



TUGAS AKHIR - KI141502

Modifikasi Metode *Histogram Shifting* dan *Neighbouring Similarity* untuk Steganografi pada Video Digital

Alek Nur Fatman
NRP 0511144000018

Dosen Pembimbing I
Tohari Ahmad, S.Kom., MIT., Ph.D.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

**Modifikasi Metode *Histogram Shifting* dan
Neighbouring Similarity untuk Steganografi
pada Video Digital**

**Alek Nur Fatman
NRP 0511144000018**

**Dosen Pembimbing I
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESES - KI141502

Modified Histogram Shifting and Neighbouring Similarity Methods for Steganography on Digital Video

**Alek Nur Fatman
NRP 0511144000018**

**First Advisor
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Department of Informatics
Faculty of Information Technology and Communication
Sepuluh Nopember Institute of Technology
Surabaya 2018**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

**Modifikasi Metode *Histogram Shifting* dan
Neighbouring Similarity untuk Steganografi
pada Video Digital**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

Alek Nur Fatman
NRP: 0511144000018

Disetujui oleh Pembimbing Tugas Akhir

1. Tohari Ahmad, S.Kom., M.T., Ph.D.
(NIP. 197505252003121002)
(Pembimbing 1)

SURABAYA
JUNI, 2018



[Halaman ini sengaja dikosongkan]

Modifikasi Metode *Histogram Shifting* dan *Neighbouring Similarity* untuk Steganografi pada Video Digital

Nama Mahasiswa : Alek Nur Fatman
NRP : 0511144000018
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1 : Tohari Ahmad, S.Kom., MIT., Ph.D.

Abstrak

Steganography merupakan salah satu metode yang digunakan untuk menyembunyikan informasi. Dengan *steganography*, informasi akan disembunyikan dengan cara menyisipkan informasi tersebut kedalam data lain. Contoh dari *steganography* adalah *histrogram shifting* dan *neighbouring similarity*. *Histogram shifting* dan *neighbouring similarity* memiliki sifat *reversible*, yang berarti video hasil penyisipan dapat dikembalikan lagi seperti semula.

Pada Tugas Akhir ini dilakukan pengembangan metode *histrogram shifting* dan *neighbouring similarity* dengan cara menggabungkan kedua metode tersebut serta melakukan modifikasi pada arah pergeseran bit dan letak penyisipan. Dari data hasil uji coba, jumlah *payload* yang bisa dimasukkan meningkat sebanyak 88% dari metode *neighbouring similarity* dan meningkat sebanyak 1658% dari metode *histrogram shifting*. Nilai PSNR juga meningkat sebanyak 2,1% dari metode *neighbouring similarity* dan meningkat sebanyak 18,01% dari metode *histrogram shifting*.

Kata kunci: *Steganography*, *Data hiding*, *Histogram shifting*, *Neighbouring similarity*.

[Halaman ini sengaja dikosongkan]

Modified Histogram Shifting and Neighbouring Similarity Methods for Steganography on Digital Video

Student's Name : Alek Nur Fatman
Student's ID : 05111440000018
Department : Informatika FTIK-ITS
First Advisor : Tohari Ahmad, S.Kom., MIT., Ph.D.

Abstract

Steganography is one of the methods used to hide information. With steganography, the information will be hidden by inserting that information into other data. An example of steganography is a histogram shifting and neighbouring similarity. Histogram shifting and neighbouring similarity have reversible properties, that mean the video result of embedding can be restored as same as before.

In this Final Project, histogram shifting and neighbouring similarity is modified by combining both methods and modifying the direction and the location of the shifting. From the data of the test result, the number of payloads that can be entered increases as much as 88% of the neighbouring similarity method and increases as much as 1658% of the histogram shifting method. The PSNR score also increased by 2.1% from the neighbouring similarity method and increased by 18.01% of the histogram shifting method.

Keywords: Steganography, Data hiding, Histogram shifting, Neighbouring similarity.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat, tauhid dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Modifikasi Metode *Histogram Shifting* dan *Neighbor Similarity* untuk Steganografi pada Video Digital**”.

Penulis berharap apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan, serta dapat memberikan kontribusi yang nyata baik sekarang maupun selanjutnya.

Selama pelaksanaan pembuatan hingga selesainya Tugas Akhir ini, penulis tidak terlepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Keluarga penulis baik ayah, ibu, adik, paman, bibi, nenek dan keluarga penulis lain yang selalu memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. selaku dosen pembimbing penulis yang telah membimbing dan memberikan motivasi dan nasehat selama penulis mengerjakan tugas akhir ini.
4. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Departemen Informatika ITS.
5. Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. selaku koordinator Tugas Akhir di Departemen Informatika ITS.
6. Seluruh dosen dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di ITS.
7. Teman-teman dari kontrakan KEBO(Ilyas, Brigas, Awer, Hanif, Adin, Udin, Mbahe) yang selalu memberikan

semangat serta selalu menghibur penulis, teman-teman yang sering bermain game bersama, teman-teman yang bisa diajak untuk bertukar pikiran dan pendapat, dan teman yang menjadi keluarga baru penulis saat berkuliah di Departemen Informatika ITS.

8. Teman-teman dari laboratorium KBJ yang telah menemani dan memberi kemudahan dalam peminjaman komputer selama pengerjaan Tugas Akhir ini.
9. Sahabat penulis sesama Topik(Hanif, Jeffry, Ibnu) yang telah membantu, menemani, berjuang bersama dan telah menjadi tempat pertukar ilmu dengan penulis saat pengerjaan Tugas Akhir.
10. Teman-teman seperjuangan RMK NCC/KBJ, yang telah menemani dan menyemangati penulis.
11. Teman-Teman Informatika 2014 yang telah menghibur, menemani dan membantu selama penulis berkuliah di Departemen Informatika ITS.
12. Teman-Teman lain yang tak bisa disebut satu per satu yang telah mendukung dan membantu penulis.

Penulis telah berusaha mengerjakan Tugas Akhir ini dengan sebaik-baiknya, apabila masih ada kesalahan atau kelalaian penulis mohon maaf. Penulis mengharapkan kritik dan saran dari pembaca untuk membantu membuat Tugas Akhir ini menjadi lebih baik kedepannya.

Surabaya, 5 Juni 2018

Alek Nur Fatman

DAFTAR ISI

Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Permasalahan	2
1.4 Tujuan	2
1.5 Manfaat.....	3
1.6 Metodologi	3
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur	3
1.6.3 Implementasi Sistem.....	4
1.6.4 Pengujian dan Evaluasi.....	4
1.6.5 Penyusunan Buku	4
1.7 Sistematika Penulisan Laporan	4
BAB II TINJAUAN PUSTAKA	7
2.1 Video Digital	7
2.2 <i>Steganography</i>	7
2.3 <i>Histogram Shifting</i>	8
2.4 <i>Neighbouring Similarity</i>	8
2.5 MATLAB	8
2.6 <i>Peak to Signal Noise Ratio (PSNR)</i>	9
2.7 <i>Correlation</i>	9
BAB III PERANCANGAN	11
3.1 Deskripsi Umum	11
3.2 Perancangan Data	11
3.2.1 Data Masukan	12
3.2.2 Data Keluaran	13
3.3 Perancangan Modifikasi Metode.....	13

