perancangan pengembangan metode yang diusulkan dari sisi penentuan level penyisipan dalam penyisipan multi lapis.

3.1. Deskripsi Umum Metode

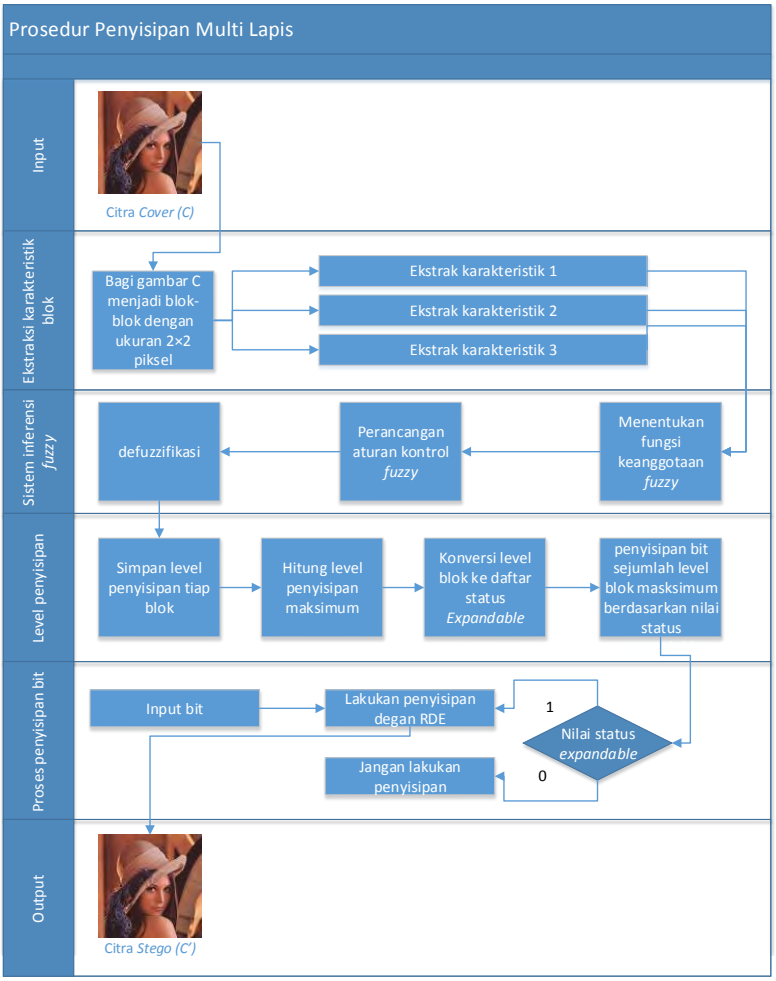
Metode pengembangan yang diusulkan terletak pada mekanisme penentuan level penyisipan. Level penyisipan ini diterapkan pada metode RDE multi lapis. Level penyisipan menentukan jumlah lapis maksimal yang boleh disisipkan pada sebuah blok. Blok yang digunakan pada penelitian ini berukuran 2×2 piksel.

Masing-masing blok memiliki level penyisipan yang berbeda-beda. Proses perhitungan level untuk setiap blok menggunakan logika *fuzzy*. Karakteristik setiap blok menjadi *input* yang akan diproses menggunakan logika *fuzzy*. Hasil akhir dari logika *fuzzy* berupa nilai level blok tersebut.

Pengembagnan metode juga terletak pada mekanisme pemilihan urutan pasangan piksel. Pemilihan urutan pasangan piksel dilakukan per blok.

3.2. Perancangan Skema Metode Keseluruhan

Tahap awal dalam skema yang diusulkan sama dengan tahap awal dalam skema steganografi adaptif berdasarkan *fuzzy inference system* (FIS) [5] yang diusulkan oleh Sara Sajasi dan Amir Masoud. Namun karakteristik blok citra yang digunakan sebagai *input fuzzy* berbeda. Perbedaan ini disebabkan oleh sifat metode penyisipan LSB dan metode penyisipan RDE yang berbeda pula. Sehingga tahap awal skema akan menghasilkan sistem inferensi *fuzzy* yang berbeda dengan yang disulkan oleh Sara Sajasi.



Gambar 3-1 Blok Diagram Prosedur Penyisipan RDE Multi Lapis Menggunakan Logika *Fuzzy*

Pada bagian penyisipan multi lapis, metode ini mengadopsi prosedur yang diusulkan oleh Lou dkk. [3]. Dalam metode yang diusulkan oleh Loud dkk, penyisiran dilakukan untuk mendapatkan

pasangan piksel yang ada pada sebuah citra. Berbeda halnya dengan metode yang penulis usulkan, penyisiran dilakukan untuk mendapatkan blok dengan ukuran 2×2 piksel yang ada pada sebuah citra. Kemudian penyisipan dilakukan sesui dengan level penyisipan tiap blok yang didapatkan menggunakan metode logika *fuzzy*. Gambar 3-1 menggambarkan prosedur penyisipan RDE multi lapis menggunakan logika *fuzzy* yang diusulkan oleh penulis.

3.3. Perancangan Penyisipan dan Ekstraksi Bit

Metode penyisipan yang digunakan adalah RDE seperti yang diusulkan oleh Lou dkk [3]. Namun metode tersebut tidak memproses pasangan piksel dengan selisih negatif. Oleh karena itu, penulis menambahkan metode yang digunakan oleh Tohari dkk [1] dalam proses penyisipan dan ekstraksi bit menggunakan RDE ketika selisih yang dihasilkan negatif.

Dengan menggunakan metode RDE, maka gambar *cover* dapat dikembalikan secara utuh setelah proses ekstraksi.

3.4. Perancangan Mekanisme Penyisipan Multi Lapis

Pada tahap perancangan mekanisme penyisipan multi lapis, dilakukan perancangan terhadap cara penyisiran blok, struktur level *map*, metode pengurutan pasangan piksel, dan perancangan *location map*.

3.4.1. Penentuan Blok

Blok yang digunakan berukuran 2×2 piksel. Gambar 3-2 menunjukkan sebuah blok yang terdiri dari piksel 1, piksel 2, piksel 3 dan piksel 4.

1	2
3	4

Gambar 3-2 Blok Piksel

Data blok didapatkan dengan cara melakukan penyisiran terhadap citra *cover*. Urutan penyisiran blok dalam sebuah citra ditunjukkan oleh Gambar 3-3. Penyisiran dilakukan secara horizontal. Nomor yang sama menunjukkan bahwa piksel tersebut termasuk dalam blok yang sama. Nomor tersebut menjadi indeks penunjuk blok yang ada.

1	1	2	2	3	3
1	1	2	2	3	3
4	4	5	5	6	6
4	4	5	5	6	6
7	7	8	8	9	9
7	7	8	8	9	9

Gambar 3-3 Urutan Penyisiran Blok

3.4.2. Struktur Level *Map*, Daftar Blok *Expandable*, dan Daftar Blok Penyisipan

Level *map* merupakan *map* yang digunakan untuk menyimpan nilai level penyisipan dari masing-masing blok. Setiap blok memiliki level penyisipannya sendiri sesuai dengan karakteristik blok tersebut. Level penyisipan menunjukkan jumlah maksimal penyisipan yang dapat dilakukan pada blok tersebut agar PSNR citra dapat dipertahankan. Level *map* berbentuk *array*, dimana kolom *array* menunjukkan indeks blok. Total blok dapat dihitung dengan cara $M \times N/4$, dimana $M \times N$ merupakan ukuran citra. Gambar 3-4 merupakan contoh level *map* dari sebuah citra dengan ukuran 4×4 piksel.

2	3	1	0

Gambar 3-4 Contoh Level *Map*

Kolom ketiga menunjukkan bahwa blok ketiga memiliki level penyisipan 1. Dari level *map* maka akan dibuat daftar blok *expandable*. Daftar blok *expandable* memperlihatkan blok mana saja yang kemungkinan dapat disisipkan pada lapis tertentu. Daftar blok *expandable* berbentuk *array* 2 dimensi, dimana kolom *array* menunjukkan indeks blok dan baris *array* menunjukkan urutan lapisan pada penyisipan multi lapis. Jumlah baris didapat dari nilai level penyisipan tertinggi. Gambar 3-5 merupakan contoh daftar blok *expandable* berdasarkan level *map* pada Gambar 3-4.

1	1	1	0
1	1	0	0
0	1	0	0

Gambar 3-5 Contoh Daftar Blok *Expandable*

Baris dua kolom dua yang bernilai 1 menunjukkan blok dengan indeks 2 diikutkan dalam proses penyisipan lapis 2. Namun baris empat kolom dua yang bernilai 0 menunjukkan blok dengan indeks 4 tidak diikutkan dalam proses penyisipan lapis 2.

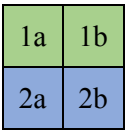
Ada kalanya sebuah blok yang dinyatakan dapat diikutkan dalam proses penyisipan mengalami kegagalan ketika proses penyisipan berlangsung. Hal ini terjadi jika penyisipan gagal dilakukan pada salah satu atau kedua pasangan piksel pada blok tersebut. Blok yang berhasil ataupun gagal pelu dicatat ulang pada daftar blok penyisipan. Struktur daftar blok penyisipan sama dengan struktur daftar blok *expandable*, namun elemen pada daftar blok penyisipan bernilai 1 jika proses penyisipan pada blok tersebut berhasil.

3.4.3. Penyisipan Bit dan Metode Urutan Pasangan Piksel

Terdapat banyak kombinasi urutan pasangan piksel yang dapat digunakan dalam penyisipan multi lapis menggunakan RDE. Pada penelitian ini, penulis menggunakan tiga jenis metode urutan

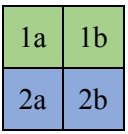
pasangan piksel antara lain urutan horizontal, horizontal vertikal, dan terdefinisi. Kombinasi urutan ini berdasarkan ukuran blok 2×2 piksel.

1. Urutan horizontal
   Pada urutan horizontal, setiap lapis penyisipan menggunakan urutan seperti pada Gambar 3-6. Pasangan piksel pertama ditunjukkan oleh piksel 1a dan piksel 1b. Pasangan piksel kedua ditunjukkan oleh piksel 2a dan piksel 2b. Masing-masing pasangan piksel dapat menampung 1 bit.

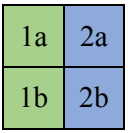


Gambar 3-6 Urutan Pasangan Piksel Horizontal

2. Urutan horizontal vertikal
   Urutan horizontal vertikal dibagi menjadi dua jenis pembagian urutan. Urutan seperti pada Gambar 3-7 diterapkan pada lapis genap, sedangkan urutan seperti pada Gambar 3-8 diterapkan pada lapis ganjil. Urutan horizontal vertikal mengadopsi cara penyisipan pasangan piksel yang diusulkan oleh Lou dkk. [3] dalam metode penyisipan multi lapis.



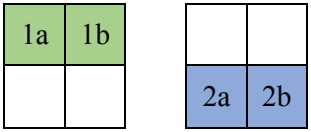
Gambar 3-7 Urutan Pasangan Piksel Horizontal Vertikal pada Lapis Genap



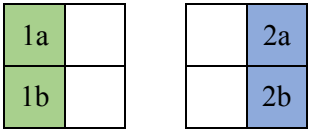
Gambar 3-8 Urutan Pasangan Piksel Horizontal Vertikal pada Lapis Ganjil

3. Urutan terdefinisi

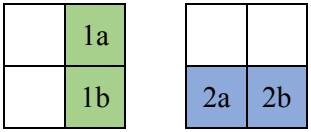
   Urutan terdifinisi merupakan cara pemilihan kombinasi pasangan piksel pada tiap lapis yang diusulkan oleh penulis. Terdapat delapan kombinasi pemilihan pasangan piksel. Tiap kombinasi diterapkan pada lapis tertentu. Penyisipan pada lapis satu hingga delapan memiliki kombinasi yang berbeda seperti ditunjukkan pada Gambar 3-9 sampai dengan Gambar 3-16. Jika lapis lebih dari delapan, maka kembali menggunakan kombinasi pertama dan seterusnya.



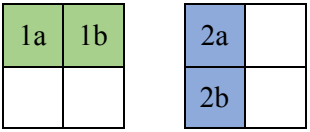
Gambar 3-9 Urutan Pasangan Piksel Terdefinisi pada Lapis 1



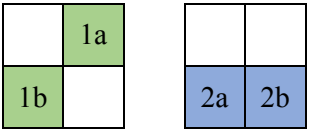
Gambar 3-10 Urutan Pasangan Piksel Terdefinisi pada Lapis 2



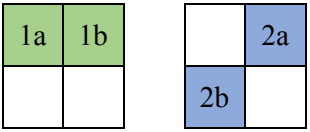
Gambar 3-11 Urutan Pasangan Piksel Terdefinisi pada Lapis 3



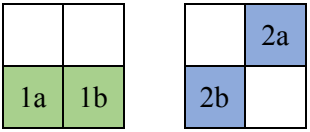
Gambar 3-12 Urutan Pasangan Piksel Terdefinisi pada Lapis 4



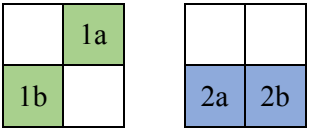
Gambar 3-13 Urutan Pasangan Piksel Terdefinisi pada Lapis 5



Gambar 3-14 Urutan Pasangan Piksel Terdefinisi pada Lapis 6



Gambar 3-15 Urutan Pasangan Piksel Terdefinisi pada Lapis 7



Gambar 3-16 Urutan Pasangan Piksel Terdefinisi pada Lapis 8

3.4.4. Struktur Location Map

Location map berupa *array* dua dimensi. Jumlah baris menunjukkan jumlah lapis penyisipan. Jumlah kolom menunjukkan jumlah pasangan piksel yang ada pada citra tersebut. Total pasangan piksel dapat dihitung dengan cara $M \times N/2$, dimana $M \times N$ merupakan ukuran citra.

3.4.5. Ekstraksi Bit Multi Lapis

Proses ekstraksi bit dimulai dari lapis terakhir atau terbesar, hingga lapis pertama. Pada masing-masing lapis, urutan pasangan piksel yang diesktrak mengikuti urutan piksel pada saat penyisipan.

Terdapat pengecualian pada jenis urutan pasangan piksel terdefinisi. Jika pada jenis urutan yang lain pasangan piksel pertama yang lebih dahulu diekstrak, maka pada jenis urutan pasangan piksel terdifinisi, pasangan piksel kedua yang lebih dahulu diekstrak. Kemudian dilanjutkan dengan ekstraksi pasangan bit pertama.

Setelah melakukan proses ekstraksi, maka semua deret bit digabungkan untuk mendapatkan pesan rahasia yang berbentuk *string*.

3.5. Perancangan Mekanisme Penentuan Level Penyisipan

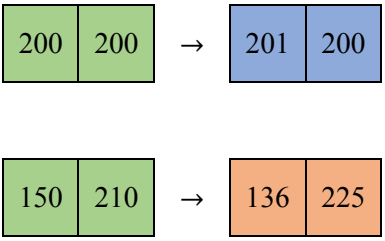
Pada proses penyisipan multi lapis, kualitas citra *stego* akan semakin rendah seiring dengan jumlah lapis yang diterapkan. Tujuan penentual level pada penyisipan multi lapis adalah menjaga kualitas citra *stego*. Metode yang digunakan untuk menentukan level penyisipan yaitu logika *fuzzy*.

3.5.1. Perancangan Variabel Fuzzy

Penentuan variabel *fuzzy* berdasarkan karakteristik metode penyisipan. Variabel *fuzzy* yang digunakan pada skema steganografi adaptif [5] seperti yang diusulkan oleh Sara Sajasi dan Amir Masoud antara lain tekstur blok, sensitivitas tepi, dan tingkat

kecerahan blok. Pemilihan variabel tersebut berdasarkan karakteristik penyisipan LSB, dimana LSB disisipkan pada bagian blok yang perbedaan warna antar pikselnya besar. Dengan kata lain, perbedaan nilai pikselnya besar.

Berbeda halnya dengan metode penyisipan RDE. Pada metode ini semakin besar perbedaan nilai antar piksel pada suatu blok, maka kualitas blok hasil penyisipan akan semakin menurun. Perbedaan hasil penyisipan dapat dilihat pada Gambar 3-17.Jika terdapat sepasang piksel x dan y, dimana $x = 200$ dan $y = 200$, maka setelah disisipkan dengan bit bernilai 1 menggunakan metode RDE, maka nilai x' dan y' yaitu x'=201 dan y'=200. Jika terdapat sepasang piksel x dan y, dimana $x = 150$ dan $y = 210$. Maka setelah disisipkan dengan bit bernilai 1 menggunakan metode RDE, maka nilai x' dan y' yaitu x'=136 dan y'=225, nilai x' dan y' yang baru berbeda 4 bit dari niai x dan y.