3.3.1	Proses Penyisipan	13
3.3.1.1	Penghitungan Frekuensi	13
3.3.1.2	Pembangkitan <i>Prediction Error</i>	14
3.3.1.3	Pemilihan Arah Pergeseran	15
3.3.1.4	Shifting(Pergeseran).....	16
3.3.1.5	Penyisipan Payload.....	17
3.3.1.6	Pencegahan <i>Overflow</i> dan <i>Underflow</i>	17
3.3.2	Proses Ekstraksi	18
3.3.2.1	Pembacaan <i>Location Map</i>	19
3.3.2.2	Shifting(Pergeseran).....	19
3.3.2.3	Pengembalian <i>Payload</i>	20
BAB IV IMPLEMENTASI.....		21
4.1	Modul Penyisipan.....	21
4.1.1	Fungsi Pemindaian.....	21
4.1.2	Fungsi Perhitungan Frekuensi	21
4.1.3	Fungsi Pembangkitan <i>Prediction Error</i>	23
4.1.4	Fungsi Pemilihan Arah Pergeseran.....	24
4.1.5	Fungsi Pergeseran	25
4.1.6	Fungsi Penyisipan <i>Payload</i>	25
4.1.7	Fungsi Pencegahan <i>Overflow</i> dan <i>Underflow</i>	26
4.1.8	Fungsi Penyimpanan <i>Location Map</i>	28
4.2	Modul Ekstraksi	28
4.2.1	Fungsi Pemindaian.....	28
4.2.2	Fungsi Pergeseran	29
4.2.3	Fungsi Pengembalian <i>Payload</i>	31
4.3	Modul Analisis	32
4.3.1	Fungsi PSNR	32
4.3.2	Fungsi Korelasi.....	32
BAB V HASIL UJI COBA DAN EVALUASI		33
5.1	Lingkungan Uji Coba.....	33
5.2	Hasil Uji Coba.....	33
5.2.1	Hasil Uji Coba 1	34
5.2.2	Hasil Uji Coba 2	37
5.2.3	Hasil Uji Coba 3	38
BAB VI KESIMPULAN DAN SARAN		43

6.1 Kesimpulan.....	43
6.2 Saran.....	43
DAFTAR PUSTAKA	45
BIODATA PENULIS.....	47

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Penyisipan Data.....	14
Gambar 3.2 Diagram Alir Ekstraksi Data	19
Gambar 4.1 Potongan Program Pemindaian Penyisipan	21
Gambar 4.2 Potongan Program Perhitungan Frekuensi	23
Gambar 4.3 Potongan Program Pembangkitan <i>Prediction Error</i>	24
Gambar 4.4 Potongan Program Pemilihan Arah Pergeseran.....	25
Gambar 4.5 Potongan Program Pergeseran.....	25
Gambar 4.6 Potongan Program Penyisipan <i>Payload</i>	26
Gambar 4.7 Potongan Program Pencegahan <i>Overflow</i> dan <i>Underflow</i>	28
Gambar 4.8 Potongan Program Penyimpanan <i>Location Map</i>	28
Gambar 4.9 Potongan Program Pemindaian Ekstraksi.....	29
Gambar 4.10 Potongan Program Ekstraksi	30
Gambar 4.11 Potongan Program Pengembalian <i>Payload</i>	32
Gambar 4.12 Potongan Program Fungsi PSNR	32
Gambar 4.13 Potongan Program Fungsi Korelasi	32

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Cover Video	12
Tabel 3.2 Payload	12
Tabel 3.3 Pencegahan <i>Overflow</i> dan <i>Underflow</i>	18
Tabel 5.1 Spesifikasi Perangkat yang Digunakan	33
Tabel 5.2 Hasil PSNR dengan <i>Payload</i> 10.000 Bit.....	34
Tabel 5.3 Hasil PSNR dengan <i>Payload</i> 50.000 Bit.....	35
Tabel 5.4 Hasil PSNR dengan <i>Payload</i> 100.000 Bit.....	36
Tabel 5.5 Jumlah Maksimal Payload.....	37
Tabel 5.6 PSNR Hasil Ekstraksi dengan <i>Payload</i> 10.000 bit.....	39
Tabel 5.7 PSNR Hasil Ekstraksi dengan <i>Payload</i> 50.000 bit.....	39
Tabel 5.8 PSNR Hasil Ekstraksi dengan <i>Payload</i> 100.000 bit...	40
Tabel 5.9 PSNR Hasil Ekstraksi dengan <i>Payload</i> 240.000.000 bit	40
Tabel 5.10 Nilai Korelasi <i>Payload</i> 10.000 bit.....	40
Tabel 5.11 Nilai Korelasi <i>Payload</i> 50.000 bit	41
Tabel 5.12 Nilai Korelasi <i>Payload</i> 100.000 bit	41
Tabel 5.13 Nilai Korelasi <i>Payload</i> 240.000.000 bit	41

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum tugas akhir.

1.1 Latar Belakang

Seiring dengan perkembangan zaman dimana perkembangan teknologi berkembang pesat, semakin besar pula kebutuhan yang diperlukan manusia untuk saling bertukar data. Dalam pertukaran data, keamanan data merupakan salah satu hal yang menjadi perhatian penting.

Steganography merupakan salah satu metode yang digunakan untuk menyembunyikan informasi. Dengan *steganography* informasi akan disembunyikan dengan cara menyisipkan informasi tersebut kedalam data lain [1]. *Steganography* dibagi menjadi dua, yaitu *reversible data hiding* dan *irreversible data hiding*. *Reversible data hiding* adalah metode dalam data hiding yang dimana hasil dari penyisipan data bisa dikembalikan atau dipulihkan sama seperti data sebelum disisipi. Sedangkan *irreversible data hiding* adalah metode dalam data hiding yang tidak dapat dipulihkan data dari hasil penyisipan kembali ke data sebelum dilakukan penyisipan.

Contoh dari *reversible data hiding* adalah *histogram shifting* dan *neighbouring similarity*. Dalam tugas akhir ini akan dilakukan penelitian untuk meningkatkan kemiripan data hasil penyisipan pada media video digital dengan modifikasi metode *histogram shifting* dan *neighbouring similarity*.

1.2 Rumusan Masalah

Tugas akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana meningkatkan kualitas penyisipan pada metode *histogram shifting* dan *neighbouring similarity*?
2. Bagaimana meningkatkan jumlah maksimal *payload* yang bisa dimasukkan dalam penyisipan pada metode *histogram shifting* dan *neighbouring similarity*?

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada tugas akhir ini memiliki batasan sebagai berikut:

1. Video yang digunakan untuk data uji berasal dari xiph.org kategori *derf collection* [2].
2. Video yang digunakan sebanyak 5 video dengan sebagian besar dipilih sama dengan video yang digunakan pada penelitian sebelumnya.
3. Data masukan untuk penyisipan didapatkan dengan cara *generate* nilai acak [3].

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah mengusulkan sebuah metode *steganography* untuk video digital yang bersifat *reversible* (bisa dikembalikan). Maksud dari *reversible* adalah video yang sudah disisipi dapat dikembalikan ke keadaan semula seperti sebelum disisipi pesan rahasia.

1.5 Manfaat

Manfaat dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Mengembangkan metode *histogram shifting* dan *neighbouring similarity* pada video sehingga hasil penyisipan semakin meningkat kemiripannya dengan video sebelum disisipi dan juga meningkatkan jumlah inputan.
2. Memberikan sumbangasih dalam bidang teknologi terutama dalam bidang steganografi

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahap awal tugas akhir ini adalah menyusun proposal tugas akhir. Pada proposal, berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Proposal juga berisi tentang garis besar tugas akhir yang akan dikerjakan sehingga memberikan gambaran untuk dapat mengerjakan tugas akhir.

1.6.2 Studi Literatur

Pada tahap ini dilakukan untuk mencari informasi dan studi literatur apa saja yang dapat dijadikan sebagai referensi untuk membantu pengerjaan tugas akhir ini. Tahap ini merupakan tahap untuk memahami semua metode yang akan dikerjakan, sehingga memberi gambaran selama pengerjaan tugas akhir. Informasi didapatkan dari buku dan literatur yang berhubungan dengan metode yang digunakan.

1.6.3 Implementasi Sistem

Implementasi merupakan tahap untuk membangun metode tersebut. Untuk mengimplementasikan metode tersebut menggunakan bahasa pemrograman *matlab*.

1.6.4 Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan bermacam citra video masukan untuk menguji aplikasi apakah telah berjalan sesuai dengan rancangan dan desain implementasi yang dibuat, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin muncul.

1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan tugas akhir.
2. Bab II. Tinjauan Pustaka
Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan tugas akhir ini.
3. Bab III. Perancangan Perangkat Lunak

Bab ini berisi pembahasan mengenai perancangan dari metode steganografi pada video digital.

4. Bab IV. Implementasi

Bab ini menjelaskan implementasi steganografi pada video digital.

5. Bab V. Pengujian dan Evaluasi

Bab ini berisikan hasil uji coba dari implementasi steganografi pada video digital yang nantinya akan dilakukan perbandingan.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan tugas akhir, dan saran untuk pengembangan solusi ke depannya.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam tugas akhir.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam tugas akhir. Teori-teori tersebut diantaranya adalah *Steganography*, *Histogram Shifting*, *Neighbouring Similarity*, dan beberapa teori lain yang mendukung pembuatan tugas akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan riset yang berkaitan.

2.1 Video Digital

Video Digital adalah representasi dari adegan video yang berbentuk digital, dimana setiap sampel diwakili oleh angka-angka yang menunjukkan nilai dari pencahayaan dan warna sampel tersebut [4]. Sebuah video terdiri dari beberapa frame yang hampir mirip antar satu frame dengan frame selanjutnya.

Dengan pergantian frame yang hampir mirip tersebut, objek dalam video tersebut terlihat bergerak. Semakin banyak frame yang digunakan dalam suatu waktu, maka akan terlihat lebih nyata pergerakan objek dalam video.

2.2 Steganography

Steganography berasal dari kata “stegos” yang berarti tertutup dan “grafia” yang berarti tulisan. *Steganography* dan *cryptography* adalah metode yang digunakan dalam data hiding [1]. *Steganography* adalah metode yang digunakan untuk menyembunyikan data kedalam data lain, sedangkan *cryptography* digunakan untuk mengubah bentuk dari sebuah data menggunakan algoritma dan kunci tertentu.

2.3 *Histogram Shifting*

Histogram Shifting adalah salah satu algoritma *reversible steganography* [5]. Metode ini bekerja dengan cara menggeser semua bit dari nilai yang memiliki frekuensi kemunculan paling besar (*peak point*) ke arah nilai yang memiliki frekuensi kemunculan paling sedikit (*zero point*) yang ada pada citra digital sebanyak satu bit. *peak point* akan digunakan untuk memasukkan pesan rahasia dengan cara menggesernya ke *zero point* jika pesan rahasianya 1 dan tidak merubahnya jika pesan rahasianya 0.

Kelemahan metode ini adalah dibutuhkannya informasi tambahan agar video hasil steganografi bisa dilakukan ekstraksi atau *recovery*. Informasi tambahan ini berupa nilai *peak point* dan *zero point*.

2.4 *Neighbouring Similarity*

Neighbouring Similarity adalah metode *reversible steganography* yang digunakan pada video [6]. Metode ini hampir sama dengan *histogram Shifting*. Perbedaannya adalah pada tahap pembangkitan *histogram*. *Histogram shifting* membangkitkan *histogram* dari nilai piksel pada frame jika diterapkan pada video, sedangkan *neighbouring similarity* membangkitkan *histogram* dari nilai selisih antara dua *frame* yang saling berdekatan.

Jumlah pesan rahasia yang bisa dimasukkan dalam metode ini lebih banyak dari *histogram shifting* karena setiap *frame* yang berdekatan memiliki nilai yang hampir sama yang menyebabkan jumlah *peak point*nya juga lebih banyak.

2.5 MATLAB

MATLAB (Matrix Laboratory) adalah sebuah bahasa pemrograman dan juga sebuah lingkungan interaktif yang dikembangkan oleh perusahaan The MathWorks, Inc [7].

2.6 Peak to Signal Noise Ratio (PSNR)

Peak to Signal Noise Ratio (PSNR) adalah tolak ukur untuk mengevaluasi metrik kualitas dari sebuah video [8]. Semakin tinggi nilai PSNR maka semakin mirip dua video. PSNR menggunakan formula Mean Square Error (MSE) untuk membandingkan nilai piksel pada cover data dengan stego data. Nilai PSNR dapat dihitung menggunakan persamaan (1.1).

$$MSE = \frac{1}{wh} \sum_{x=1}^w \sum_{y=1}^h (s_{xy} - c_{xy})^2$$

$$PSNR = 10 \log_{10} \frac{C^2max}{MSE} \quad (1.1)$$

2.7 Correlation

Correlation adalah tolak ukur untuk menganalisa tingkat kemiripan antara dua data. Nilai maksimal dari *correlation* adalah 1. Semakin dekat nilai *correlation* dengan nilai 1, semakin mirip kedua data yang dibandingkan. Nilai *correlation* dapat dihitung menggunakan persamaan (1.2).

$$Correlation = \frac{\sum_{i=1}^N (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (1.2)$$

[Halaman ini sengaja dikosongkan]

BAB III PERANCANGAN

Bab ini membahas mengenai perancangan implementasi sistem yang dibuat pada Tugas Akhir. Bagian yang akan dijelaskan pada bab ini berawal dari deskripsi umum, perancangan skenario, hingga alur dan implementasinya.

3.1 Deskripsi Umum

Pada Tugas Akhir ini penulis memodifikasi metode *histogram shifting* [5] dan *neighbour similarity* [6] untuk melakukan steganografi pada video dengan bantuan Matlab. Proses pada Metode yang diajukan secara garis besar dibagi menjadi 2 proses, proses *encoding* dan proses *decoding*. Proses *encoding* adalah proses memasukkan data rahasia kedalam video. Sedangkan proses *decoding* adalah proses mendapatkan kembali pesan rahasia dari video hasil *encoding*. Pada metode yang diajukan ini proses *encoding* dan proses *decoding* bersifat *reversible*. *Reversible* berarti video hasil proses *decoding* sama seperti video sebelum dilakukan steganografi.

Modifikasi dilakukan dengan cara mengubah tempat pengisian *payload* dan arah pergeseran bit. Pada modifikasi ini *payload* dapat dimasukkan kedalam setiap *frame* selain *frame* pertama, berbeda dengan *histogram shifting* yang dapat dimasukkan di semua *frame* dan *neighbour similarity* yang hanya bisa dimasukkan pada satu *frame* dari dua *frame*. Modifikasi juga dilakukan dengan mengubah arah pergeseran bit ke arah nilai paling kecil, nilai paling besar, atau ke arah nilai dengan frekuensi paling kecil.

3.2 Perancangan Data

Pada subbab ini akan dijelaskan masukan untuk perangkat lunak dan keluaran yang dihasilkan perangkat lunak

3.2.1 Data Masukkan

Pada proses penyisipan, data yang digunakan adalah berkas video dan berkas *payload*. Sedangkan pada proses ekstraksi dibutuhkan berkas *stego video* dan *location map*.