Gambar 3-17 Contoh Hasil Penyisipan Bit 1

Berawal dari karakteristik RDE tersebut, maka penulis mengusulkan beberapa karakteristik blok sebagai variabel *fuzzy*. Karakteristik blok yang digunakan dapat menunjukkan seberapa dekat jarak atau hubungan antar piksel dalam sebuah blok. Karakteristik yang digunakan antara lain *entropy* lokal, jarak lokal dan standar deviasi lokal dari blok tersebut. Nilai *entropy* lokal yang digunakan yaitu nilai rata-rata *entropy* lokal dari setiap piksel dalam sebuah blok. Begitu pula nilai jarak lokal dan standar deviasi lokal yang digunakan yaitu nilai rata-rata dari seluruh piksel yang ada dalam sebuah blok.

Untuk menentukan rentang nilai masing-masing variabel, maka dibuat sampel blok yang berukuran 2×2 piksel. Terdapat empat scenario pembuatan sampel. Dan masing-masing scenario diulangi dua kali. *Pseudocode* pembuatan sampel dapat dilihap pada Gambar 3-18 sampai dengan Gambar 3-21.

```
1. Ulangi hingga nilai max=255, dengan nilai max
   awal 0, dan nilai min awal 0
2.    Nilai max ditambah 5 setiap iterasi
3.    Buat blok 2×2 piksel dengan nilai piksel
      random antara min sampai max
4.    Simpan blok sebagai gambar
```

Gambar 3-18 *Pseudocode* Pembuatan Sampel Blok Skenario 1

```
1. Ulangi hingga nilai min=255, dengan nilai max
   awal 255 dan niai min awal 255
2.    Nilai min dikurangi 5 setiap iterasi
3.    Buat blok 2×2 piksel dengan nilai piksel
      random antara min sampai max
4.    Simpan blok sebagai gambar
```

Gambar 3-19 *Pseudocode* Pembuatan Sampel Blok Skenario 2

```
1. Ulangi hingga nilai max=255 dan nilai min=255,
   dengan nilai max awal 0 dan nilai min awal 0
2.    Nilai max dan min ditambah 5 setiap iterasi
3.    Buat blok 2×2 piksel dengan nilai piksel
      random antara min sampai max
4.    Simpan blok sebagai gambar
```

Gambar 3-20 *Pseudocode* Pembuatan Sampel Blok Skenario 3

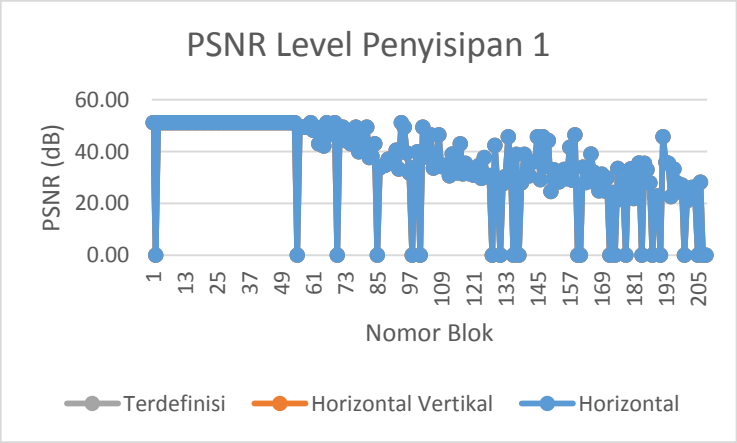
```
1. Ulangi sebanyak 51 kali, dengan nilai max awal
   125 dan nilai min awal 125
2.    Nilai max ditambah 2 dan min dikurangi 2
      setiap iterasi
3.    Buat blok 2×2 piksel dengan nilai piksel
      random antara min sampai max
4.    Simpan blok sebagai gambar
```

Gambar 3-21 *Pseudocode* Pembuatan Sampel Blok Skenario 4

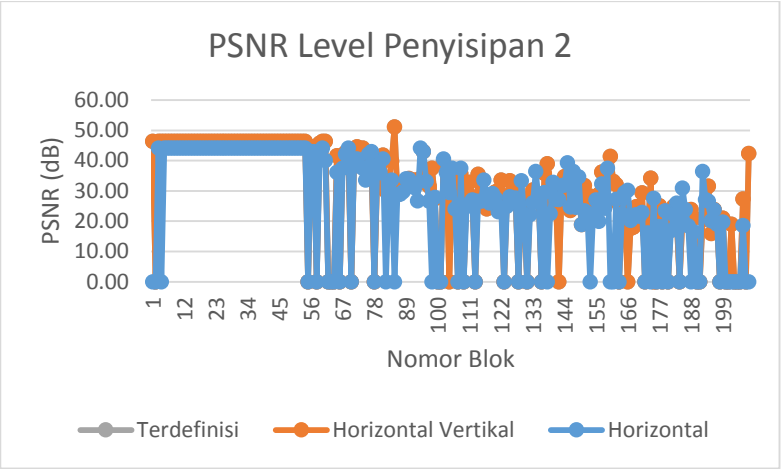
Setelah sampel terbentuk, maka dilakukan penyisipan multi lapis pada semua sampel. Penyisipan multi lapis dilakukan hingga lapis kedelapan. Dalam proses penyisipan semua pasangan piksel disisipkan dengan bit bernilai 1.

Penyisipan dilakukan dengan tiga kombinasi, yaitu kombinasi urutan pasangan piksel horizontal, horizontal vertikal, dan terdefinisi. PSNR tiap lapis akan dihitung. Selain PSNR, nilai *entropy* lokal, jarak lokal, dan standar deviasi lokal dari setiap blok juga dihitung. Nilai *entropy* lokal, jarak lokal, dan standar deviasi lokal selanjutnya disimpan dan digunakan untuk menentukan fungsi keanggotaan *fuzzy*.

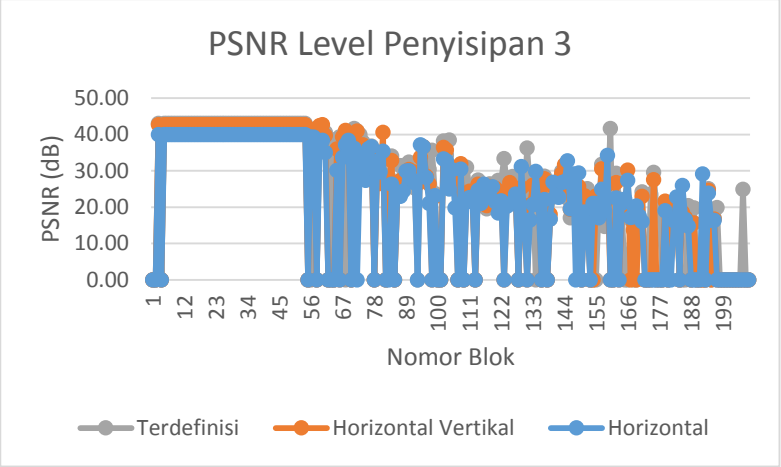
Sebelum melanjutkan ke tahap pembuatan fungsi keanggotaan *fuzzy*, penulis menyeleksi kombinasi urutan pasangan mana saja yang memiliki hasil PSNR tertinggi. Hasil PSNR dapat dilihat pada Gambar 3-22 sampai dengan Gambar 3-29, dimana data diurutkan berdasarkan nilai standar deviasi lokal.



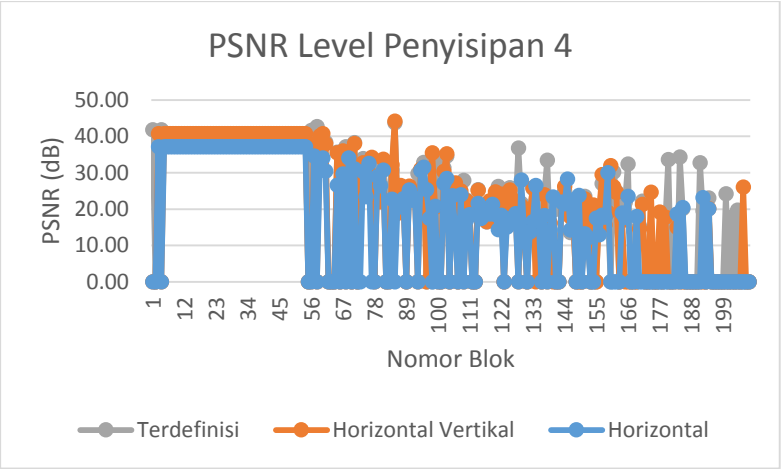
Gambar 3-22 Grafik PSNR Sampel Blok Level Penyisipan 1



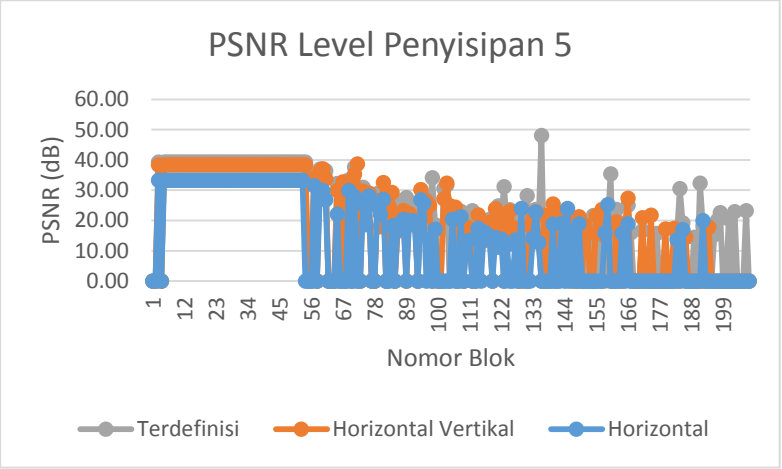
Gambar 3-23 Grafik PSNR Sampel Blok Level Penyisipan 2



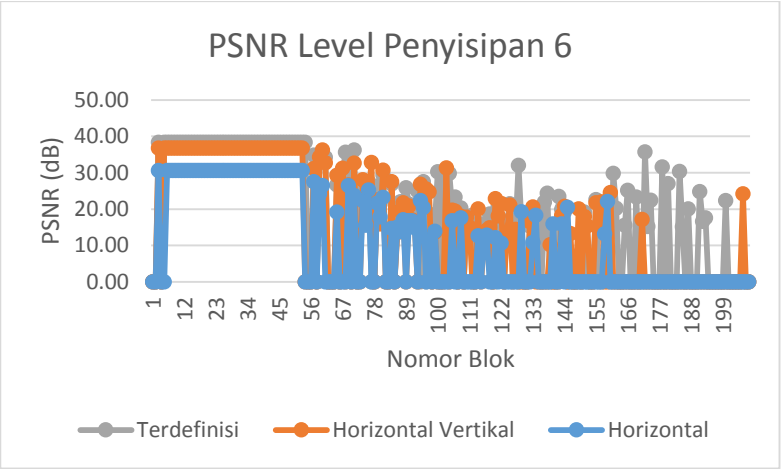
Gambar 3-24 Grafik PSNR Sampel Blok Level Penyisipan 3



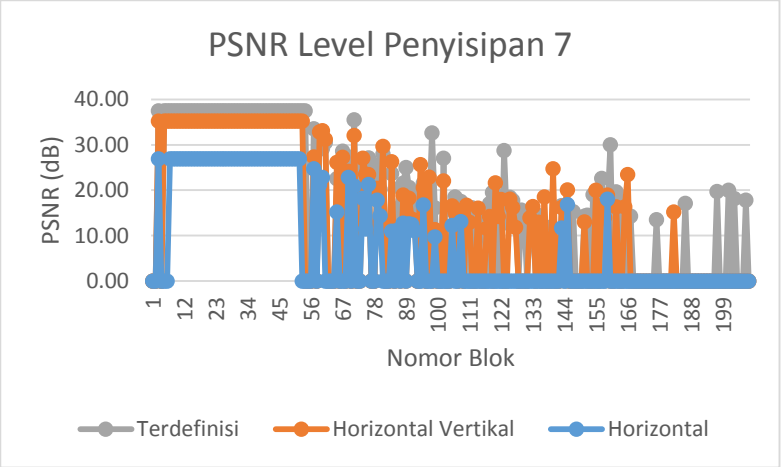
Gambar 3-25 Grafik PSNR Sampel Blok Level Penyisipan 4



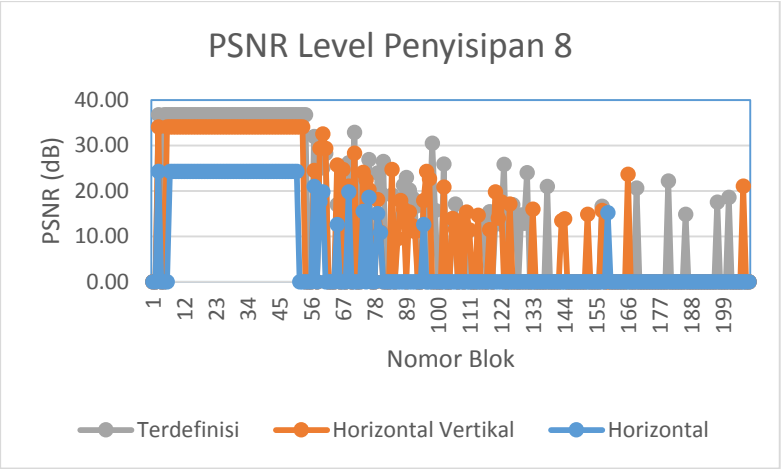
Gambar 3-26 Grafik PSNR Sampel Blok Level Penyisipan 5



Gambar 3-27 Grafik PSNR Sampel Blok Level Penyisipan 6



Gambar 3-28 Grafik PSNR Sampel Blok Level Penyisipan 7



Gambar 3-29 Grafik PSNR Sampel Blok Level Penyisipan 8

Sesuai hasil grafik PSNR sampel blok maka kombinasi urutan pasangan piksel yang digunakan yaitu horizontal vertikal, dan terdefinisi. Kombinasi urutan pasangan piksel horizontal tidak terpilih karena hasil PSNRnya secara konstan lebih kecil disbanding yang lainnya.

3.5.2. Perancangan Fuzzifikasi dan Fungsi Keanggotaan *Fuzzy*

Dari pemrosesan data hasil penyisipan pada sampel blok, maka kita dapat menentukan fungsi keanggotaan *fuzzy*. Fungsi keanggotaan *fuzzy* terdiri dari masing-masing karakteristik blok yang sudah ditentukan sebelumnya.

Penentuan keanggotaan *fuzzy* berdasarkan nilai PSNR pada tiap level penyisipan. Penulis memberi batasan nilai PSNR terendah yaitu 30. Kemudian ditentukan rentang nilai karakteristik blok (*entropy* lokal, jarak lokal, dan standar deviasi lokal) terhadap masing-masing level penyisipan agar memenuhi nilai PSNR terendah 30.

Setelah dilakukan penghitungan pada rentang yang ada, penulis menentukan semesta pembicaraan dan domain dari masing-masing fungsi keanggotaan yang ada. Semesta pembicaraan dan domain fungsi keanggotaan *entropy* lokal antara 0 sampai dengan 4. Semesta pembicaraan dan domain fungsi keanggotaan jarak lokal yaitu antara 0 hingga 255. Semesta pembicaraan dan domain fungsi keanggotaan standar deviasi lokal antara 0 sampai dengan 150. Kemudian nilai level penyisipan sebagai *output fuzzy* memiliki semesta pembicaraan dan domain dari 0 sampai dengan 8,3.

Dalam fungsi keanggotaan *fuzzy*, penulis menggunakan kurva trapesium. Kurva trapesium memiliki 4 titik ditunjukkan oleh *a*, *b*, *c* dan *d*. Titik *a* menunjukkan nilai domain terkecil yang mempunyai derajat keanggotaan nol. Titik *b* menunjukkan nilai domain terkecil yang mempunyai derajat keanggotaan satu. Titik *c* menunjukkan nilai domain terbesar yang mempunyai derajat keanggotaan satu. Titik *d* menunjukkan nilai domain terbesar yang mempunyai derajat keanggotaan nol. Tabel 3-1 sampai dengan Tabel 3-3 menunjukkan fungsi keanggotaan yang digunakan pada urutan pasangan piksel horizontal vertikal. Sedangkan Tabel 3-4 sampai dengan Tabel 3-6 menunjukkan fungsi keanggotaan yang digunakan pada urutan pasangan piksel terdefinisi. Tabel 3-7 merupakan fungsi keanggotaan *output* berupa nilai level penyisipan.