Berkas video yang dipakai sebagai *cover video* adalah berkas video yang memiliki ekstensi .avi. Pada tugas akhir ini, akan digunakan 5 berkas video [2] dengan durasi 10 detik, jumlah frame sebanyak 300, ukuran *frame* 176x144, dan *frame rate* 29. Detail berkas – berkas video tersebut, ditunjukkan pada Tabel 3.1.

Tabel 3.1 Cover Video

No	Nama Berkas	Ukuran Video (KB)
1	akiyo_qcif	235
2	coastguard_qcif_mono	772
3	foreman_qcif_mono	598
4	hall_monitor_qcif	421
5	mother_daughter_qcif	307

Berkas yang dipakai sebagai *payload* adalah berkas teks yang memiliki ekstensi .txt. Isi dari *payload* adalah bilangan biner [3]. Pada tugas akhir ini digunakan 3 berkas teks dengan jumlah isi yang berbeda-beda. Detail *payload* ditunjukkan pada Tabel 3.2.

Tabel 3.2 Payload

No.	Nama Berkas	Jumlah Data(bit)
1	10kb_new.txt	10000
2	50kb_new.txt	50000
3	100kb_new.txt	100000
4	big_new.txt	24000000

Pada proses ekstraksi, data yang dibutuhkan adalah berkas *stego video* dan *location map*. *Stego video* adalah video yang telah diberi data dari *payload*. *Location map* berisi data tempat pergeseran bit, arah pergeseran bit, dan data untuk mencegah *overflow* dan *underflow*.

3.2.2 Data Keluaran

Data keluaran dari proses penyisipan adalah *stego video* dan *location map* yang akan digunakan untuk proses ekstraksi. Data keluaran dari proses ekstraksi adalah video dan *payload* yang memiliki data yang sama dengan video *cover* dan *payload* yang digunakan saat proses penyisipan.

3.3 Perancangan Modifikasi Metode

Pada subbab ini membahas perancangan sebuah proses pada sistem untuk menggambarkan proses lebih detil terhadap alur implementasi metode pada steganografi ini.

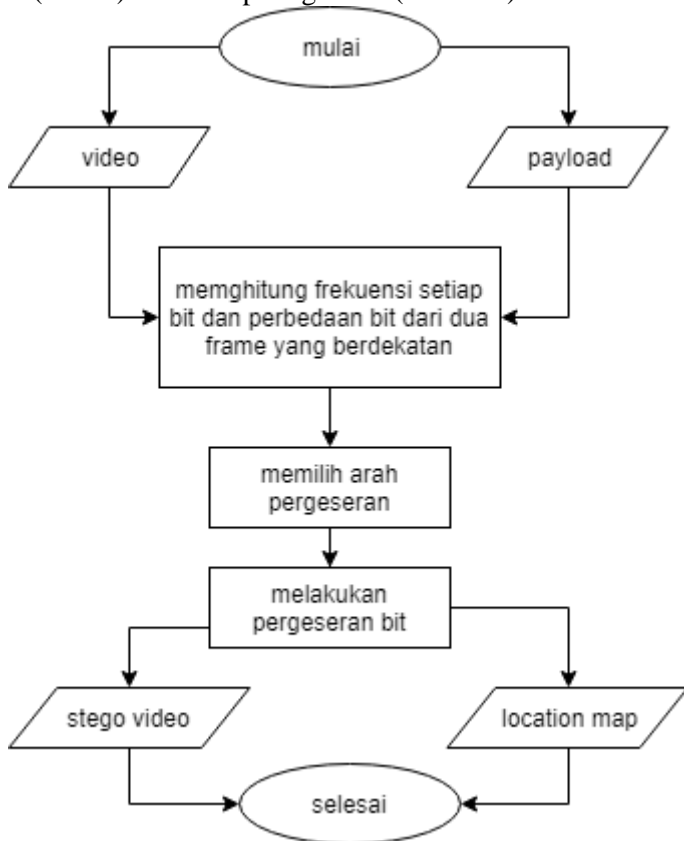
3.3.1 Proses Penyisipan

Pada proses penyisipan data, digunakan dua berkas yaitu berkas video dan berkas *payload*. Data dari berkas *payload* akan dimasukkan ke dalam berkas video berdasarkan diagram alir pada Gambar 3.1.

3.3.1.1 Penghitungan Frekuensi

Pada tahapan ini video akan dipindai secara vertikal dan horizontal terlebih dahulu. Setelah itu pilih *zero point*(nilai dengan frekuensi terendah) dan *peak point*(nilai dengan frekuensi tertinggi) di setiap *frame* selain *frame* pertama. Yang membedakan dengan metode *histogram shifting*, *zero point* tidak hanya nilai

dengan frekuensi terendah tetapi juga ditambahkan dengan nilai paling kiri(nilai 0) dan nilai paling kanan(nilai 255).



Gambar 3.1 Diagram Alir Penyisipan Data

3.3.1.2 Pembangkitan *Prediction Error*

Prediction error berisi nilai dari sebuah *frame* dikurangi dengan *frame* yang dibelakangnya (*frame* 1 dengan *frame* 2, *frame* 2 dengan *frame* 3 dan seterusnya), berbeda dengan *neighbouring similarity* yang hanya ada satu *prediction error* untuk dua

frame. Setelah itu pilih *zero point* dan *peak point* di setiap *prediction error*. Pilih juga nilai paling kiri (nilai -255) dan nilai paling kanan (nilai 255) sebagai *zero point*.

3.3.1.3 Pemilihan Arah Pergeseran

Pada tahapan ini, pilih *zero point* yang paling dekat dengan *peak point* di setiap *frame* dan *prediction error*. Maksud dari *zero point* paling dekat disini adalah *zero point* dimana frekuensi diantara *zero point* dan *peak point* paling sedikit. Setelah itu hitung nilai perbandingan antara jumlah frekuensi *peak point* dan jumlah frekuensi yang berada diantara *peak point* dan *zero point* paling dekat untuk setiap *frame* dan *prediction error*. Pemilihan arah pergeseran dilakukan sesuai persamaan (3.1).

Jumlah *peak point* ada 2, *peak point* pertama berada pada *frame* dan *peak point* kedua berada pada *prediction error*. Jumlah *zero point* ada 6. *Zero point 1* adalah *zero point* pada *frame*, *zero point 2* nilai tertinggi pada *frame* (255), *zero point 3* adalah nilai terendah pada *frame* (0), *zero point 4* adalah *zero point* pada *prediction error*, *zero point 5* adalah nilai tertinggi pada *prediction error* (255), dan *zero point 6* adalah nilai terendah pada *prediction error* (-255).

μ_i adalah jumlah frekuensi pada *peak point* ke- i . Ω_{ij} adalah jumlah frekuensi yang memiliki nilai diantara *peak point* ke- i dan *zero point* ke- j .

$$S1 = \frac{\mu_1}{\Omega_{11}}$$

$$S2 = \frac{\mu_1}{\Omega_{12}}$$

$$S3 = \frac{\mu_1}{\Omega_{13}}$$

$$S4 = \frac{\mu_2}{\Omega_{24}}$$

$$S5 = \frac{\mu_2}{\Omega_{25}}$$

$$S6 = \frac{\mu_2}{\Omega_{26}}$$

$$S = \text{MAX}(S1, S2, S3, S4, S5, S6) \quad (3.1)$$

3.3.1.4 Shifting(Pergeseran)

Jika nilai terbesar perbandingan pada tahap sebelumnya adalah nilai perbandingan *frame*, pergeseran akan dilakukan pada *frame*. Jika nilai terbesar perbandingan *prediction error* terbesar, maka pergeseran akan dilakukan pada *prediction error*. Pada tahap ini nilai yang berada diantara *zero point* dan *peak point* akan digeser sebanyak 1 bit dari arah *peak point* ke *zero point*. Pergeseran dilakukan sebagai berikut:

- i. Jika S sama dengan S1, lakukan pergeseran sesuai dengan persamaan (3.2). $\alpha_{x,y,z}$ adalah nilai *frame* pada kolom z, baris y, dan *frame* ke-z, p adalah *peak point* yang terpilih, dan z adalah *zero point* yang terpilih.

$$\alpha_{x,y,z} = \alpha_{x,y} + 1, \text{ jika } p < z, \alpha_{x,y} > p, \alpha_{x,y} < z$$

$$\alpha_{x,y,z} = \alpha_{x,y} - 1, \text{ jika } p > z, \alpha_{x,y} < p, \alpha_{x,y} > z \quad (3.2)$$

- ii. Jika S sama dengan S2, lakukan pergeseran sesuai persamaan (3.2).
- iii. Jika S sama dengan S3, lakukan pergeseran sesuai persamaan (3.2).
- iv. Jika S sama dengan S4, lakukan pergeseran sesuai persamaan (3.3). $PE_{x,y,z}$ nilai *prediction error* pada kolom z, baris y, dan *prediction error* ke-z

$$\begin{aligned}
 PE_{x,y,z} &= PE_{x,y} + 1, \text{ jika } p < z, PE_{x,y} > p, \alpha_{x,y} < z \\
 PE_{x,y,z} &= PE_{x,y} - 1, \text{ jika } p > z, PE_{x,y} < p, \alpha_{x,y} > z \quad (3.3)
 \end{aligned}$$

- v. Jika S sama dengan S5, buat prediction error lalu lakukan pergeseran pada prediction error menggunakan persamaan (3.3).
- vi. Jika S sama dengan S6, buat prediction error lalu lakukan pergeseran pada prediction error menggunakan persamaan (3.3).

3.3.1.5 Penyisipan Payload

Pada tahap ini nilai dari *peak point* akan digeser sebanyak nilai *payload* ke arah *zero point*. Proses penyisipan menggunakan persamaan (3.4) jika nilai S terbesar adalah S1, S2, atau S3, dan memakai persamaan (3.5) jika nilai S4, S5, atau S6 adalah nilai S yang terbesar. β_n adalah nilai *payload* ke-n.

$$\begin{aligned}
 \alpha_{x,y,z} &= \alpha_{x,y,z} + \beta_n, \text{ jika } p < z, \alpha_{x,y,z} = p \\
 \alpha_{x,y,z} &= \alpha_{x,y,z} - \beta_n, \text{ jika } p > z, \alpha_{x,y,z} = p \quad (3.4)
 \end{aligned}$$

$$\begin{aligned}
 PE_{x,y,z} &= PE_{x,y,z} + \beta_n, \text{ jika } p < z, PE_{x,y,z} = p \\
 PE_{x,y,z} &= PE_{x,y,z} - \beta_n, \text{ jika } p > z, PE_{x,y,z} = p \quad (3.5)
 \end{aligned}$$

3.3.1.6 Pencegahan *Overflow* dan *Underflow*

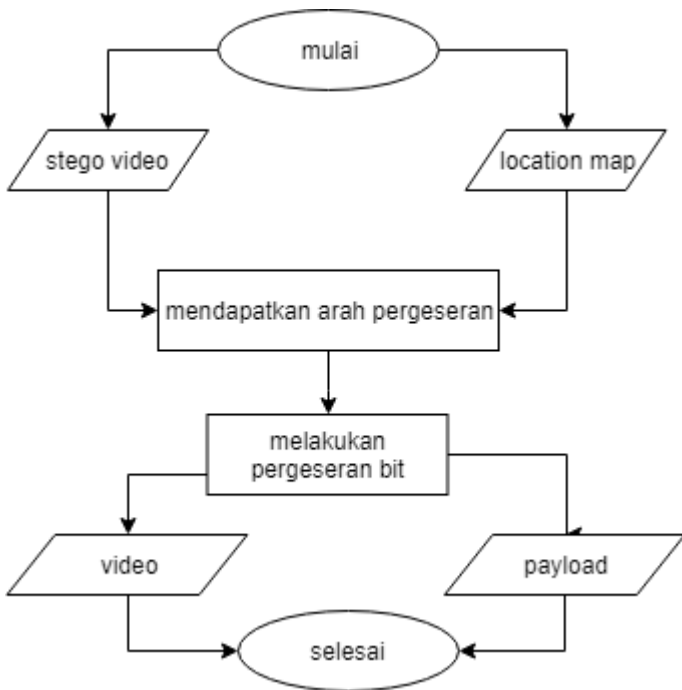
Untuk mencegah *overflow* dan *underflow*, maka diperlukan data tambahan untuk ditambahkan ke dalam *location map*. Penambahan data pada *location map* dapat dilihat pada tabel Tabel 3.3. Pada Tabel 3.3 terdapat bagian yang terblok warna hitam. Hal ini menyatakan bahwa nilai yang ada pada bagian tersebut diabaikan.

Tabel 3.3 Pencegahan *Overflow* dan *Underflow*

s	z	frame	PE	payload	lm	frame hasil	
s1,s2,s3	>p	z			0	z	
		z+1			1	z	
	<p	z			0	z	
		z-1			1	z	
s4,s5,s6	>p	255	p	0	0	255	
		255	p	1	2	255	
		254	p	1	1	255	
		254	<p, >z		1	255	
		255	<p, >z		0	255	
			z		0	z	
	<p		z+1			1	z
		0	p	0	0	0	
		0	p	1	2	0	
		1	p	1	1	0	
		1	>p, <z		1	0	
		0	>p, <z		0	0	
			z		0	z	
			z+1		1	z	

3.3.2 Proses Ekstraksi

Pada proses ekstraksi data, data *stego video* dan data *location map* diperlukan untuk mendapatkan berkas video asli. Diagram alir dari proses ekstraksi dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir Ekstraksi Data

3.3.2.1 Pembacaan *Location Map*

Tahap pertama proses *decoding* dilakukan dengan cara mendapatkan data yang berisi nilai *S*, *zero point* dan *peak point* dari *location map*.

3.3.2.2 Shifting(Pergeseran)

Pada tahap ini, semua nilai yang berada diantara *peak point* dan *zero point* akan digeser sebanyak 1 bit dari arah *peak point* ke *zero point*. Tahap ini dilakukan pada *frame* apabila nilai *S* sama dengan 0 dan dilakukan pada *prediction error* apabila nilai *S* sama

1. Jika nilai S adalah 0, maka pergeseran menggunakan persamaan (3.7). Jika nilai S adalah 1 maka menggunakan persamaan (3.8)

$$\begin{aligned}\alpha_{x,y,z} &= \alpha_{x,y} - 1, \text{ jika } p < z, \alpha_{x,y} > p, \alpha_{x,y} < z \\ \alpha_{x,y,z} &= \alpha_{x,y} + 1, \text{ jika } p > z, \alpha_{x,y} < p, \alpha_{x,y} > z\end{aligned}\quad (3.7)$$

$$\begin{aligned}PE_{x,y,z} &= PE_{x,y} - 1, \text{ jika } p < z, PE_{x,y} > p, \alpha_{x,y} < z \\ PE_{x,y,z} &= PE_{x,y} + 1, \text{ jika } p > z, PE_{x,y} < p, \alpha_{x,y} > z\end{aligned}\quad (3.8)$$

3.3.2.3 Pengembalian *Payload*

Nilai *payload* bisa didapatkan dengan mencari nilai *peak point* dan nilai selisih satu dari *peak point* yang mengarah ke *zero point*. Pengembalian *payload* dilakukan menggunakan persamaan (3.9) jika nilai s adalah 0 dan menggunakan persamaan (3.10) jika nilai s adalah 1.

$$\begin{aligned}\beta &= 0, \text{ jika } \alpha_{x,y,z} = p \\ \beta &= 1, \text{ jika } \alpha_{x,y,z} = p - 1, p > z \\ \beta &= 1, \text{ jika } \alpha_{x,y,z} = p + 1, p < z\end{aligned}\quad (3.9)$$

$$\begin{aligned}\beta &= 0, \text{ jika } PE_{x,y,z} = p \\ \beta &= 1, \text{ jika } PE_{x,y,z} = p - 1, p > z \\ \beta &= 1, \text{ jika } PE_{x,y,z} = p + 1, p < z\end{aligned}\quad (3.10)$$

BAB IV IMPLEMENTASI

Bab ini membahas mengenai implementasi dari perancangan sistem yang telah dijabarkan pada bab perancangan. Implementasi berupa kode sumber untuk membangun program.