Tabel 3-1 Fungsi Keanggotaan *Entropy* Lokal (Urutan Pasangan Piksel Horizontal Vertikal)

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-5,00	-1,00	0,00	1,00	Kecil
0,50	1,00	1,50	1,75	Sedang
1,50	1,75	4,00	5,00	Besar

Tabel 3-2 Fungsi Keanggotaan Jarak Lokal (Urutan Pasangan Piksel Horizontal Vertikal)

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-5,00	-1,00	0,00	1,00	Sangat kecil
0,50	1,00	9,50	10,50	Kecil
9,50	10,50	18,00	22,00	Kecil ke sedang
18,00	22,00	28,00	32,00	Sedang ke besar
28,00	32,00	80,00	100,00	Besar
80,00	100,00	255,00	260,00	Sangat besar

Tabel 3-3 Fungsi Keanggotaan Standar Deviasi Lokal (Urutan Pasangan Piksel Horizontal Vertikal)

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-5,00	-1,00	0,00	1,00	Sangat kecil
0,50	1,00	3,50	4,50	Kecil
3,50	4,50	7,50	8,50	Kecil ke sedang
7,50	8,50	15,00	17,50	Sedang ke besar
15,00	17,50	30,00	35,00	Besar
30,00	35,00	150,00	160,00	Sangat besar

Tabel 3-4 Fungsi Keanggotaan Entropy Lokal (Urutan Pasangan Piksel Terdefinisi)

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-5,00	-1,00	0,00	1,00	Kecil
0,50	1,00	1,50	1,75	Sedang
1,50	1,75	4,00	5,00	Besar

Tabel 3-5 Fungsi Keanggotaan Jarak Lokal (Urutan Pasangan Piksel Terdefinisi)

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-5,00	-1,00	0,00	1,00	Sangat kecil
0,50	1,00	9,50	10,50	Kecil
9,50	10,50	14,00	16,00	Kecil ke sedang
14,00	16,00	33,00	36,00	Sedang ke besar
33,00	36,00	80,00	100,00	Besar
80,00	100,00	255,00	260,00	Sangat besar

Tabel 3-6 Fungsi Keanggotaan Standar Deviasi Lokal (Urutan Pasangan Piksel Terdefinisi)

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-5,00	-1,00	0,00	1,00	Sangat kecil
0,50	1,00	3,50	4,50	Kecil

3,50	4,50	7,50	8,50	Kecil ke sedang
7,50	8,50	15,00	17,50	Sedang ke besar
15,00	17,50	30,00	35,00	Besar
30,00	35,00	150,00	160,00	Sangat besar

Tabel 3-7 Fungsi Keanggotaan *Output* Level Penyisipan

Titik a	Titik b	Titik c	Titik d	Variabel Linguistik
-1	0	0,25	0,5	Sangat kecil
0,25	0,5	1,5	1,75	Kecil
1,5	1,75	3	3,5	Kecil ke sedang
3	3,5	5	5,5	Sedang ke besar
5	5,5	7	7,75	Besar
7	7,75	8,3	9	Sangat besar

3.5.3. Perancangan Aturan Kontrol Fuzzy

Terdapat beberapa kombinasi aturan kontrol *fuzzy* yang diterapkan pada penelitian ini. Dari tiga buah variabel *fuzzy* yang dipilih, dibuat tujuh buah kombinasi aturan kontrol terhadap masing-masing tipe pemilihan pasangan piksel. Tujuh buah kombinasi aturan kontrol untuk tipe urutan pasangan piksel horizontal vertikal dan tujuh buah kombinasi aturan untuk tipe urutan pasangan terdefinisi.

Tabel 3-8 sampai dengan Tabel 3-11 menunjukkan aturan yang diterapkan pada tiap variabel *fuzzy* yang ditunjukkan secara linguistik antara lain : sangat kecil (SK), kecil (K), kecil ke sedang (KKS), sedang ke besar (KKB), besar (B) dan sangat besar (SB).

Tabel 3-8 Aturan Kontrol *Entropy* Lokal

	Entropy lokal		
	K	S	B
Level Penyisipan	SB	B	SK

Aturan pada Tabel 3-8 digunakan ketika aturan *entropy* lokal sebagai satu-satunya aturan yang digunakan dalam sistem inferensi *fuzzy*.

Tabel 3-9 Aturan Kontrol Jarak Lokal

	Jarak lokal					
	SK	K	KKS	SKB	B	SB
Level Penyisipan	SB	B	SKB	KKS	K	SK

Aturan pada Tabel 3-9 digunakan ketika aturan jarak lokal sebagai satu-satunya aturan yang digunakan dalam sistem inferensi *fuzzy*.

Tabel 3-10 Aturan Kontrol Standar Deviasi Lokal

	Standar deviasi lokal					
	SK	K	KS	SB	B	SB
Level Penyisipan	SB	B	SKB	KKS	K	SK

Aturan pada Tabel 3-10 digunakan ketika aturan standar deviasi lokal sebagai satu-satunya aturan yang digunakan dalam sistem inferensi *fuzzy*.Aturan juga dapat dikombinasikan antara

aturan *entropy* lokal dengan jarak lokal, aturan *entropy* lokal dengan standar deviasi lokal sepertei ditunjukkan pada Tabel 3-11, aturan jarak lokal dengan standar deviasi lokal, maupun ketiganya.

Tabel 3-11 Kombinasi Aturan Kontrol Jarak Lokal dan Standar Deviasi Lokal

		Jarak lokal					
		SK	K	KS	SB	B	SB
Standar deviasi lokal	SK	SB	SB	SB	SB	SB	SB
	K	SB	B	SKB	KKS	K	SK
	KS	SB	SKB	SKB	KKS	K	SK
	SB	SB	KKS	KKS	KKS	K	SK
	B	SB	K	K	K	K	SK
	SB	SB	SK	SK	SK	SK	SK

3.5.4. Perancangan Defuzzyfikasi

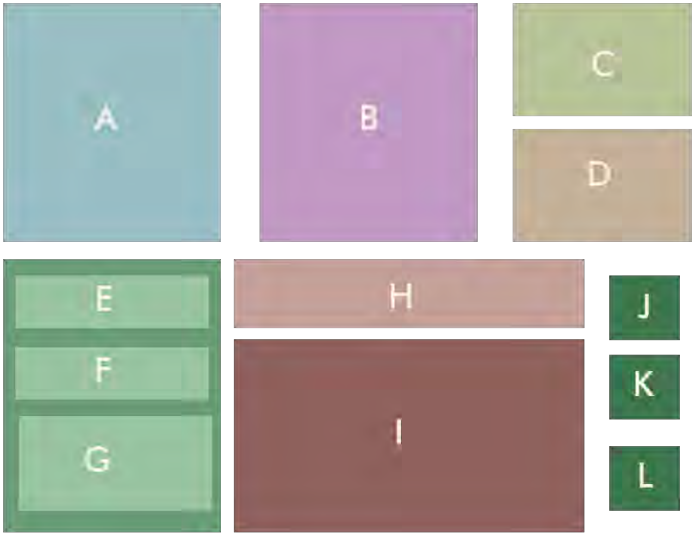
Proses defuzzyfikasi menggunakan metode *centroid* atau sering disebut dengan *Center of Gravity* (COG). Hasil *fuzzy* awal memiliki kemungkinan menghasilkan bilangan pecahan. Namun level penyisipan harus berupa bilangan bulat. Oleh karena itu perlu adanya pembulatan nilai hasil *fuzzy* sebelum digunakan sebagai nilai level penyisipan.

3.6. Rancangan Antarmuka

Rancangan antarmuka dibuat untuk mempermudah pengguna dalam menguji fungsionalitas metode yang diusulkan penulis. Selain itu, pengguna juga dapat memilih urutan pasangan piksel dan kombinasi aturan *fuzzy* yang akan dicoba. Dengan adanya antarmuka, pengguna dapat.

Terdapat sebuah jendela utama yang digunakan dalam antarmuka ini. Penyisipan dan ekstraksi pesan rahasia dilakukan

pada jendela utama. Rancangan jendela utama dapat dilihat pada Gambar 3-30.



Gambar 3-30 Rancangan jendela utama

Komponen-komponen yang terdapat pada jendela utama antara lain:

A. Tampilan citra *cover* yang akan disisipkan pesan rahasia.
B. Tampilan citra *stego* setelah disisipkan pesan rahasia.
C. Tampilan *location map* yang dihasilkan dari proses penyisipan pesan rahasia.
D. Tampilan daftar blok penyisipan atau *block embed map* yang dihasilkan dari proses penyisipan pesan rahasia.
E. Pilihan metode steganografi yang dapat digunakan. Terdapat pilihan metode yang diusulkan oleh Lou dkk. dan metode yang diusulkan oleh penulis.
F. Pilihan metode urutan pasangan piksel yang akan digunakan.

G. Pilihan kombinasi aturan *fuzzy* yang digunakan.
H. Tampilan status penyisipan dan ekstraksi berupa nilai PSNR dan jumlah bit yang berhasil disisipkan atau diekstrak.
I. Tampilan pesan rahasia yang akan disisipkan. Pesan rahasia berupa teks.
J. Tombol penyisipan, untuk memulai proses penyisipan pesan rahasia.
K. Tombol ekstraksi, untuk memulai proses ekstraksi pesan rahasia.
L. Tombol pembersih untuk membersihkan semua elemen tampilan ke tampilan semula.

BAB IV
IMPLEMENTASI

Pada bab ini dibahas implementasi dari rancangan pengembangan metode yang telah dibahas pada Bab III. Bab ini meliputi implementasi RDE *(Reduced Difference Expansion)*, penyisipan pesan rahasia, ekstraksi pesan rahasia dan sistem inferensi *fuzzy* dalam penentuan level.

4.1. Lingkungan Implementasi

Dalam implementasi algoritma digunakan perangkat-perangkat sebagai berikut:

4.1.1. Perangkat Keras

Spesifikasi perangkat keras yang digunakan saat implementasi adalah sebagai berikut.

1. Processor 2.3 GHz Intel Core i3-2350M
2. RAM sebesar 4GB
3. Ruang penyimpanan 640 GB

4.1.2. Perangkat Lunak

Berikut perangkat lunak yang digunakan dalam implementasi:

1. Windows 8.1 Pro
2. MATLAB R2014a

4.2. Implementasi RDE (*Reduced Difference Expansion*)

Terdapat dua buah bagian dalam implementasi RDE. Bagian pertama yaitu implementasi penyisipan bit ke dalam pasangan piksel. Bagian kedua yaitu implementasi proses ekstraksi bit dan pengembalian nilai pasangan piksel ke nilai semula. *Pseudocode* penyisipan RDE dapat dilihat pada Gambar 4-1, dan *pseudocode* ekstraksi RDE dapat dilihat pada Gambar 4-2.

Masukan	Bit yang disisipkan, piksel 1, piksel 2
Keluaran	Piksel 1 yang baru, piksel 2 yang baru, *location map*

```
1. embeddedBit ← input bit
2. x ← piksel 1
3. y ← piksel 2
4. average ← floor((x+y)/2)
5. difference ← x-y
6.
7. differenceReduced ← difference
8.
9. if differenceReduced >= 0
10.    differenceReduced ← difference -
       2^(floor(log2(abs(difference)))-1)
11. else
12.    differenceReduced ← difference +
       2^(floor(log2(abs(difference)))-1)
13.
14. if abs(difference)<2
15.    differenceReduced ← difference
16.
17. if (2^ (floor (log2 (abs (differenceReduced))))
    ==2^(floor (log2 (abs (difference))))) || (abs
    (differenceReduced)==abs(difference))
18.    locationMap ← 0
19. else
20.    locationMap ← 1
21.
22. if differenceReduced>=0
23.    differenceReduced2 ← 2 * differenceReduced +
       embeddedBit
24. else
25.    differenceReduced2 ← 2 * differenceReduced -
       embeddedBit
26.
27. statusValid ←periksa status valid
    differenceReduced2
28. if statusValid==1
29.    x = average + floor((differenceReduced2 +
       1)/2)
30.    y = average - floor(differenceReduced2/2)
```

Gambar 4-1 *Pseudocode* Penyisipan Menggunkan RDE

Masukan	Piksel 1, piksel 2, *input location map*
Keluaran	Piksel 1 awal, piksel 2 awal, bit yang disisipkan

```
1. locationMap ← input location map
2. x ← piksel 1
3. y ← piksel 2
4.
5. average ← floor((x+y)/2)
6. difference ← x-y
7. embeddedBit ← mod(abs(differenceReduced2), 2)
8.
9. if differenceReduced2>=0
10.    differenceReduced ← floor(
       differenceReduced2/2)
11.else
12.    differenceReduced ←
       ceil(differenceReduced2/2)
13.
14.difference ← differenceReduced
15.
16.if locationMap==0
17.    if differenceReduced>=0
18.       if differenceReduced==1
19.          difference ← differenceReduced
20.       else
21.          difference ← differenceReduced +
             2^(floor(log2(abs(differenceReduced)))-
             1)
22.    else
23.       if differenceReduced==-1
24.          difference ← differenceReduced
25.       else
26.          difference ← differenceReduced -
             2^(floor(log2(abs(differenceReduced)))-
             1)
27.
28.elseif locationMap==1
29.    if differenceReduced>=0
30.       if differenceReduced==1
31.          difference ← differenceReduced
32.       else
```

```
33.          difference ← differenceReduced +
             2^(floor(log2(abs(differenceReduced))))
34.    else
35.       if differenceReduced==-1
36.          difference ← differenceReduced
37.       else
38.       difference ← differenceReduced -
          2^(floor(log2(abs(differenceReduced))))
39.
40.
41. x ← average + floor((difference + 1)/2)
42. y ← average - floor(difference/2)
43.
44. if((x-y)==1) && (locationMap==1)
45.    y←y-1
46. elseif((x-y)==-1) && (locationMap==1)
47.    x←x-1
48.
```

Gambar 4-2 *Pseudocode* Ekstraksi Menggunakan RDE

4.3. Implementasi Penyisipan Pesan Rahasia Multi Lapis

Dalam implementasi penyisipa pesan rahasia multi lapis dilakukan bembacaan pada matriks gambar, serta pesan rahasia yang akan disisipkan. Dari matriks gambar yang ada, maka didapatkan seluruh blok yang ada, kemudian menentukan level penyisipan masing-masing blok dengan menggunakan sistem inferensi *fuzzy*. Pesan rahasia yang sebelumnya berupa *string* dirubah menjadi deret bit sehingga dapat disisipkan menggunakan metode RDE.

Dari daftar level penyisipan seluruh blok, dipilih level penyisipan yang paling tinggi sebagai indikator jumlah iterasi atau lapis yang akan diproses. Begitu pula dari daftar level dibuat daftar blok yang *expandable* yang berisi status apakah sebuah blok dapat diikutkan dalam proses penyisipan atau tidak. Dalam setiap iterasi, dilakukan penyisiran pada tiap blok satu per satu. Ketika status blok *expandable* adalah satu, maka akan dilakukan proses penyisipan pada blok tersebut.

Dalam sebuah blok terdapat dua buah pasangan piksel, sehingga input bit yang diperlukan juga berjumlah dua piksel. Oleh karena itu setiap dilakukan penyisipan pada suatu blok diambil dua bit dari deret bit pesan rahasia. Jika proses penyisipan RDE berhasil, maka akan didapatkan nilai piksel baru dan juga nilai *location map*.

Ketika seluruh blok sudah diproses, maka proses berlanjut ke iterasi berikutnya sampai jumlah iterasi maksimal. *Pseudocode* penyisipan pesan rahasia multi lapis dapat dilihat pada Gambar 4-3.

Masukan	Citra *grayscale* 8 bit, pesan rahasia
Keluaran	Citra stego, *location map*, daftar blok yang berhasil mengalami penyisipan bit

```
imageMatrix ← baca matrik citra
message ← pesan rahasia

listCapacityLevel ← hitung level setiap blok
menggunakan logika fuzzy

listInputBit ← konversi message ke dalam array
dari bit

levelLayer ← level maksimal pada
listCapacityLevel
listBlockExpandable ← konversi
listCapacityLevel ke dalam array dua dimensi
dengan nilai 1 dan 0 sesuai dengan level yang
disimpan

 for k ← 1 to levelLayer do
    Baca blok pada imageMatrix satu per satu
    if listBlockExpandable[k,indekBlok] == 1
       Sisipkan bit dari listInputBit ke dalam
       blok menggunakan RDE
       Jika penyisipan berhasil
          Simpan nilai blok baru
          listLocationMap[k,indeks location map]
          ← nilai location map
          listBlockEmbed[k,indeks blok] ← 1
```

```
20. Simpan nilai imageMatrix yang baru sebagai citra
    stego
21. Simpan listLocationMap dan listBlockEmbed agar
    dapat digunakan dalam proses ekstraksi citra
22.
```

Gambar 4-3 *Pseudocode* Penyisipan Multi Lapis

4.4. Implementasi Ekstraksi Pesan Rahasia Multi Lapis

Dalam implementasi ekstraksi pesan rahasia multi lapis dilakukan pembacaan pada matriks gambar *stego*, location map, dan daftar status blok yang berhasil disisipkan. Dari daftar status blok yang disisipkan dicari level penyisipan terbesar sebagai jumlah iteraksi maksimal.