4.1 Modul Penyisipan

Modul penyisipan berisi implementasi dari tahapan-tahapan pada proses penyisipan.

4.1.1 Fungsi Pemindaian

Fungsi Pemindaian berguna untuk memindai video *cover* dan *payload*. Fungsi pemindaian dapat dilihat pada Gambar 4.1.

```
1 video = VideoReader('nama_file.avi');
2 payload = fopen('nama_payload.txt');
```

Gambar 4.1 Potongan Program Pemindaian Penyisipan

4.1.2 Fungsi Perhitungan Frekuensi

Fungsi perhitungan frekuensi berguna untuk mencari *peak point* dan *zero point*. Fungsi penyisipan dapat dilihat pada Gambar 4.2.

```
1 fr = 0
2 while hasFrame(video)
3     frek2 = zeros(511,1);
4     if fr == 1 || masukpl == 0
5         a = readFrame(obj);
6         c = double(a);
7         writeVideo(newobj, a);
8     elseif masukpl == 1
9         a = c;
10        b = readFrame(obj);
11        b = double(b);
12        c = b;
```

```

13     for i = 1:1:size(frame2,1)
14         frek1(frame2(i)+1) = frek1(frame2(i)+1)+1;
15     end
16     [maxim1,peakpoint1]=max(frek1);
17     minim1=min(frek1);
18     locmin1bawah=1;
19     locmin1atas=256;
20     valuebawah=0;
21     valueatas=0;
22     if peakpoint1>0
23         for i = peakpoint1:-1:1
24             valuebawah = valuebawah+frek1(i);
25             if frek1(i)==minim1
26                 if i==peakpoint1-1
27                     if frek1(i)==0
28                         locmin1bawah=i;
29                         break;
30                     end
31                 else
32                     locmin1bawah=i;
33                     break;
34                 end
35             end
36         end
37     else
38         valuebawah = inf;
39     end
40     if peakpoint1<256
41         for i = peakpoint1:1:256
42             valueatas = valueatas+frek1(i);
43             if frek1(i)==minim1
44                 if i==peakpoint1+1
45                     if frek1(i)==0
46                         locmin1atas=i;
47                         break;
48                     end
49                 else
50                     locmin1atas=i;
51                     break;
52                 end
53             end
54         end
55     else
56         valueatas = inf;
57     end
58     if valuebawah < valueatas
59         zeropoint1 = locmin1bawah-1;
60         value1 = valuebawah;

```

```

61         else
62             zeropoint1 = locmin1atas-1;
63             value1 = valueatas;
64         end
65         peakpoint1 = peakpoint1-1;
66     end
67 end

```

Gambar 4.2 Potongan Program Perhitungan Frekuensi

4.1.3 Fungsi Pembangkitan *Prediction Error*

Fungsi pembangkitan *prediction error* berguna untuk mendapatkan nilai pengurangan dari dua *frame* serta mencari *peak point* dan *zero point* pada *prediction error* tersebut. Fungsi pembangkitan *prediction error* dapat dilihat pada Gambar 4.3.

```

1   frek2 = zeros(511,1);
2   framedif = reshape(diff,[],1);
3   for i = 1:1:size(framedif,1)
4       frek2(framedif(i)+256) = frek2(framedif(i)+256)+1;
5   end
6   [maxim2,peakpoint2]=max(frek2);
7   valuebawah=0;
8   valueatas=0;
9   for i = peakpoint2:-1:1
10      valuebawah = valuebawah+frek2(i);
11      if frek2(i)==minim2
12          if i==peakpoint2-1
13              if frek2(i)==0
14                  locmin2bawah=i;
15                  break;
16              end
17          else
18              locmin2bawah=i;
19              break;
20          end
21      end
22  end
23  for i = peakpoint2:1:511
24      valueatas = valueatas+frek2(i);
25      if frek2(i)==minim2
26          if i==peakpoint2+1
27              if frek2(i)==0
28                  locmin2atas=i;
29                  break;

```

```

30         end
31     else
32         locmin2atas=i;
33         break;
34     end
35 end
36 end
37 if valuebawah < valueatas
38     zeropoint2 = locmin2bawah-256;
39     value2 = valuebawah;
40 else
41     zeropoint2 = locmin2atas-256;
42     value2 = valueatas;
43 end
44 peakpoint2 = peakpoint2-256;

```

Gambar 4.3 Potongan Program Pembangkitan *Prediction Error*

4.1.4 Fungsi Pemilihan Arah Pergeseran

Fungsi pemilihan arah pergeseran dilakukan dengan cara mencari arah pergeseran yang memiliki *peak point* berfrekuensi banyak dan memiliki nilai yang berfrekuensi antara *peak point* dan *zero point* yang sedikit. Fungsi pembangkitan *prediction error* dapat dilihat pada Gambar 4.4.

```

1  if (maxim1/value1)>maxim2/value2
2      s=0;
3      lm='0';
4      lm=strcat(lm,',');
5      lm=strcat(lm,num2str(peakpoint1));
6      lm=strcat(lm,',');
7      lm=strcat(lm,num2str(zeropoint1));
8      lm=strcat(lm,',');
9      if frek1(zeropoint1+1)==0
10         lm=strcat(lm,'2');
11         fz1=1;
12     else
13         fz1=0;
14     end
15 else
16     s=1;
17     lm='1';
18     lm=strcat(lm,',');

```



```

19     lm=strcat(lm,num2str(peakpoint2));
20     lm=strcat(lm,',');
21     lm=strcat(lm,num2str(zeropoint2));
22     lm=strcat(lm,',');
23 end

```

Gambar 4.4 Potongan Program Pemilihan Arah Pergeseran

4.1.5 Fungsi Pergeseran

Fungsi pergeseran dilakukan dengan cara mencari arah pergeseran yang memiliki *peak point* berfrekuensi banyak dan nilai yang . Fungsi pergeseran dapat dilihat pada Gambar 4.5.

```

1  if s==0
2      if peakpoint1<zeropoint1
3          b(i,j,k)=b(i,j,k)+1;
4      else
5          b(i,j,k)=b(i,j,k)-1;
6      end
7  else
8      if peakpoint2<zeropoint2
9          b(i,j,k)=b(i,j,k)-1;
10     else
11         b(i,j,k)=b(i,j,k)+1;
12     end
13 end

```

Gambar 4.5 Potongan Program Pergeseran

4.1.6 Fungsi Penyisipan *Payload*

Fungsi penyisipan *payload* dilakukan dengan cara menggeser nilai dari *peak point* sebanyak nilai *payload* ke arah *zero point*. Fungsi pembangkitan *prediction error* dapat dilihat pada Gambar 4.6.

```

1  if s==0
2      if peakpoint1<zeropoint1
3          if b(i,j,k)==peakpoint1
4              b(i,j,k)=b(i,j,k)+payloadbin(payloadke,1);
5              payloadke=payloadke+1;
6          end
7      end