Dalam setiap iterasi dilakukan penyisiran pada setiap blok satu per satu. Ketika status penyisipan blok tersebut pada itersi yang bersangkutan adalah satu, maka akan dilakukan proses ekstraksi. Dalam proses ekstraksi diperlukan nilai location map dan nilai pasangan piksel. Setelah proses ekstraksi selesai pada blok tersebut maka akan didapatkan nilai bit rahasia dan nilai pasangan piksel awal.

Ketika seluruh blok sudah diproses, maka proses berlanjut ke iterasi berikutnya sampai jumlah iterasi maksimal. Semua deret bit rahasia yang didapatkan kemudian digabung kemudian dikonversi ke bentuk *string* untuk mendapatkan pesan rahasia secara utuh. *Pseudocode* ekstraksi pesan rahasia multi lapis dapat dilihat pada Gambar 4-4.

Masukan	Citra stego, *location map*, daftar blok yang berhasil mengalami penyisipan bit
Keluaran	Citra awal, pesan rahasia

```
1. listLocationMap ← location map
2. listBlockEmbed ← daftar blok yang berhasil
   mengalami penyisipan bit
3. levelLayer ← jumlah baris dari array
   listBlockEmbed
4.
5. for k ← 1 to levelLayer do
```

```
6.     Baca blok pada imageMatrix satu per satu
7.     if listBlockEmbed [k,indekBlok] == 1
8.       ekstrak bit berdasarkan berdasarkan
         location map menggunakan RDE
9.       Jika penyisipan berhasil
10.         Simpan nilai blok baru
11.         listInputBit[idexBit] ← hasil ekstraksi
12.
23. Simpan nilai imageMatrix yang baru sebagai citra
    cover
13. Message ← konversi listInputBit ke dalam string
14.
```

Gambar 4-4 *Pseudocode* Ekstraksi Multi Lapis

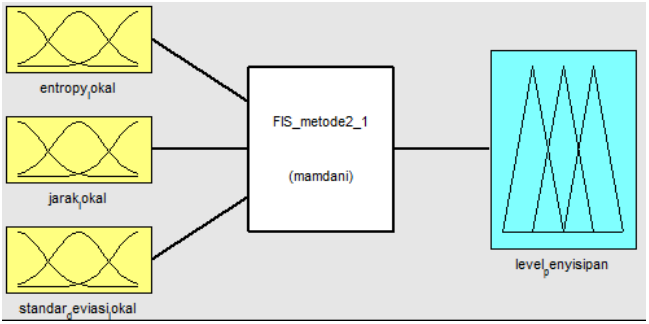
4.5. Implementasi Sistem Inferensi *Fuzzy*

Dalam implementasi sistem inferensi *fuzzy* dilakukan pembuatan fungsi keanggotaan dan pembuatan aturan kontrol.

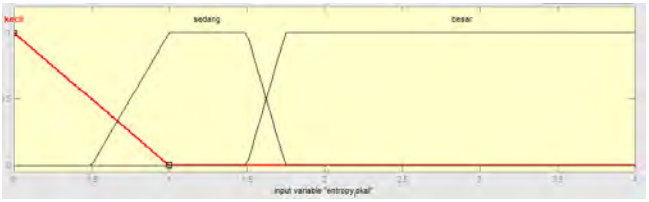
4.3.1. Implementasi Fungsi Keanggotaan

Dalam implementasi fungsi keanggotaan dilakukan pembuatan fungsi keanggotaan berdasarkan pada Tabel 3-1 sampai dengan Tabel 3-6**.** Begitu pula dilakuakn pembuatan fungsi keanggotaan *output* seperti pada Tabel 3-7. Implementasi fungsi keanggotaan dibuat menggukan kurva trapesium.

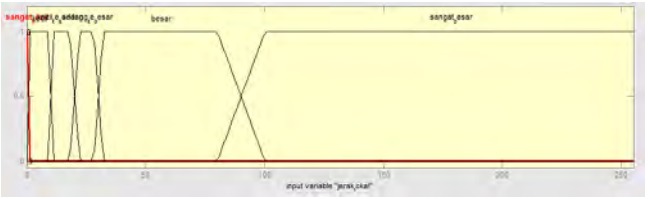
Pada MATLAB R2014a, hasil implementasi fungsi keanggotaan dapat dilihat pada Gambar 4-5 sampai dengan Gambar 4-8. Gambar 4-5 Menunjukkan skema hubungan antara fungsi keanggotaan *input*, aturan dan fungi keanggotaan *output*. Gambar 4-6 sampai dengan Gambar 4-8 masing-masing merupakan grafik yang menunjukkan fungsi keanggotaan entropy lokal, jarak lokal dan standar deviasi lokal.



Gambar 4-5 Bagan Keseluruhan Implementasi Fungsi Keanggotaan *Fuzzy*



Gambar 4-6 Grafik Fungsi Keanggotaan *Entropy* Lokal



Gambar 4-7 Grafik Fungsi Keanggotaan Jarak Lokal

Gambar 4-8 Grafik Fungsi Keanggotaan Standar Deviasi Lokal

4.3.2. Implementasi Aturan Kontrol

Dalam implementasi aturan kontrol dilakukan penerjemahan aturan seperti pada Tabel 3-8 sampai dengan Tabel 3-11 menjadi bentu aturan IF-THEN. A sampai dengan merupakan implementasi aturan kontrol *fuzzy* dalam bentuk aturan IF-THEN.

```
1. if (entropy_lokal is kecil) then
   (level_penyisipan is sangat_besar)
2. if (entropy_lokal is sedang) then
   (level_penyisipan is besar)
3. if (entropy_lokal is besar) then
   (level_penyisipan is sangat_kecil)
```

Gambar 4-9 Implementasi Aturan *Entropy* Lokal

```
1. if (jarak_lokal is sangat_kecil) then
   (level_penyisipan is sangat_besar)
2. if (jarak_lokal is kecil) then (level_penyisipan
   is besar)
3. if (jarak_lokal is kecil_ke_sedang) then
   (level_penyisipan is sedang_ke_besar)
4. if (jarak_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
5. if (jarak_lokal is besar) then (level_penyisipan
   is kecil)
6. if (jarak_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
```

Gambar 4-10 Implementasi Aturan Jarak Lokal

```
1. if (standar_deviasi_lokal is sangat_kecil) then
   (level_penyisipan is sangat_besar)
2. if (standar_deviasi_lokal is kecil) then
   (level_penyisipan is besar)
3. if (standar_deviasi_lokal is kecil_ke_sedang)
   then (level_penyisipan is sedang_ke_besar)
4. if (standar_deviasi_lokal is sedang_ke_besar)
   then (level_penyisipan is kecil_ke_sedang)
5. if (standar_deviasi_lokal is besar) then
   (level_penyisipan is kecil)
6. if (standar_deviasi_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
```

Gambar 4-11 Implementasi Aturan Standar Deviasi Lokal

```
1. if (entropy_lokal is kecil) and (jarak_lokal is
   sangat_kecil) then (level_penyisipan is
   sangat_besar)
2. if (entropy_lokal is sedang) then
   (level_penyisipan is besar)
3. if (jarak_lokal is kecil) then (level_penyisipan
   is besar)
4. if (jarak_lokal is kecil_ke_sedang) then
   (level_penyisipan is sedang_ke_besar)
5. if (jarak_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
6. if (jarak_lokal is besar) then (level_penyisipan
   is kecil)
7. if (jarak_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
```

Gambar 4-12 Implementasi Kombinasi Aturan *Entropy* Lokal dan Jarak Lokal

```
1. if (entropy_lokal is kecil) and
   (standar_deviasi_lokal is sangat_kecil) then
   (level_penyisipan is sangat_besar)
2. if (entropy_lokal is sedang) then
   (level_penyisipan is besar)
3. if (standar_deviasi_lokal is kecil) then
   (level_penyisipan is besar)
4. if (standar_deviasi_lokal is kecil_ke_sedang)
   then (level_penyisipan is sedang_ke_besar)
5. if (standar_deviasi_lokal is sedang_ke_besar)
   then (level_penyisipan is kecil_ke_sedang)
6. if (standar_deviasi_lokal is besar) then
   (level_penyisipan is kecil)
7. if (standar_deviasi_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
```

Gambar 4-13 Implementasi Kombinasi Aturan *Entropy* Lokal dan Standar Deviasi Lokal

```
1. if (jarak_lokal is sangat_kecl) and
   (standar_deviasi_lokal is sangat_kecil) then
   (level_penyisipan is sangat_besar)
2. if (jarak_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
3. if(standar_deviasi_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
```

```
4. if (jarak_lokal is besar) and
   (standar_deviasi_lokal is not sangat_besar) then
   (level_penyisipan is kecil)
5. if (jarak_lokal is not sangat_besar) and
   (standar_deviasi_lokal is besar) then
   (level_penyisipan is kecil)
6. if (jarak_lokal is sedang_ke_besar) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is kecil_ke sedang)
7. if (jarak_lokal is sedang_ke_besar) and
   (standar_deviasi_lokal kecil_ke_sedang) then
   (level_penyisipan is kecil_ke_sedang)
8. if (jarak_lokal is sedang_ke_besar) and
   (standar_deviasi_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
9. if (jarak_lokal is kecil_ke_sedang) and
   (standar_deviasi_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
10. if (jarak_lokal is kecil) and
   (standar_deviasi_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
11. if (jarak_lokal is kecil_ke_sedang) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is sedang_ke_besar)
12. if (jara_lokal is kecil_ke_sedang) and
   (standar_deviasi_lokal is kecil_ke_sedang) then
   (level_penyisipan is sedang_ke_besar)
13. if (jarak_lokal is kecil) and
   (standar_deviasi_lokal is kecil_ke_sedang) then
   (level_penyisipan is sedang_ke_besar)
14. if (jarak_lokal is kecil) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is besar)
```

Gambar 4-14 Implementasi Kombinasi Aturan Jarak Lokal dan Standar Deviasi Lokal

```
1. if (entropy_lokal is kecil) and (jarak_lokal is
   sangat_kecil) and (standar_deviasi_lokal is
   sangat_kecil) then (level_penyisipan is
   sangat_besar)
2. is (jarak_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
```

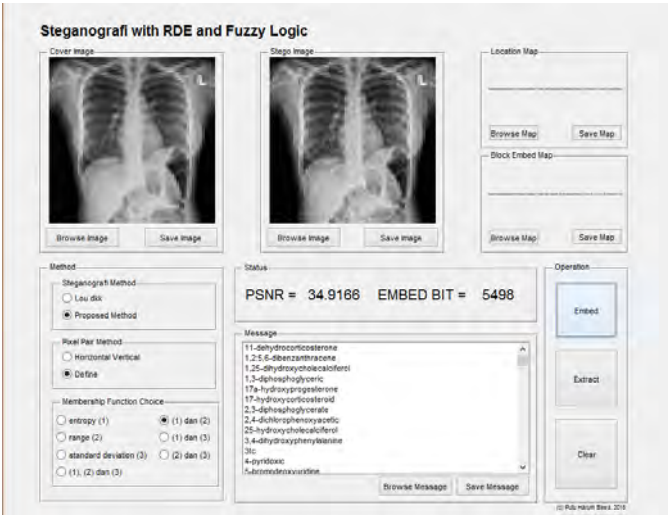
```
3. if (standar_deviasi_lokal is sangat_besar) then
   (level_penyisipan is sangat_kecil)
4. if (jarak_lokal is besar) and
   (standar_deviasi_lokal is not sangat_besar) then
   (level_penyisipan is kecil)
5. if (jarak_lokal is not sangat_besar) and
   (standar_deviasi_lokal is besar) then
   (level_penyisipan is kecil)
6. if (jarak_lokal is sedang_ke_besar) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is kecil_ke_sedang)
7. if (jarak_lokal is sedang_ke_besar) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is kecil_ke_sedang)
8. if (jarak_lokal is sedang_ke_besar) and
   (standar_deviasi_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
9. if (jarak_lokal is kecil_ke_sedang) and
   (standar_deviasi_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
10. if (jarak_lokal is kecil) and
   (standar_deviasi_lokal is sedang_ke_besar) then
   (level_penyisipan is kecil_ke_sedang)
11. if (jarak_lokal is kecil_ke_sedang) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is sedang_ke_besar)
12. if (jarak_lokal is kecil_ke_sedang) and
   (standar_deviasi_lokal is kecil_ke_sedang) then
   (level_penyisipan is sedang_ke_besar)
13. if (jarak_lokal is kecil) and
   (standar_deviasi_lokal is kecil_ke_sedang) then
   (level_penyisipan is sedang_ke_besar)
14. if (jarak_lokal is kecil) and
   (standar_deviasi_lokal is kecil) then
   (level_penyisipan is besar)
15. if (entropy_lokal is sedang) then
   (level_penyisipan is besar)
```

Gambar 4-15 Implementasi Kombinasi Aturan Entropy Lokal, Jarak Lokal dan Standar Deviasi Lokal

4.6. Implementasi Antarmuka

Antarmuka diimplementasikan menggunakan MATLAB R2014a. Antarmuka dibuat secara sederhana dengan memanfaatkan elemen *static text*, *edit text*, *radio button*, *panel*, *axes*, dan *push button*. Tampilan jendela utama dapat dilihat pada Gambar 4-16.

Static text digunakan untuk menampilkan nilai PSNR dan jumlah bit yang berhasil disisipkan ataupun diekstrak. *Edit text* digunakan untuk menampilkan pesan rahasia yang akan disisipkan atau berhasil diekstrak. *Radio button* digunakan untuk menampilkan pilihan metode, urutan pasangan piksel dan kombinasi aturan *fuzzy*. *Panel* digunakan untuk mengelompokkan elemen-elemen antarmuka agar lebih mudah digunakan oleh pengguna. *Axes* digunakan untuk menampilkan citra *cover*, citra *stego*, *location map* dan *block embed map*. *Push button* digunakan sebagai tombol untuk mengeksekusi perintah-perintah yang telah disediakan.



Gambar 4-16 Tampilan Jendela Utama

Penjelasan tombol-tombol yang terdapat pada jendela utama antara lain:

1. *Browse Image* digunakan untuk membuka file citra, baik citra *cover* maupun citra *stego* dengan ekstensi file .png.
2. *Save Image* digunakan untuk menyimpan file citra, baik citra *cover* maupun citra *stego*. File disimpan ke dalam ekstensi file .png.
3. *Browse Map* digunakan untuk membuka file *map*, baik *location map* maupun *block embed map* dengan ekstensi file .bmp.
4. *Save Map* digunakan untuk menyimpan file *map*, baik *location map* maupun *block embed map*. File disimpan ke dalam ekstensi file .bmp.
5. *Browse Message* digunakan untuk membuka file pesan rahasia dengan ekstensi .txt.
6. *Save Message* digunakan untuk menyimpan file pesan rahasia dengan ekstensi .txt.
7. *Embed* digunakan untuk memulai proses penyisipan pesan rahasia ke dalam citra *cover*.
8. *Extract* digunakan untuk memulai proses ekstraksi pesan rahasia.
9. *Clear* digunakan untuk membersihkan elemen antarmuka ke kondisi awal.

BAB V
UJI COBA DAN EVALUASI

Pada bab ini, dibahas uji coba dan evaluasi dari implementasi metode yang diusulkan.

5.1. Lingkungan Uji Coba

Uji coba dilakukan pada komputer yang telah terpasang perangkat lunak MATLAB R2014a. Untuk spesifikasi dari computer yang digunakan untuk uji coba dapat dilihat pada Tabel 5-1.

Tabel 5-1 Spesifikasi Komputer Uji Coba

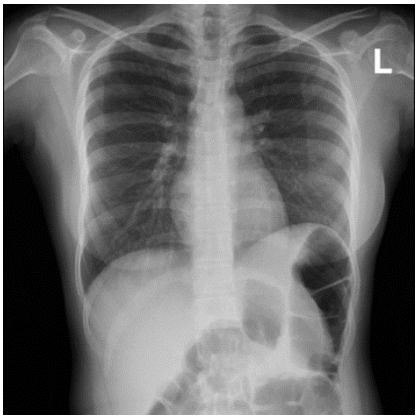
Komponen	Spesifikasi
CPU	Intel® Core™ i3-3240 CPU @ 3.40GHz 3.40 GHz
Sistem Operasi	Windows 8 64
Memori	4GB PC3-10600 DDR3 SDRAM 1333 MHz
Grafis	Intel® HD Graphics 2000 (integrated)
Port I/O	6 USB 2.0 ports
	2 USB 3.0 ports (optional)
	7-in-1 card reader (SD, SDHC, SDXC, MS, MS Pro, MMC, MMC Plus)
	Headphone jack
	Microphone jack
	VGA
	HDMI
Penyimpanan	1TB, 2TB HDD (7200 rpm)
Konektifitas	LAN: 10/100/1,000M (RJ-45 connector)

Pengujian dilakukan dengan cara membuat sebuah kode program untuk menjalankan secara berulang metode yang akan diuji. Kemudian hasilnya akan disimpan dalam file berekstensi .xlsx.

5.2. Data Uji Coba

Terdapat lima buah citra medis [15] yang digunakan dalam proses uji coba. Kelima daftar citra medis tersebut dapat dilihat pada Gambar 5-1 Radiograf dada sampai dengan Gambar 5-5 Radiograf leher. Masing-masing citra tersebut akan menjadi citra *cover* dalam proses penyisipan data rahasia. Citra medis yang digunakan berupa citra *grayscale* berukuran 64×64 piksel dan 512×512 piksel. Citra dengan ukuran 64×64 piksel digunakan dalam uji coba fungsionalitas sedangkan citra dengan ukuran 512×512 piksel digunakan dalam uji coba performa.

Data rahasia yang digunakan berupa daftar istilah kedokteran [16] yang kemudian disimpan dalam bentuk file .txt. Untuk keperluan uji coba, file tersebut dibuat menjadi beberapa versi dimana masing-masing versi memiliki ukurannya tersendiri. Masing-masing versi file dan ukurannya dapat dilihat pada Tabel 5-2.

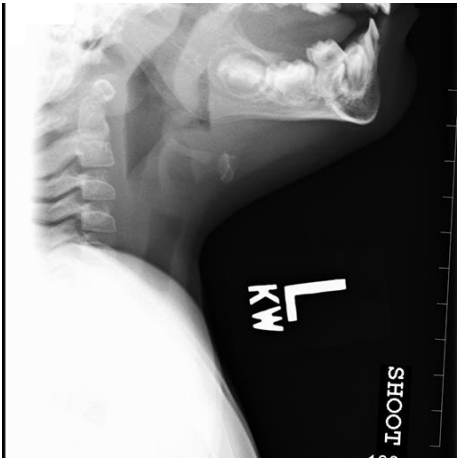


Gambar 5-1 Radiograf dada

Gambar 5-2 Radiograf tangan

Gambar 5-3 Radiograf kaki

Gambar 5-4 Radiograf paru-paru



Gambar 5-5 Radiograf leher

Tabel 5-2 Daftar Ukuran Data Rahasia

Nama file data rahasia	Ukuran data rahasia
1.txt	50000 bit
2.txt	250000 bit
3.txt	500000 bit
4.txt	1,148 Kilo Byte

5.3. Skenario Uji Coba

Skenario uji coba diperlukan untuk mengetahui kinerja metode yang diusulkan. Skenario uji coba juga dapat digunakan untuk membandingkan kemampuan metode yang diusulkan dengan metode yang telah dikembangkan sebelumnya.

5.3.1. Skenaro Uji Coba Fungsionalitas

Uji coba fungsionalitas dilakukan dengan cara menjalankan aplikasi yang sudah diimplementasikan pada Bab 4. Uji fungsionalitas meliputi dua buah proses pada steganografi, yaitu pengujian pada proses penyisipan pesan rahasia dan pengujian pada proses ekstraksi data rahasia.

Metode steganografi yang diusulkan bekerja dengan baik bila pesan rahasia dapat diekstrak secara utuh. Pengertian diekstrak secara utuh yaitu tidak terdapat karakter yang berubah jika dibandingkan dengan pesan rahasia yang disisipkan sebelumnya. Namun jumlah karakter ataupun jumlah bit yang dapat disisipkan berbeda-beda sesuai dengan kapasitas citra *cover* bersangkutan. Sehingga ada kemungkinan ketika proses penyisipan dilakukan dan jumlah pesan rahasia melebihi ukuran citra *cover*, maka pesan yang disisipkan hanya sejumlah kapasitas penyisipan citra *cover* tersebut.

Pada uji coba fungsionalitas citra yang digunakan yaitu citra yang ditunjukkan pada Gambar 5-3 namun dengan ukuran 64×64 piksel. Uji coba fungsionalitas dilakukan terhadap semua

kombinasi yang mungkin dari metode yang diusulkan. Uji coba fungsionalitas juga perlu dilakukan pada metode Lou dkk. yang sudah diimplementasikan. Metode Lou dkk. akan dijadikan pembanding terhadap performa metode yang diusulkan.

Keseluruhan kombinasi uji coba fungsionalitas dapat dilihat pada tabel Tabel 5-3. Aturan kontrol pada *fuzzy* diwakili oleh F1 sampai dengan F7. F1, F2, F3, F4, F5, F6 dan F7 secara berurutan antara lain aturan *entropy* lokal, jarak lokal, standar deviasi lokal, kombinasi aturan *entropy* lokal dan jarak lokal, *etropy* lokal dan standar deviasi

Tabel 5-3 Skenario Uji Coba Fungsionalitas

No	Urutan Pasangan Piksel	Kombinasi Aturan Fuzzy
1	Lou dkk.	
2	Horizontal Vertikal	F1
3		F2
4		F3
5		F4
6		F5
7		F6
8		F7
9	Terdefinisi	F1
10		F2
11		F3
12		F4
13		F5
14		F6
15		F7

Pesan rahasia yang akan disisipkan pada uji coba fungsionalitas dapat dilihat pada Gambar 5-6.

```
StudyDate: '20150716'
SeriesDate: '20150716'
AcquisitionDate: '20150716'
ImageDate: '20150716'
StudyTime: '102008'
SeriesTime: '102008'
AcquisitionTime: '102008'
ImageTime: '102008'
AccessionNumber: ''
Modality: 'MR'
Manufacturer: 'GE Medical Systems'
InstitutionName: 'GENERAL HOSPITAL'
ReferringPhysiciansName: [1x1 struct]
StationName: 'MRS1'
StudyDescription: 'LEG'
SeriesDescription: 'FSE PD AXIAL OBL'
PerformingPhysiciansName: [1x1 struct]
OperatorName: [1x1 struct]
OperatorsName: 'HARUM BAWA'
PatientsName: [1x1 struct]
PatientID: '12345687'
PatientsBirthDate: ''
PatientsSex: 'F'
PatientsAge: '028Y'
PatientsWeight: 61.2350
```

Gambar 5-6 Pesan rahasia pada uji coba fungsionalitas

5.3.2. Skenaro Uji Coba Performa

Uji coba performa dilakukan untuk mengetahui kualitas metode yang diusulkan. Semakin mirip citra *stego* yang dihasilkan dengan citra *cover*, maka semakin baik metode steganografi yang diusulkan. Tingkat kemiripan ataupun kualitas citra *stego* diukur menggunakan nilai PSNR (dalam dB) citra *stego*.

Terdapat beberapa skenario yang digunakan untuk mengetahui kualitas citra *stego* yang dihasilkan, antara lain skenario penyisipan 50 kb, 250 kb dan 500 kb data rahasia. Penyisipan dilakukan dengan dua buah metode. Metode pertama yaitu metode yang diusulkan oleh penulis, metode kedua yaitu metode yang diusulkan oleh Lou dkk. dengan beberapa perubahan. Metode awal Lou dkk. [3] dikombinasikan dengan metode Tohari dkk. [1] dalam hal

penanganan selisih piksel yang bernilai negatif. Namun metode penyisipan multi lapis tetap sesuai dengan metode yang diusulkan oleh Lou dkk.

Nilai PSNR masing-masing metode dibandingkan satu sama lain. Dengan cara ini, dapat diketahui seberapa baik metode yang dikembagnkan dibandingkan dengan metode yang telah dibuat sebelumnya.

Terdapat pula skenario untuk mengetahui kapasitas maksimal bit yang bisa disisipkan menggunakan masing-masing metode. Pada skenario ini disiapkan sebuah data rahasia yan berukuran 1,148 kilo byte.

5.4. Uji Coba

Dalam uji coba ditampilkan hasil uji coba sesuai skenario. Hasil uji coba selanjutnya dapat digunakan sebagai bahan evaluasi untuk menilai kinerja metode yang diusulkan.

5.4.1. Uji Coba Fungsionalitas

Hasil uji coba fungsionalitas dapat dilihat pada Tabel 5-4.

Tabel 5-4 Hasil Uji Coba Fungsionalitas

No	Urutan Pasangan Piksel	Kombinasi Aturan *Fuzzy*	Status Penyisipan	Status Ekstraksi
1	Lou dkk.		Sukses	Sukses
2	Horizontal Vertikal	F1	Sukses	Sukses
3		F2	Sukses	Sukses
4		F3	Sukses	Sukses
5		F4	Sukses	Sukses
6		F5	Sukses	Sukses
7		F6	Sukses	Sukses
8		F7	Sukses	Sukses
9	Terdefinisi	F1	Sukses	Sukses

10		F2	Sukses	Sukses
11		F3	Sukses	Sukses
12		F4	Sukses	Sukses
13		F5	Sukses	Sukses
14		F6	Sukses	Sukses
15		F7	Sukses	Sukses

5.4.2. Uji Coba Performa

Hasil uji coba penyisipan data rahasia dengan ukuran 50 kbit, 250 kbit dan 500 kbit dapat dilihat pada Tabel 5-5. Hasil uji coba berupa nilai PSNR (dalam dB). Aturan kontrol pada *fuzzy* diwakili oleh F1 sampai dengan F7. F1, F2, F3, F4, F5, F6 dan F7 secara berurutan antara lain aturan *entropy* lokal, jarak lokal, standar deviasi lokal, kombinasi aturan *entropy* lokal dan jarak lokal, *etropy* lokal dan standar deviasi lokal, jarak lokal dan standar deviasi lokal, serta yang terakhir kombinasi aturan *entropy* lokal, jarak lokal dan standar deviasi lokal.

Tabel 5-5 Hasil Uji Coba Berupa Nilai PSNR pada Penyisipan 50 Kbit, 250 Kbit dan 500 Kbit Data Rahasia

Gambar	Metode urutan pasangan piksel	Aturan Fuzzy	Ukuran Data Rahasia		
			50 kbit	250 kbit	500 kbit
Dada	Lou dkk.		52.89	45.97	39.95
	Horizontal vertikal	F1	53.47	46.09	40.36
		F2	52.97	46.00	40.31
		F3	53.47	46.06	40.32
		F4	53.47	46.09	40.36
		F5	52.97	45.90	39.83
		F6	52.97	46.00	40.31
		F7	53.47	46.06	40.32

	Terdefinisi	F1	53.47	46.08	40.58
		F2	52.97	45.98	40.56
		F3	53.47	46.06	40.31
		F4	53.47	46.08	40.58
		F5	52.97	45.90	39.82
		F6	52.97	45.98	40.56
		F7	53.47	46.06	40.31
Tangan	Lou dkk.		52.12	43.81	37.66
	Horizontal vertikal	F1	52.14	44.59	39.74
		F2	52.14	44.55	39.66
		F3	52.14	44.46	39.66
		F4	52.14	44.59	39.74
		F5	52.14	43.71	37.74
		F6	52.14	44.55	39.66
		F7	52.14	44.46	39.66
	Terdefinisi	F1	52.14	44.53	40.09
		F2	52.14	44.48	40.05
		F3	52.14	44.46	39.69
		F4	52.14	44.53	40.09
		F5	52.14	43.71	37.79
		F6	52.14	44.48	40.05
		F7	52.14	44.46	39.69
Kaki	Lou dkk.		51.29	45.41	39.73
	Horizontal vertikal	F1	54.11	46.41	40.80
		F2	54.01	46.39	40.78
		F3	54.11	46.36	40.77
		F4	54.11	46.42	40.81
		F5	51.54	45.37	39.85
		F6	54.01	46.39	40.78

		F7	54.11	46.37	40.78
	Terdefinisi	F1	54.11	46.39	40.94
		F2	54.01	46.36	40.92
		F3	54.11	46.36	40.79
		F4	54.11	46.40	40.94
		F5	51.54	45.37	39.78
		F6	54.01	46.36	40.93
		F7	54.11	46.37	40.79
Paru-paru	Lou dkk.		53.41	46.88	39.96
	Horizontal vertikal	F1	54.29	46.22	38.70
		F2	54.29	46.19	38.74
		F3	54.29	46.20	38.69
		F4	54.29	46.22	38.69
		F5	53.53	45.68	37.89
		F6	54.29	46.19	38.74
		F7	54.29	46.20	38.71
	Terdefinisi	F1	54.29	46.51	39.14
		F2	54.29	46.51	39.24
		F3	54.29	46.43	39.22
		F4	54.29	46.52	39.15
		F5	53.53	45.97	38.28
		F6	54.29	46.52	39.26
		F7	54.29	46.43	39.22
Leher	Lou dkk.		52.42	44.33	38.71
	Horizontal vertikal	F1	52.65	45.72	40.04
		F2	52.53	45.69	40.05
		F3	52.65	45.59	40.02
		F4	52.65	45.81	40.05
		F5	52.43	44.91	38.76

		F6	52.53	45.76	40.07
		F7	52.65	45.66	40.04
	Terdefinisi	F1	52.65	45.64	40.21
		F2	52.53	45.61	40.22
		F3	52.65	45.59	40.08
		F4	52.65	45.72	40.22
		F5	52.43	44.91	38.66
		F6	52.53	45.67	40.25
		F7	52.65	45.66	40.09

Hasil uji coba penyisipan maksimal dapat dilihat pada Tabel 5-6. Dalam hasil uji coba ini didapatkan nilai kapasitas berupa jumlah bit maksimal yang dapat disisipkan serta nilai PSNR ketika penyisipan maksimal dilakukan.

Tabel 5-6 Hasil Uji Coba Penyisipan Maksimal

Gambar	Metode	Aturn Fuzzy	Kapasitas (bit)	PSNR (dB)
Dada	Lou dkk.		1014463	31.04
	Horizontal vertikal	F1	710960	36.59
		F2	735792	36.39
		F3	760874	35.94
		F4	710960	36.59
		F5	992770	31.01
		F6	735792	36.39
		F7	760874	35.94
	Terdefinisi	F1	709718	35.77
		F2	737012	35.56
		F3	769378	35.08
		F4	709718	35.77

		F5	1000006	27.21
		F6	737012	35.56
		F7	769378	35.08
Tangan	Lou dkk.		1045556	28.59
	Horizontal vertikal	F1	670600	36.53
		F2	770854	35.65
		F3	882844	33.76
		F4	670600	36.53
		F5	1044236	28.60
		F6	770854	35.65
		F7	882836	33.76
	Terdefinisi	F1	664906	36.64
		F2	764394	35.88
		F3	882686	33.90
		F4	664906	36.64
		F5	1039050	25.93
		F6	764394	35.88
		F7	882686	33.90
Kaki	Lou dkk.		1028764	31.07
	Horizontal vertikal	F1	712878	36.64
		F2	779018	36.07
		F3	847000	34.95
		F4	712948	36.67
		F5	1011498	31.14
		F6	779070	36.07
		F7	846948	34.95
	Terdefinisi	F1	711924	36.36
		F2	778036	35.90
		F3	848396	34.83

		F4	711752	36.37
		F5	1010920	28.28
		F6	777892	35.91
		F7	848338	34.86
Paru-paru	Lou dkk.		872825	32.96
	Horizontal vertikal	F1	513586	38.47
		F2	523014	38.34
		F3	535850	37.85
		F4	513710	38.46
		F5	683056	33.08
		F6	522962	38.34
		F7	535992	37.87
	Terdefinisi	F1	548254	37.88
		F2	557262	37.73
		F3	573420	37.27
		F4	548350	37.90
		F5	718658	29.85
		F6	557364	37.76
		F7	572924	37.28
Leher	Lou dkk.		948052	30.68
	Horizontal vertikal	F1	652496	36.96
		F2	714452	36.42
		F3	766808	35.57
		F4	650694	36.99
		F5	898804	30.93
		F6	713282	36.45
		F7	765578	35.62
	Terdefinisi	F1	653146	37.05
		F2	719144	36.57

		F3	780108	35.66
		F4	652330	37.07
		F5	908830	28.75
		F6	718442	36.59
		F7	779812	35.68

5.5. Evaluasi Uji Coba

Pada bagian ini dilakukan evaluasi atas hasil uji coba yang telah diteangkan pada subbab 5.4. Evaluasi uji coba secara umum terdiri dari dua bagian yaitu evaluasi terhadap hasil uji coba fungsional dan evaluasi terhadap hasil uji coba performa.