```

```

8      else
9          if b(i,j,k)==peakpoint1
10             b(i,j,k)=b(i,j,k)-payloadbin(payloadke,1);
11             payloadke=payloadke+1;
12         end
13     end
14 else
15     if peakpoint2<zeropoint2
16         if (a(i,j,k)-b(i,j,k))==peakpoint2
17             b(i,j,k)=b(i,j,k)-payloadbin(payloadke,1);
18         end
19     else
20         if (a(i,j,k)-b(i,j,k))==peakpoint2
21             b(i,j,k)=b(i,j,k)+payloadbin(payloadke,1);
22         end
23     end
24 end

```

Gambar 4.6 Potongan Program Penyisipan *Payload*

4.1.7 Fungsi Pencegahan *Overflow* dan *Underflow*

Pencegahan *overflow* dan *underflow* dilakukan dengan cara menyimpan informasi pencegahan *overflow* dan *underflow* ke dalam *location map*. Fungsi pembangkitan *prediction error* dapat dilihat pada Gambar 4.7.

```

1  if s==0
2      if peakpoint1<zeropoint1
3          if b(i,j,k)>peakpoint1 && b(i,j,k)<zeropoint1
4              if b(i,j,k)+1==zeropoint1
5                  b(i,j,k)=b(i,j,k)+1;
6                  if fz1==0
7                      lm=strcat(lm,'1');
8                  end
9              end
10             elseif b(i,j,k)==zeropoint1
11                 lm=strcat(lm,'0');
12             end
13         else
14             if b(i,j,k)<peakpoint1 && b(i,j,k)>zeropoint1
15                 if b(i,j,k)-1==zeropoint1
16                     b(i,j,k)=b(i,j,k)-1;
17                     if fz1==0
18                         lm=strcat(lm,'1');
19                 end

```

```

20         end
21     elseif b(i,j,k)==zeropoint1
22         lm=strcat(lm, '0');
23     end
24     end
25 else
26     if peakpoint2<zeropoint2
27         if (a(i,j,k)-b(i,j,k))>peakpoint2 && (a(i,j,k)-
b(i,j,k))<zeropoint2
28             if b(i,j,k)-1==0
29                 b(i,j,k)=b(i,j,k)-1;
30                 lm=strcat(lm, '1');
31             elseif b(i,j,k)==0
32                 lm=strcat(lm, '0');
33             elseif (a(i,j,k)-b(i,j,k))==zeropoint2-1
34                 b(i,j,k)=b(i,j,k)-1;
35                 lm=strcat(lm, '1');
36             end
37             elseif (a(i,j,k)-b(i,j,k))==zeropoint2
38                 lm=strcat(lm, '0');
39             elseif (a(i,j,k)-b(i,j,k))==peakpoint2
40                 if b(i,j,k)==0
41                     if payloadbin(nomor,1)==0
42                         lm=strcat(lm, '0');
43                     else
44                         lm=strcat(lm, '2');
45                     end
46                 end
47             end
48         else
49             if (a(i,j,k)-b(i,j,k))<peakpoint2 && (a(i,j,k)-
b(i,j,k))>zeropoint2
50                 if b(i,j,k)+1==255
51                     b(i,j,k)=b(i,j,k)+1;
52                     lm=strcat(lm, '1');
53                 elseif b(i,j,k)==255
54                     lm=strcat(lm, '0');
55                 elseif (a(i,j,k)-b(i,j,k))==zeropoint2+1
56                     b(i,j,k)=b(i,j,k)+1;
57                     lm=strcat(lm, '1');
58                 end
59                 elseif (a(i,j,k)-b(i,j,k))==zeropoint2
60                     lm=strcat(lm, '0');
61                 elseif (a(i,j,k)-b(i,j,k))==peakpoint2
62                     if b(i,j,k)==255
63                         if payloadbin(nomor,1)==0
64                             lm=strcat(lm, '0');
65                     else

```

```

66         lm=strcat(lm,'2');
67         end
68         elseif b(i,j,k)+1==255
69             if payloadbin(nomor,1)==1
70                 lm=strcat(lm,'1');
71                 b(i,j,k)=b(i,j,k)+1;
72             end
73         end
74     end
75 end
76 end

```

Gambar 4.7 Potongan Program Pencegahan *Overflow* dan *Underflow*

4.1.8 Fungsi Penyimpanan *Location Map*

Fungsi penyimpanan *location map* dilakukan untuk menyimpan informasi-informasi yang diperlukan untuk proses ekstraksi. Fungsi penyimpanan *location map* dapat dilihat pada Gambar 4.8.

```

1  Location_map = fopen('LM.txt','w');
2  fprintf(Location_map,'%s\n',lm);

```

Gambar 4.8 Potongan Program Penyimpanan *Location Map*

4.2 Modul Ekstraksi

Modul ekstraksi berisi *implementasi* dari tahapan-tahapan pada proses ekstraksi. Ekstraksi dilakukan untuk mendapatkan data video dan *payload* semula.

4.2.1 Fungsi Pemindaian

Fungsi Pemindaian dilakukan untuk memindai *stego* video dan *location map*. Fungsi pemindaian dapat dilihat pada Gambar 4.9.

```

1  video = VideoReader('stego_video.avi');
2

```

```

fid1 = fopen('location_map.txt');
while hasFrame(video)
    lm1 = fscanf(fid1, '%d,%d,%d');
    lm2 = fscanf(fid1, ',%s');
    s = lm1(1);
    peak = lm1(2);
    zero = lm1(3);
end

```

Gambar 4.9 Potongan Program Pemindaian Ekstraksi

4.2.2 Fungsi Pergeseran

Fungsi ekstraksi dilakukan dengan tujuan untuk mendapatkan kembali *payload* dan *cover* video dari *location map* dan *stego* video. Fungsi ekstraksi dapat dilihat pada Gambar 4.10.

```

1  if s==0
2      if peak<zero
3          if b(i,j,k)>peak && b(i,j,k)<zero
4              b(i,j,k)=b(i,j,k)-1;
5          elseif b(i,j,k)==zero
6              if lm2(lmke)=='2'
7                  b(i,j,k)=b(i,j,k)-1;
8              else
9                  b(i,j,k)=b(i,j,k)-str2num(lm2(lmke));
10                 lmke=lmke+1;
11             end
12         end
13     else
14         if b(i,j,k)<peak && b(i,j,k)>zero
15             b(i,j,k)=b(i,j,k)+1;
16         elseif b(i,j,k)==zero
17             if lm2(lmke)=='2'
18                 b(i,j,k)=b(i,j,k)+1;
19             else
20                 b(i,j,k)=b(i,j,k)+str2num(lm2(lmke));
21                 lmke=lmke+1;
22             end
23         end
24     end
25 elseif s==1
26     if peak<zero
27         if (a(i,j,k)-b(i,j,k))>peak && (a(i,j,k)-
b(i,j,k))<zero
28             if b(i,j,k)==0