5.5.1. Evaluasi Uji Coba Fungsionalitas

Dari hasil uji coba fungsionaliyas terhadap metode yang dikembangkan, maka dapat dilihat bahwa metode bekerja dengan baik. Pesan rahasia yang disisipkan sebelumnya dapat diekstrak kemabali secara utuh dan akurat.

5.5.2. Evaluasi Uji Coba Performa

Selain uji coba fungsionalitas, metode yang diajukan oleh penulis juga diuji performanya. Berdasarkan kombinasi antara metode pemilihan urutan pasangan piksel dan kombinasi aturan kontrol *fuzzy*, terdapat beberapa tahap untuk menguji performa metode yang dikembangkan.

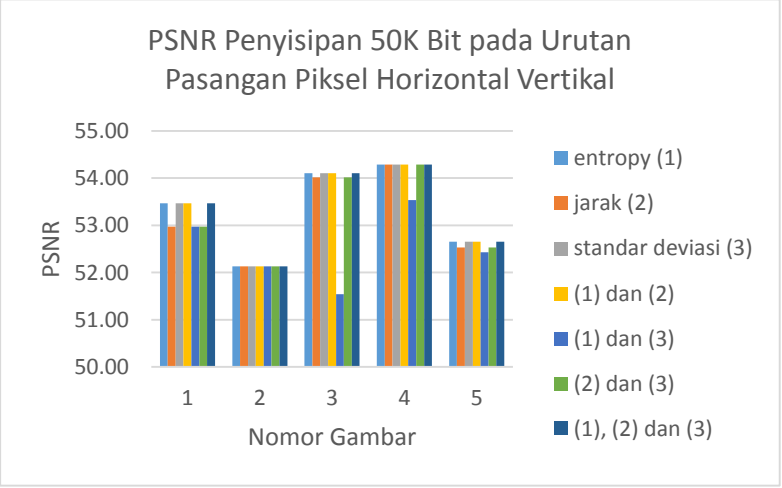
Pertama dilakukan evaluasi terhadap hasil uji coba penyisipan menggunakan metode urutan pasangan piksel horizontal dan vertikal. Kemudian dievaluasi aturan kontrol *fuzzy* yang mana yang memiliki hasil PSNR terbaik ketika menggunakan metode urutan pasangan piksel tersebut. Evaluasi ini juga dilakukan pada metode urutan pasangan piksel terdefinisi.

Tahap selanjutnya yaitu melakukan perbandingan nilai PSNR dari masing-masing kombinasi metode terpilih dengan

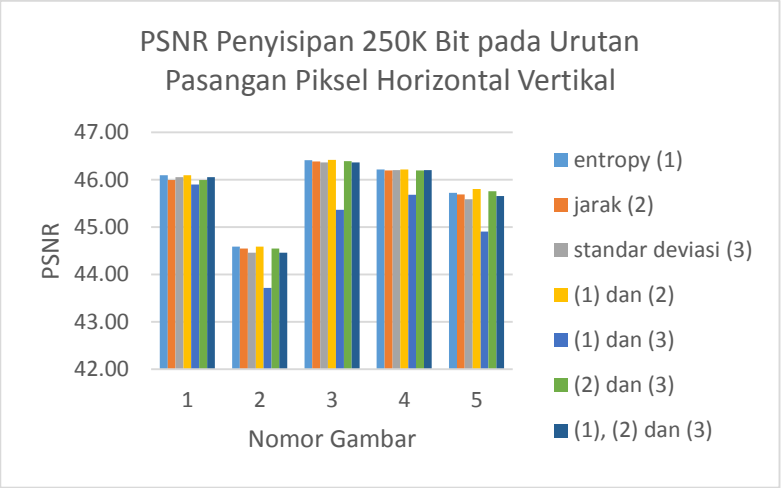
metode yang sudah dikembangkan sebelumnya. Dari evaluasi ini dapat diketahui apakah metode yang diusulkan lebih baik atau lebih buruk dari metode sebelumnya.

Grafik pada Gambar 5-7 sampai dengan Gambar 5-12 menunjukkan bahwa masing-masing gambar uji coba memiliki nilai PSNR berbeda-beda ketika dilakukan proses penyisipan pesan rahasia. Gambar uji coba ditampilkan dalam nomor gambar dimana nomor 1 merupakan gambar dada, 2 merupakan gambar tangan, 3 merupakan gambar kaki, 4 merupakan gambar paru-paru dan nomor gambar 5 merupakan gambar leher.

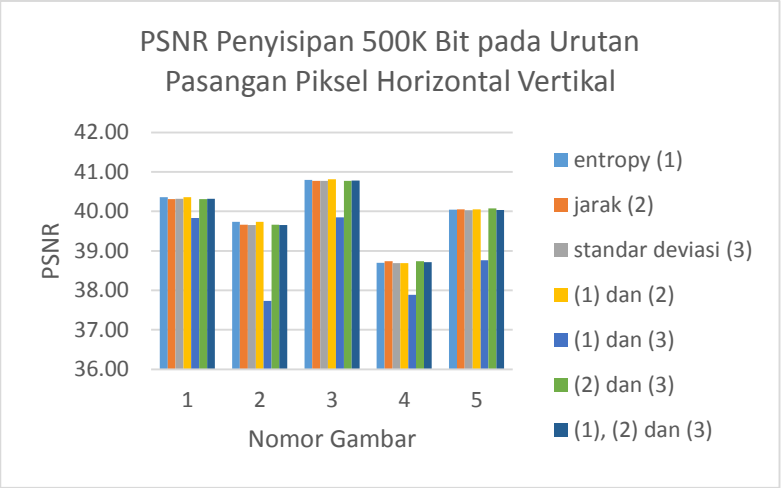
Grafik pada Gambar 5-7 sampai dengan Gambar 5-12 juga menunjukkan bahwa metode pemilihan urutan pasangan piksel dan pemilihan aturan kontrol *fuzzy* berpengaruh pada nilai PSNR citra *stego*. Hasil uji coba menunjukkan bahwa metode yang diusulkan penulis dapat mempertahankan nilai PSNR citra di atas 35 dB ketika dilakukan penyisipan hingga ukuran pesan 500 kbit.



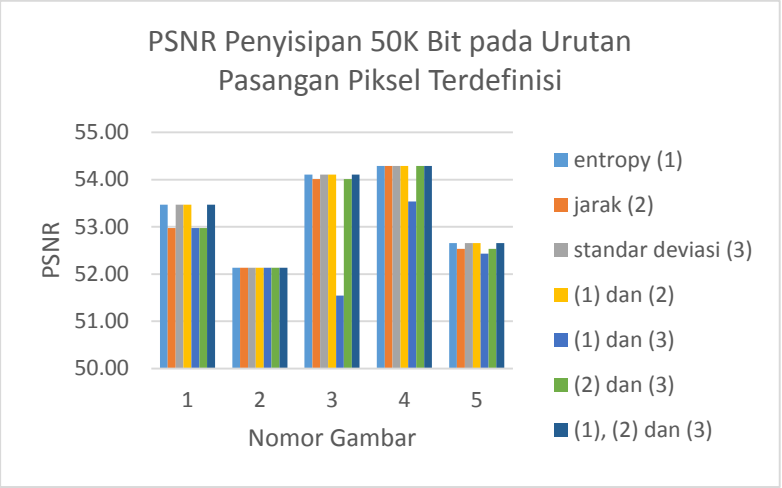
Gambar 5-7 Grafik PSNR Penyisipan 50kbit pada Urutan Pasangan Piksel Horizontal Vertikal



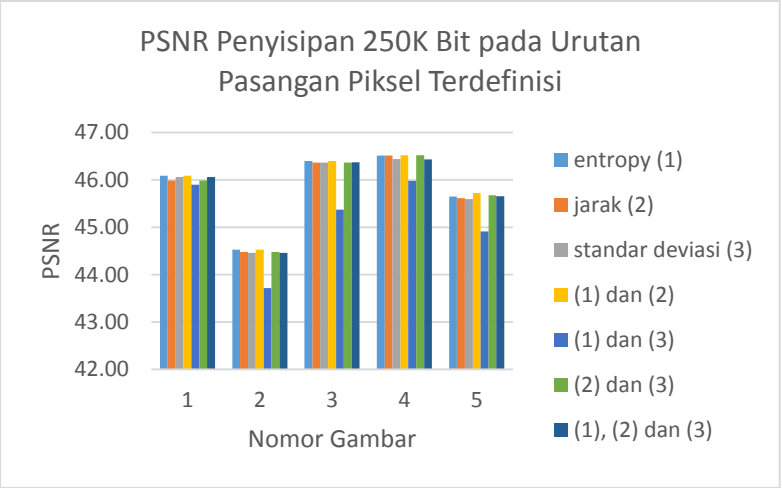
Gambar 5-8 Grafik PSNR Penyisipan 250 Kbit pada Urutan Pasangan Piksel Horizontal Vertikal



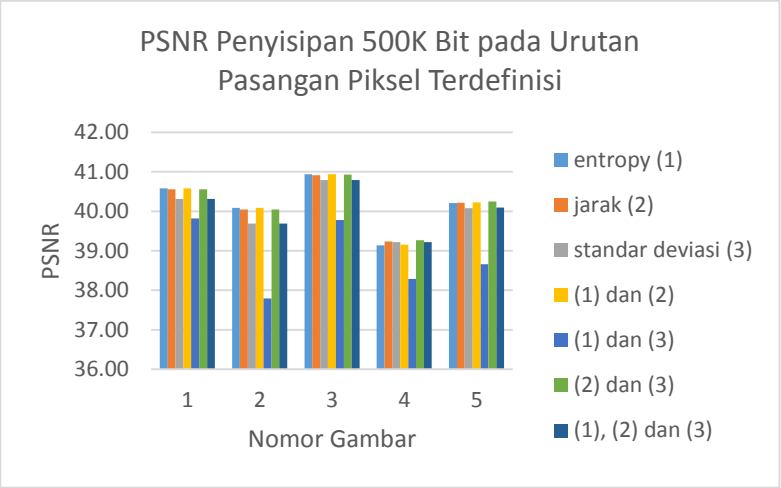
Gambar 5-9 Grafik PSNR Penyisipan 500 Kbit pada Urutan Pasangan Piksel Horizontal Vertikal



Gambar 5-10 Grafik PSNR Penyisipan 50 Kbit pada Urutan Pasangan Piksel Terdefinisi



Gambar 5-11 Grafik PSNR Penyisipan 250 Kbit pada Urutan Pasangan Piksel Terdefinisi



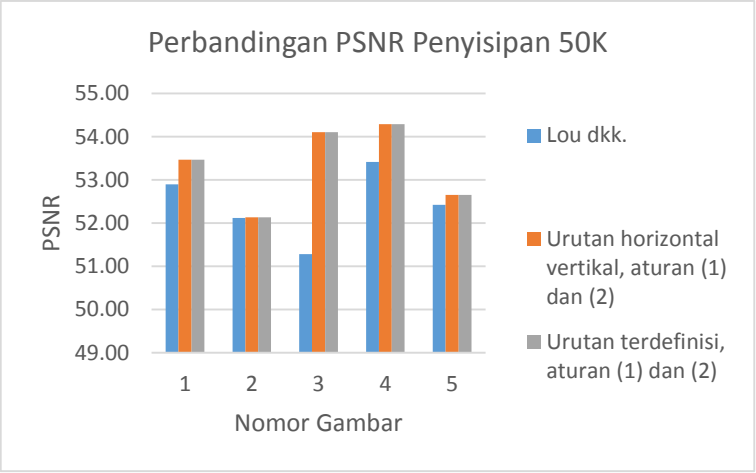
Gambar 5-12 Grafik PSNR Penyisipan 500 Kbit pada Urutan Pasangan Piksel Terdefinisi

Dari grafik pada Gambar 5-7 sampai dengan Gambar 5-9 didapatkan kesimpulan bahwa kombinasi aturan *fuzzy* yang menghasilkan nilai PSNR terbaik pada urutan pasangan piksel horizontal vertikal yaitu kombinasi antara *entropy* lokal dan jarak lokal dengan rata-rata PSNR yaitu 46.36 dB.

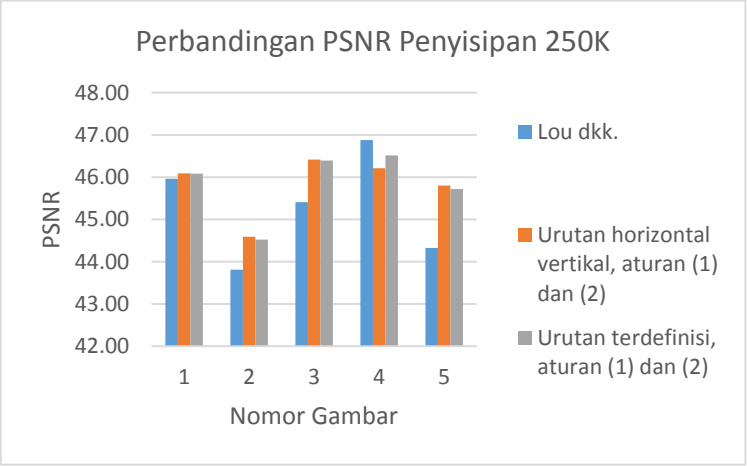
Dari dua hasil rata-rata PSNR tersebut, metode urutan pasangan piksel terdefinisi memiliki rata-rata PSNR lebih baik. Namun kedua metode tersebut tetap dibandingkan dengan metode Lou dkk. Perbanding metode urutan piksel horizontal vertikal menggunakan aturan *fuzzy entropy* lokal dan jarak lokal, metode urutan piksel menggunakan aturan *fuzzy entropy* lokal dan jarak lokal, dan metode Loud dkk. dapat dilihat pada Gambar 5-13 sampai dengan Gambar 5-15.

Hasil perbandingan PSNR yang ditunjukkan pada Gambar 5-13 sampai dengan Gambar 5-15, menunjukkan bahwa penggunaan logika *fuzzy* pada penyisipan RDE dapat meningkatkan nilai PSNR citra *stego* yang dihasilkan. Pada

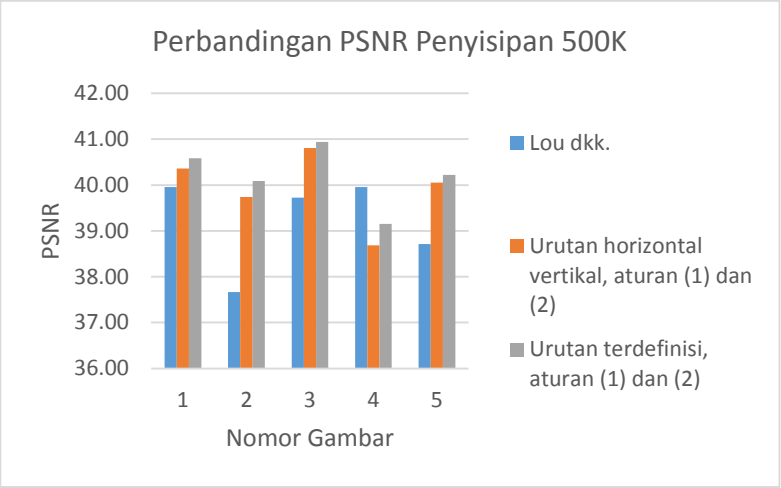
gambar uji coba 1, 2, 3 dan 5, metode yang diusulkan memiliki nilai PSNR lebih baik pada setiap ukuran data rahasia yang disisipkan.



Gambar 5-13 Perbandingan PSNR Penyisipan 50 kbit



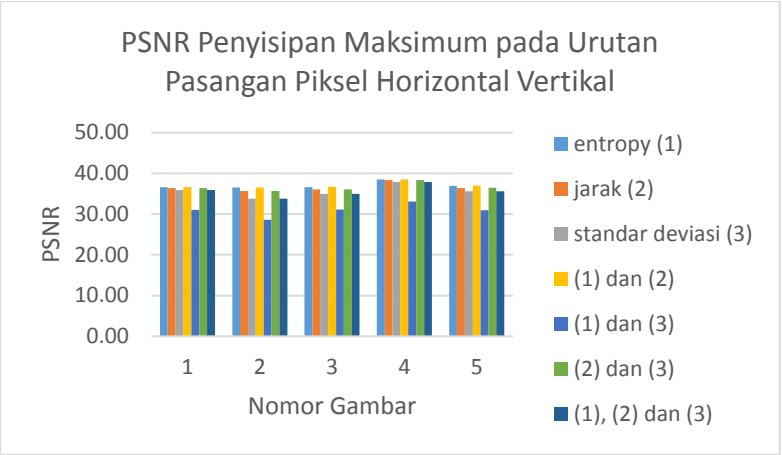
Gambar 5-14 Perbandingan PSNR Penyisipan 250 kbit



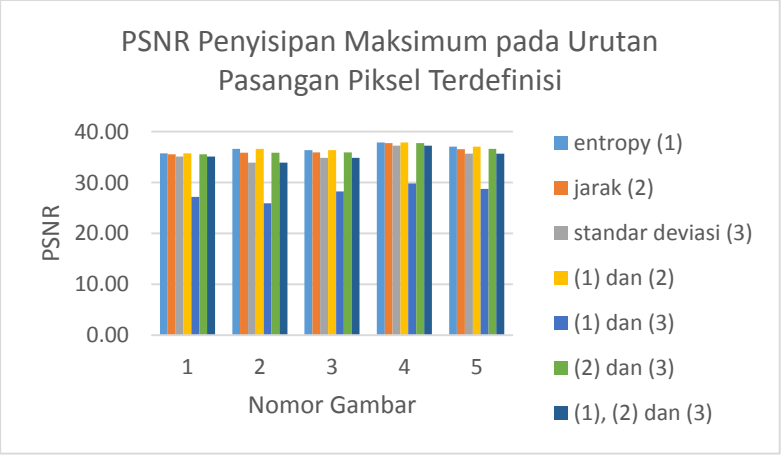
Gambar 5-15 Perbandingan PSNR Penyisipan 500 kbit

Uji coba juga dijalanan dengan cara melakukan penyisipan secara maksimal pada citra *cover* 1 sampai dengan citra *cover* 5. Grafik PSNR masing-masing metode yang diusulkan dapat dilihat pada Gambar 5-16 sampai dengan Gambar 5-17. Dari grafik tersebut dapat dilihat bahwa rata-rata nilai PSNR mencapai 35 dB. Jika dibandingkan dengan metode Lou dkk., metode yang diusulkan penulis mempertahankan nilai PSNR citra *stego* dengan lebih baik, seperti terlihat pada Gambar 5-18.

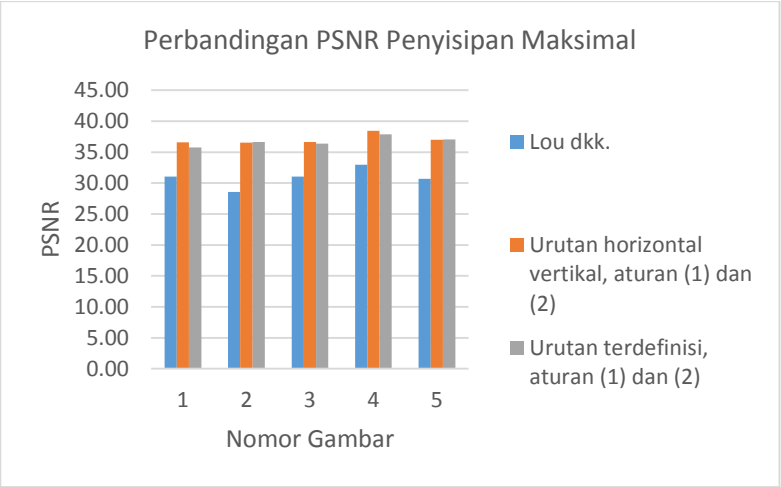
Nilai kapasitas maksimal dari masing-masing metode yang diusulkan terbilang cukup tinggi seperti terlihat pada Gambar 5-19 dan Gambar 5-20. Namun kapasitas metode yang diusulkan penulis lebih kecil dibandingkan dengan kapasitas yang dihasilkan oleh metode Lou dkk., seperti terlihat pada grafik Gambar 5-21. Hal ini disebabkan oleh metode yang diusulkan penulis melakukan seleksi terhadap pasangan piksel yang akan disisipi, sedangkan metode Lou dkk. melakukan penyisipkan pada semua pasangan piksel yang *expandable*.



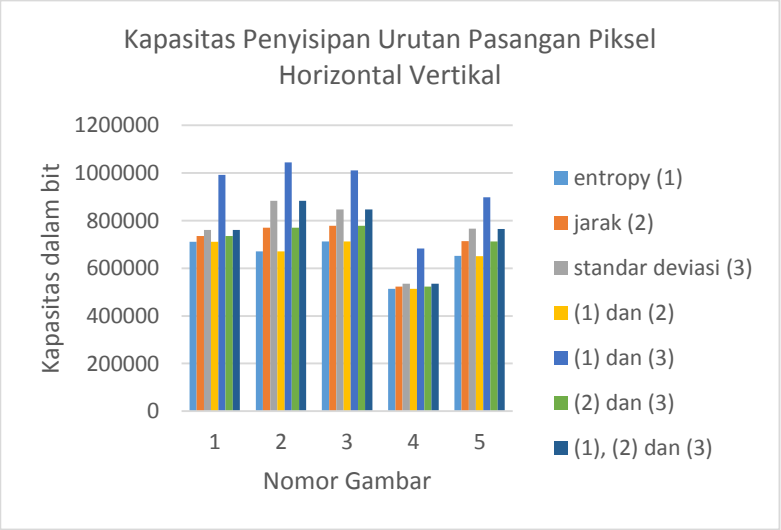
Gambar 5-16 Grafik PSNR Penyisipan Maksimal pada Urutan Pasangan Piksel Horizontal Vertikal



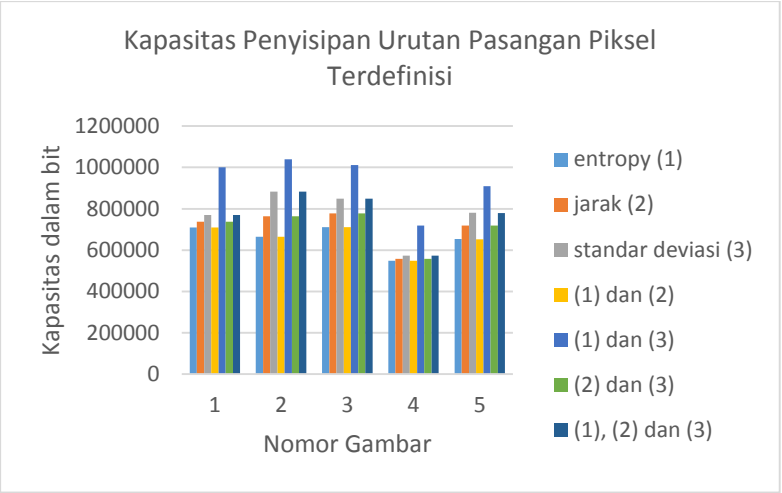
Gambar 5-17 Grafik PSNR Penyisipan Maksimal pada Urutan Pasangan Piksel Terdefinisi



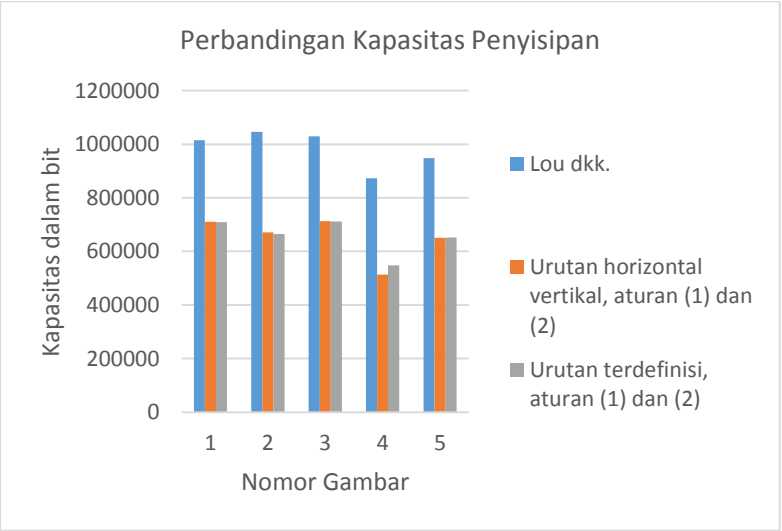
Gambar 5-18 Grafik Perbandingan PSNR Penyisipan Maksimal



Gambar 5-19 Kapasitas Penyisipan Urutan Pasangan Piksel Horizontal Vertikal



Gambar 5-20 Kapasitas Penyisipan Urutan Pasangan Piksel Terdefinisi



Gambar 5-21 Perbandingan Kapasitas Penyisipan

BAB VI
KESIMPULAN DAN SARAN

Pada bab ini dibahas mengenai kesimpulan yang dapat diambil dari perancangan metode hingga hasil uji coba dan evaluasi. Penjelasan pada bab ini juga akan menjawab permasalahan yang dikemukakan pada Bab 1. Dan terakhir terdapat beberapa saran untuk pengembangan selanjutnya.

6.1. Kesimpulan

Dari hasil pengamatan selama perancangan, pengimplementasian, dan proses uji coba metode dapat diambil kesimpulan sebagai berikut:

1. Logika *fuzzy* dapat digunakan dalam menentukan tingkat penyisipan dalam steganografi multi lapis.
2. Variabel yang digunakan dalam fungsi keanggotaan *fuzzy* dipengaruhi oleh karakteristik metode penyisipan. Jika menggunakan RDE sebagai metode penyisipan, maka variabel yang dapat digunakan antara lain nilai *entropy* lokal, jarak lokal, dan standar deviasi lokal dari sebuah blok piksel. Masing-masing kombinasi yang mungkin dari variabel tersebut menghasilkan hasil PSNR yang bervariasi. Pada bercobaan ini, kombinasi fungsi keanggotaan *entropy* lokal dan jarak lokal menghasilkan PSNR tertinggi.
3. Untuk merancang sebuah sistem inferensi *fuzzy* yang dapat meningkatkan PSNR dan *imperceptibility* dari citra *stego* dibutuhkan perancangan fungsi keanggotaan variabel *input* maupun *output* yang akurat. Fungsi keanggotaan dapat menentukan tingkat penyisipan atau *embedding level* sebuah pasangan piksel. Dengan adanya *embedding level*, pasangan piksel yang memiliki kemungkinan untuk menurunkan PSNR secara drastis tidak akan diproses, sedangkan sisanya memiliki tingkat penyisipan sesuai dengan karakteristik blok piksel masing-masin

4. Steganografi mengunakan RDE dan logika *fuzzy* yang diterapkan secara multi lapis memiliki rata-rata nilai PSNR lebih tinggi dibandingkan hasil metode RDE multi lapis yang diusulkan oleh Lou dkk., namun memiliki kapasitas total yang lebih rendah.

6.2. Saran

Saran yang dapat diberikan dalam pengembanganan metode ini yaitu:

1. Diperlukannya uji coba lebih lanjut mengenai kombinasi nilai fungsi keanggotaan dalam sebuah variabel untuk mencapai hasil optimal.
2. Pengujian kinerja metode lebih lanjut dapat dilakukan dengan cara mencoba metode ini pada studi kasus citra yang memiliki karakteristik berbeda dengan citra medis.

LAMPIRAN

Lampiran 1. Kumpulan Kode Sumber yang Digunakan

```
1.  function statusValid = embed(R, b)
2.      R.embeddedBit = b;
3.      R.statusValid = R.checkValid();
4.      if R.statusValid==1
5.          R.average = floor((R.x+R.y)/2);
6.          R.difference = R.x - R.y;
7.
8.          R.differenceReduced = R.difference;
9.          % Reduced difference
10.
11.         if(R.difference>=0)
12.             R.differenceReduced = R.difference -
    2^(floor(log2(abs(R.difference)))-1);
13.         else
14.             R.differenceReduced = R.difference +
    2^(floor(log2(abs(R.difference)))-1);
15.         end
16.
17.         if abs(R.difference)<2
18.             R.differenceReduced = R.difference;
19.         end
20.
21.         %locatioan map
22.         if
    (2^(floor(log2(abs(R.differenceReduced))))==2^(f
    loor(log2(abs(R.difference))))) ||
    (abs(R.differenceReduced)==abs(R.difference))
23.             R.locationMap = 0;
24.         else
25.             R.locationMap = 1;
26.         end
27.         %end location map
28.
29.         % end Reduced difference
30.
31.         if(R.differenceReduced>=0)
32.             R.differenceReduced2 = 2 *
    R.differenceReduced + R.embeddedBit;
```

```
33.         else
34.             R.differenceReduced2 = 2 *
    R.differenceReduced - R.embeddedBit;
35.         end
36.
37.         R.statusValid =
    R.checkOverflowUnderflow();
38.         if R.statusValid==1
39.             R.x = R.average +
    floor((R.differenceReduced2 + 1)/2);
40.             R.y = R.average -
    floor(R.differenceReduced2/2);
41.         end
42.
43.         if(R.x<0) || (R.x>255)
44.             R.statusValid = 0;
45.         end
46.         if(R.y<0) || (R.y>255)
47.             R.statusValid = 0;
48.         end
49.     end
50.
51.     statusValid = R.statusValid;
52.
53. end
```

Kode Sumber 7-1 Implementasi Penyisipan Bit dengan Metode RDE

```
1.  function statusValid = extract(R, h)
2.      R.locationMap = h;
3.
4.      R.statusValid = R.checkValid();
5.      if R.statusValid==1
6.          R.average = floor((R.x+R.y)/2);
7.          R.differenceReduced2 = R.x - R.y;
8.
9.          R.embeddedBit =
    mod(abs(R.differenceReduced2), 2);
10.         if(R.differenceReduced2>=0)
11.             R.differenceReduced =
    floor(R.differenceReduced2/2);
12.         else
```

```
13.             R.differenceReduced =
    ceil(R.differenceReduced2/2);
14.         end
15.
16.         R.difference = R.differenceReduced;
17.
18.         % recovery based on RDE
19.         if R.locationMap==0
20.             if(R.differenceReduced>=0)
21.                 if(R.differenceReduced==1)
22.                     R.difference =
    R.differenceReduced;
23.                 else
24.                     R.difference =
    R.differenceReduced +
    2^(floor(log2(abs(R.differenceReduced)))-1);
25.                 end
26.             else
27.                 if(R.differenceReduced==-1)
28.                     R.difference =
    R.differenceReduced;
29.                 else
30.                     R.difference =
    R.differenceReduced -
    2^(floor(log2(abs(R.differenceReduced)))-1);
31.                 end
32.             end
33.         elseif R.locationMap==1
34.             if(R.differenceReduced>=0)
35.                 if(R.differenceReduced==1)
36.                     R.difference =
    R.differenceReduced;
37.                 else
38.                     R.difference =
    R.differenceReduced +
    2^(floor(log2(abs(R.differenceReduced))));
39.                 end
40.             else
41.                 if(R.differenceReduced==-1)
42.                     R.difference =
    R.differenceReduced;
43.                 else
```

```
44.                     R.difference =
   R.differenceReduced -
   2^(floor(log2(abs(R.differenceReduced))));
45.                 end
46.             end
47.         end
48.         % end recovery based on RDE
49.
50.         R.x = R.average + floor((R.difference +
   1)/2);
51.         R.y = R.average - floor(R.difference/2);
52.
53.         if((R.x-R.y)==1) && (R.locationMap==1)
54.             R.y=R.y-1;
55.         elseif((R.x-R.y)==-1) &&
   (R.locationMap==1)
56.             R.x=R.x-1;
57.         end
58.     end
59.
60.     if(R.x<0) || (R.x>255)
61.         R.statusValid = 0;
62.     end
63.     if(R.y<0) || (R.y>255)
64.         R.statusValid = 0;
65.     end
66.
67.     statusValid = R.statusValid;
68.
69. end
```

Kode Sumber 7-2 Implementasi Ekstraksi Bit dengan Metode RDE

```
1.  % proses penyisipan bit ke blok-blok yang ada
   pada citra
2.  function embedToBlock(I, messageString)
3.      % mengubah string ke bit
4.      I.message = messageString;
5.      messageBinary =
   reshape(dec2bin(I.message,8)',1,[]);
6.
```

```
7.      for j=1:size(messageBinary,2)
8.          I.listInputBit(1,j) =
   str2num(messageBinary(1,j));
9.      end
10.     I.bitLong = size(I.listInputBit,2);
11.     % akhir mengubah string ke bit
12.
13.     levelLayer = getLevelMax(I.listLocationMap);
14.     listBlockExpandable =
   getBlockExpandable(I.listLocationMap);
15.     resetLevelPointer(I.listLocationMap);
16.
17.     indexInputBit=1;
18.     I.statusBitLong=1;
19.
20.     tic;
21.     % wait bar/loading bar embedding process
22.     totalStep =
   (I.imageSize*I.imageSize)/(I.blockSize*I.blockSi
   ze)*levelLayer;
23.     step = 0;
24.     loadingBar = waitbar(0,'Embedding Process
   ...');
25.
26.     % penyisipan dilakukan/diulang sebanyak
   level teringgi
27.     for k = 1 : levelLayer
28.         listLocation = zeros(1,
   (I.imageSize*I.imageSize)/2);
29.         listBlockEmbed = zeros(1,
   (I.imageSize*I.imageSize)/(I.blockSize*I.blockSi
   ze));
30.         listBlockFailedtoMax = zeros(1,
   (I.imageSize*I.imageSize)/(I.blockSize*I.blockSi
   ze));
31.         indexLocation= 1;
32.         indexListBlockEmbed = 1;
33.         indexlistBlockFailedtoMaxtoMax = 1;
34.         indexListBlockExpandable = 1;
35.
36.         statusImage = 1;
37.         currentRow = 1;
38.         currentColumn = 1;
39.         nextRow = currentRow;
```

```
40.         nextColumn = currentColumn;
41.
42.         while (statusImage==1) &&
    (I.statusBitLong==1)
43.             newMatrix = [];
44.             newRow = 1;
45.             newColumn = 1;
46.
47.             % menyisir blok satu per satu
48.             for i=1:I.blockSize
49.                 for j=1:I.blockSize
50.                     newMatrix(newRow,newColumn)
    = I.newImageMatrix(nextRow,nextColumn);
51.
52.                     nextColumn = nextColumn+1;
53.                     newColumn = newColumn + 1;
54.                 end
55.                 nextColumn = currentColumn;
56.                 nextRow = nextRow + 1;
57.
58.                 newColumn = 1;
59.                 newRow = newRow+1;
60.             end
61.
62.             % jika blok expandable maka
    dilakukan penyisipan bit
63.             if
    listBlockExpandable(k,indexListBlockExpandable)=
    =1
64.                 newPixelBlock =
    PixelBlock(2,newMatrix,I.mode, I.fisMode);
65.                 I.statusEmbed =
    embed(newPixelBlock, k,
    I.listInputBit(indexInputBit:indexInputBit+1));
66.
67.                 % jika salah satu piksel gagal
    diembed, maka status
68.                 % blok tersebut pada level
    tersebut dicatat tidak
69.                 % dilakukan penyisipan
70.                 if(I.statusEmbed == 0)
71.
    listBlockFailedtoMax(indexlistBlockFailedtoMaxto
    Max) = 1;
```

```
72.                 end
73.             else
74.                 I.statusEmbed = 0;
75.             end
76.
77.             % menyimpan perubahan pixel ke dalam
   matrix gambar baru
78.             if I.statusEmbed==1
79.                 [tempNewBlock,
   tempListLocationMap] = newPixelBlock.getBlock();
80.
81.                 nextRow = currentRow;
82.                 nextColumn = currentColumn;
83.                 newRow = 1;
84.                 newColumn = 1;
85.
86.                 for i=1:I.blockSize
87.                     for j=1:I.blockSize
88.
   I.newImageMatrix(nextRow,nextColumn) =
   tempNewBlock(newRow,newColumn);
89.
90.                         nextColumn =
   nextColumn+1;
91.                         newColumn = newColumn +
   1;
92.                     end
93.                     nextColumn = currentColumn;
94.                     nextRow = nextRow + 1;
95.
96.                     newColumn = 1;
97.                     newRow = newRow+1;
98.                 end
99.                 % akhir menyimpan perubahan
   pixel ke dalam matrix gambar baru
100.
101.                    indexInputBit =
   indexInputBit+2;
102.                    listLocation(indexLocation) =
   tempListLocationMap(1);
103.                    listLocation(indexLocation+1)
   = tempListLocationMap(2);
104.
   listBlockEmbed(indexListBlockEmbed) = 1;
```

```
105.                end
106.
107.                indexLocation = indexLocation +
    2;
108.                indexListBlockEmbed =
    indexListBlockEmbed+1;
109.                indexListBlockExpandable =
    indexListBlockExpandable+1;
110.                indexlistBlockFailedtoMaxtoMax =
    indexlistBlockFailedtoMaxtoMax+1;
111.
112.                if
    (currentColumn+I.blockSize)<I.imageSize
113.                    currentColumn = currentColumn
    + I.blockSize;
114.                else
115.                    currentColumn = 1;
116.
117.                    if
    (currentRow+I.blockSize)<I.imageSize
118.                        currentRow = currentRow +
    I.blockSize;
119.                    else
120.                        statusImage=0;
121.                    end
122.                end
123.
124.                nextColumn = currentColumn;
125.                nextRow = currentRow;
126.
127.                if (indexInputBit>I.bitLong)
128.                    I.statusBitLong = 0;
129.                end
130.
131.                waitbar(step / totalStep,
    loadingBar, sprintf('Embedding Process %d /
    %d',step, totalStep))
132.                step = step+1;
133.            end
134.
135.            addLocationMap(I.listLocationMap,
    listLocation, listBlockEmbed,
    listBlockFailedtoMax);
136.        end
```

```
137.       close(loadingBar)
138.       I.timeEmbedding = toc;
139.   end
```

Kode Sumber 7-3 Implementasi Penyisipan Pesan Rahasia ke Dalam Blok-Blok pada Citra

```
1.  function extractPerBlock(I, locationMap)
2.      I.listLocationMap = locationMap;
3.
4.      listBlockEmbed =
    getListBlockEmbed(I.listLocationMap);
5.      levelLayer =
    getExtractLevelMax(I.listLocationMap);
6.      resetLevelPointer(I.listLocationMap);
7.
8.      tic;
9.      % wait bar/loading bar embedding process
10.     totalStep =
    (I.imageSize*I.imageSize)/(I.blockSize*I.blockSi
    ze)*levelLayer;
11.     step = 0;
12.     loadingBar = waitbar(0,'Extracting Process
    ...');
13.
14.     for k = 1 : levelLayer
15.         [listLocation, listBlockEmbed] =
    getLocationMap(I.listLocationMap, (levelLayer-
    k+1));
16.
17.         indexLocation= 1;
18.         indexListBlockEmbed = 1;
19.
20.         statusImage = 1;
21.         currentRow = 1;
22.         currentColumn = 1;
23.         nextRow = currentRow;
24.         nextColumn = currentColumn;
25.
26.         indexInputBit=1;
27.         tempListInputBit=[];
28.
29.         while statusImage==1
```

```
30.             newMatrix = [];
31.             newRow = 1;
32.             newColumn = 1;
33.
34.             for i=1:I.blockSize
35.                 for j=1:I.blockSize
36.                     newMatrix(newRow,newColumn)
    = I.newImageMatrix(nextRow,nextColumn);
37.
38.                     nextColumn = nextColumn+1;
39.                     newColumn = newColumn + 1;
40.                 end
41.                 nextColumn = currentColumn;
42.                 nextRow = nextRow + 1;
43.
44.                 newColumn = 1;
45.                 newRow = newRow+1;
46.             end
47.
48.             if
    listBlockEmbed(indexListBlockEmbed)==1
49.                 newPixelBlock =
    PixelBlock(2,newMatrix,I.mode, I.fisMode);
50.
51.                 statusExtract =
    extract(newPixelBlock, levelLayer-k+1,
    listLocation(indexLocation:indexLocation+1));
52.             else
53.                 statusExtract = 0;
54.             end
55.
56.
57.             % menyimpan perubahan pixel ke dalam
    matrix gambar baru
58.             if statusExtract==1
59.                 [tempNewBlock, newListInputBit]
    = newPixelBlock.getExtractBlock();
60.
61.                 nextRow = currentRow;
62.                 nextColumn = currentColumn;
63.                 newRow = 1;
64.                 newColumn = 1;
65.
66.                 for i=1:I.blockSize
```

```
67.                     for j=1:I.blockSize
68.
   I.newImageMatrix(nextRow,nextColumn) =
   tempNewBlock(newRow,newColumn);
69.
70.                         nextColumn =
   nextColumn+1;
71.                         newColumn = newColumn +
   1;
72.                     end
73.                     nextColumn = currentColumn;
74.                     nextRow = nextRow + 1;
75.
76.                     newColumn = 1;
77.                     newRow = newRow+1;
78.                 end
79.                 % akhir menyimpan perubahan
   pixel ke dalam matrix gambar baru
80.
81.                 tempListInputBit(indexInputBit)
   = newListInputBit(1);
82.
   tempListInputBit(indexInputBit+1) =
   newListInputBit(2);
83.
84.                 indexInputBit = indexInputBit +
   2;
85.             end
86.
87.             if
   (currentColumn+I.blockSize)<I.imageSize
88.                 currentColumn = currentColumn +
   I.blockSize;
89.             else
90.                 currentColumn = 1;
91.
92.                 if
   (currentRow+I.blockSize)<I.imageSize
93.                     currentRow = currentRow +
   I.blockSize;
94.                 else
95.                     statusImage=0;
96.                 end
97.             end
```

```
98.
99.             nextColumn = currentColumn;
100.               nextRow = currentRow;
101.               indexListBlockEmbed =
   indexListBlockEmbed + 1;
102.               indexLocation = indexLocation+2;
103.
104.               waitbar(step / totalStep,
   loadingBar, sprintf('Extracting Process %d /
   %d',step, totalStep))
105.               step = step+1;
106.           end
107.
108.           I.listInputBit =
   cat(2,tempListInputBit,I.listInputBit);
109.       end
110.       close(loadingBar)
111.       I.timeExtracting = toc;
112.
113.       n = size(I.listInputBit);
114.       n = n(2);
115.       outputCharacter = [];
116.       i = 1;
117.       j = 8;
118.       k = 1;
119.       s = '';
120.
121.       while i+7 <= n
122.           % ambil per 8 bit
123.           input8Bit = I.listInputBit(i:j);
124.           str_x = num2str(input8Bit);
125.           str_x(isspace(str_x)) = '';
126.           asciiNumber = bin2dec(str_x);
127.
128.           character = char(asciiNumber);
129.           outputCharacter(k)=character;
130.
131.           i = i+8;
132.           j = j+8;
133.           k = k+1;
134.       end
135.       s = strcat(outputCharacter);
136.       I.message = s;
137.
```

```
138.
139.       CLI.newImageMatrix =
   uint8(I.newImageMatrix);
140.
   calculateBitSizeandCapacity(I.listLocationMap);
141.       disp(I.message);
142.   end
```

Kode Sumber 7-4 Implementasi Ekstraksi Pesan Rahasia ke Dalam Blok-Blok pada Citra

Lampiran 2. Perbandingan Metode yang Diusulkan

Tabel 7-1 Perbandingan Metode yang Diusulkan dengan yang Sudah Ada Sebelumnya

No	**Kategori**	**Metode yang Diusulkan**	**Multiple Layer Data Hiding Scheme for medical images** [3]	**A High Quality Image Steganography Scheme Based on Fuzzy Inference System** [5]
1	Metode steganografi	RDE	RDE	LSB
2	*Reversible*	Ya	Ya	Tidak
3	Pemrosesan per blok	Blok berukuran 2×2 piksel.	Tidak menggunakan blok	Blok berukuran 3×3 piksel.
4	Adaptif	Ya	Tidak	Ya
5	Sistem inferensi *fuzzy*	Digunakan untuk menentukan level penyisipan dengan	Tidak menggunakan sistem inferensi *fuzzy*	Digunakan untuk menentukan jumlah bit yang dirubah

		menggukan variabel *entropy* lokal, jarak lokal dan standar deviasi lokal dari sebuah blok		dalam satu piksel dengan variabel tekstur, sensitivitas tepi dan tingkat kecerahan dari sebuah blok
6	Multi lapis	Ya	Ya	Tidak
7	Urutan pasangan piksel per lapis	Terdapat beberapa metode urutan yaitu : • Pada lapis ganjil pasangan piksel yang diambil secara horizontal, sedangkan pada lapis genap secara vertikal. • Terdefinisi, dimana setiap lapis memiliki urutan pasangan piksel yang unik.	Pada lapis ganjil pasangan piksel yang diambil secara horizontal, sedangkan pada lapis genap secara vertikal.	Tidak

DAFTAR PUSTAKA

[1] M. H. W. W. R. M. Tohari Ahmad, "An Improved Quad and RDE-Based Medical Data Hiding Method," dalam *Computational Intelligence and Cybernetics (CYBERNETICSCOM)*, Yogyakarta , 2013.

[2] J. Tian, "Reversible data embedding using a difference expansion," *Circuits and Systems for Video Technology, IEEE Transactions,* vol. 13, no. 8, pp. 890 - 896 , 2003.

[3] M.-C. H. J.-L. L. Der-Chyuan Lou, "Multiple Layer Data Hiding Scheme for medical images," *Computer Standards & Interfaces,* vol. 31, pp. 329-335, 2009.

[4] A. Z. A. S. S. Mahdi Hassani Goodarzi, "Convergence between Fuzzy Logic and Steganography for High Payload Data Embedding and More Security," dalam *The 6th International Conference on Telecomunication Systems, Services, and Applications*, Bali , 2011.

[5] A. M. E. M. Sara Sajasi, "A High Quality Image Steganography Scheme Based on Fuzzy Inference System," dalam *Fuzzy Systems (IFSC) 13th Iranian Conference*, Qazvin, 2013.

[6] D. Putra, Pengolahan Citra Digital, Yogyakarta: ANDI, 2010.

[7] R. C. Gonzalez dan R. E. Woods, Digital Image Processing, New Jersey: Prentice Hall, 2002.

[8] S. AnnaDurai dan R. Shanmugalakshmi, Fundamentals of Digital Image Processing, Delhi, India: Dorling Kindersley, 2007.

[9] P. H. Niels Provos, "Hide and Seek: An Introduction to Steganography," *Security & Privacy, IEEE,* vol. 1, no. 3, pp. 32-44, 2003.

[10] Suparman, Mengenal Artificial Intelligence, Yogyakarta: Andi Offset, 1991.

[11] S. Kusumadewi, Artificial Intelligence (Teknik danAplikasinya), Yogyakarta: Graha Ilmu, 2002.

[12] Y. Bai, H. Zhuang dan D. Wang, Advanced Fuzzy Logic Technologies in Industrial, Springer, 2006.

[13] E. Irwansyah dan M. Faisal, Advanced Clustering: Teori dan Aplikasi, DeePublish, 2015.

[14] MATLAB, "MATLAB - The Language of Technical Computing," MATLAB, [Online]. Available: http://www.mathworks.com/products/matlab/. [Diakses 13 6 2015].

[15] "Partners Infectious Disease Images eMicrobes Digital Library," Partners Healthcare, [Online]. Available: http://www.idimages.org/images/browse/ImageTechnique/. [Diakses 1 6 2015].

[16] "GitHub Glutanimate/ wordlist-medicalterms-en," [Online]. Available: https://github.com/Glutanimate/wordlist-medicalterms-en. [Diakses 1 6 2015].

BIODATA PENULIS

Putu Harum Bawa, merupakan anak pertama dari tiga bersaudara. Penulis dilahirkan di Kedewatan, Ubud pada 8 September 1993. Penulis telah menempuh pendidikan formal di TK Dewata Kumara I (1998-1999), SD Negeri 7 Batubulan (1999-2005), SMP Negeri 1 Sukawati (2005-2008), SMA Negeri 3 Denpasar (2008-2011), dan pada tahun 2011 diterima di S1 Jurusan Teknik Informatika.

Selama kuliah penulis pernah menjadi asisten beberapa mata kuliah, diantaranya Sistem Digital, Sistem Operasi, Organisasi Komputer, Jaringan Komputer, serta Keamanan Informasi dan Jaringan. Selain itu, penulis juga aktif sebagai asisten laboratorim di Laboratorium Arsitektur dan Jaringan Komputer. Penulis juga pernah terlibat dalam beberapa proyek yang diadakan oleh institut.

Selain aktif dalam kegiatan akademis, penulis juga aktif dalam kegiatan kemahasiswaan, diantaranya aktif di Tim Pembina Kerohanian Hindu ITS, Himpunan Mahasiswa Teknik Computer Informatika ITS, dan kepemanduan FTIF.

Ketertarikan penulis di bidang informatika antara lain pada bidang jaringan dan sekuritas jaringan. Ketertarikan penulis tidak hanya pada bidang akademis, penulis juga merupakan penggemar fotografi.

Penulis dapat dihubungi dengan mengirimkan pesan elektronik ke alamat lightharum.108@gmail.com.