```

```

29         if lm2(lmke)=='0'
30             lmke=lmke+1;
31         elseif lm2(lmke)=='1'
32             b(i,j,k)=b(i,j,k)+1;
33             lmke=lmke+1;
34         end
35     else
36         b(i,j,k)=b(i,j,k)+1;
37     end
38     elseif (a(i,j,k)-b(i,j,k))==zero
39         if lm2(lmke)=='0'
40             lmke=lmke+1;
41         elseif lm2(lmke)=='1'
42             b(i,j,k)=b(i,j,k)+1;
43             lmke=lmke+1;
44         end
45     end
46 else
47     if (a(i,j,k)-b(i,j,k))<peak && (a(i,j,k)-
b(i,j,k))>zero
48         if b(i,j,k)==255
49             if lm2(lmke)=='0'
50                 lmke=lmke+1;
51             elseif lm2(lmke)=='1'
52                 b(i,j,k)=b(i,j,k)-1;
53                 lmke=lmke+1;
54             end
55         else
56             b(i,j,k)=b(i,j,k)-1;
57         end
58     elseif (a(i,j,k)-b(i,j,k))==zero
59         if lm2(lmke)=='0'
60             lmke=lmke+1;
61         elseif lm2(lmke)=='1'
62             b(i,j,k)=b(i,j,k)-1;
63             lmke=lmke+1;
64         end
65     end
66 end
67 end

```

Gambar 4.10 Potongan Program Ekstraksi

4.2.3 Fungsi Pengembalian *Payload*

Fungsi pengembalian payload dilakukan untuk bertujuan untuk mendapatkan kembali data *payload* dari *stego* video. Fungsi pengembalian payload dapat dilihat pada Gambar 4.11.

```

1  if s==0
2      if peak<zero
3          if b(i,j,k)==peak
4              binary(textke)=0;
5              textke=textke+1;
6          elseif b(i,j,k)==peak+1
7              b(i,j,k)=b(i,j,k)-1;
8              binary(textke)=1;
9              textke=textke+1;
10         end
11     else
12         if b(i,j,k)==peak
13             binary(textke)=0;
14             textke=textke+1;
15         elseif b(i,j,k)==peak-1
16             b(i,j,k)=b(i,j,k)+1;
17             binary(textke)=1;
18             textke=textke+1;
19         end
20     end
21 elseif s==1
22     if peak<zero
23         if (a(i,j,k)-b(i,j,k))==peak
24             binary(textke)=0;
25             textke=textke+1;
26         elseif (a(i,j,k)-b(i,j,k))==peak+1
27             b(i,j,k)=b(i,j,k)+1;
28             binary(textke)=1;
29             textke=textke+1;
30         end
31     else
32         if (a(i,j,k)-b(i,j,k))==peak
33             binary(textke)=0;
34             textke=textke+1;
35         elseif (a(i,j,k)-b(i,j,k))==peak-1
36             b(i,j,k)=b(i,j,k)-1;
37             binary(textke)=1;
38             textke=textke+1;
39         end
40     end

```

```

41 end
42 payload = fopen('hasil_payload.txt','w');
43 fprintf(payload, '%s\n', binary);

```

Gambar 4.11 Potongan Program Pengembalian Payload

4.3 Modul Analisis

Modul analisis berisi *implementasi* dari tahapan-tahapan untuk menganalisa hasil uji coba. Fungsi yang digunakan adalah fungsi PSNR dan fungsi korelasi.

4.3.1 Fungsi PSNR

Fungsi PSNR digunakan untuk menghitung tingkat kemiripan dari dua video. Fungsi PSNR dapat dilihat pada Gambar 4.12.

```

1 tot=0;
2 for i=1:1:size(video1,1)
3     for j=1:1:size(video1,2)
4         for k=1:1:size(video1,3)
5             tot=tot+(power(video1(i,j,k)-video2(i,j,k),2));
6         end
7     end
8 end
9 mse=tot/(size(video1,1)*size(video1,2)*size(video1,3));
10 psnr=10*log10(power(255,2)/mse);

```

Gambar 4.12 Potongan Program Fungsi PSNR

4.3.2 Fungsi Korelasi

Fungsi korelasi berguna untuk menghitung nilai korelasi antara *payload* asli dan *payload* hasil ekstraksi. Fungsi korelasi dapat dilihat pada Gambar 4.13.

```

1 Payload1 = fopen('payload.txt');
2 Payload2 = fopen('hasil_payload.txt');
3 Korelasi = corrcoef(Payload1, Payload2);

```

Gambar 4.13 Potongan Program Fungsi Korelasi

BAB V

HASIL UJI COBA DAN EVALUASI

Bab ini membahas mengenai uji coba dan evaluasi dari skenario yang telah dilakukan.

5.1 Lingkungan Uji Coba

Uji coba dilakukan pada perangkat dengan spesifikasi seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat yang Digunakan

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel Core i7-2630QM CPU @ 2.00 GHz
	Memori	8 Gb DDR3 665.1 MHz
Perangkat Lunak	Sistem Operasi	Windows 8.1 64-bit
	Perangkat Pengembang	Matlab R2016a
	Bahasa Pemrograman	Matlab

5.2 Hasil Uji Coba

Hasil uji coba terdiri dari hasil uji coba skenario jumlah *payload* yang bisa dimasukkan, hasil uji coba kemiripan video hasil penyisipan dengan video asli, dan hasil uji coba kemiripan video dan *payload* hasil ekstraksi dengan video dan *payload* sebelum dilakukan penyisipan.

5.2.1 Hasil Uji Coba 1

Uji coba 1 dilakukan dengan cara menyisipkan *payload* sebesar 10.000 bit, 50.000 bit, dan 100.000 bit ke dalam 5 video. Uji coba ini dimaksudkan untuk mengetahui kualitas dari hasil penyisipan. Hasil uji coba ditunjukkan oleh Tabel 5.2, Tabel 5.3, dan Tabel 5.4.

Tabel 5.2 Hasil PSNR dengan *Payload* 10.000 Bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	83,3929	66,1002	82,2759
coastguard	76,3721	68,1189	74,5602
foreman	77,3344	79,9066	74,6645
hall monitor	77,6305	65,1902	74,6543
mother daughter	79,9674	68,2671	77,7032

Dari data yang ditunjukkan oleh Tabel 5.2, dapat dilihat bahwa nilai PSNR untuk *payload* 10.000 bit meningkat. Berikut perbandingan peningkatan PSNR:

- Nilai PSNR video akiyo meningkat sebanyak 17,2926 dB dari HS dan meningkat sebanyak 1,117 dB dari NS.
- Nilai PSNR video coastguard meningkat sebanyak 8,2532 dB dari HS dan meningkat sebanyak 1,8119 dB dari NS.
- Nilai PSNR video foreman menurun sebanyak 2,5721 dB dari HS dan meningkat sebanyak 2,6699 dB dari NS.
- Nilai PSNR video hall monitor meningkat sebanyak 12,4403 dB dari HS dan meningkat sebanyak 2,9762 dB dari NS.

- Nilai PSNR video mother daughter meningkat sebanyak 11,7003 dB dari HS dan meningkat sebanyak 2,2642 dB dari NS.

Dapat dilihat bahwa rata-rata nilai PSNR meningkat sebanyak 13,55% dari HS dan meningkat sebanyak 2,82% dari NS. Dapat dilihat juga bahwa hanya PSNR video foreman saja yang mengalami penurunan. Hal ini dikarenakan nilai *peak point* dan *zero point* pada beberapa *frame* awal video foreman cukup dekat, sehingga bit yang digeser saat proses penyisipan sangat sedikit.

Tabel 5.3 Hasil PSNR dengan Payload 50.000 Bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	77,4757	59,2389	77,1612
coastguard	69,1969	60,4401	68,2847
foreman	73,3260	63,0052	70,7876
hall monitor	73,3728	58,3434	71,1010
mother daughter	76,3327	64,7684	75,1532

Dari data yang ditunjukkan oleh Tabel 5.3, dapat dilihat bahwa nilai PSNR untuk *payload* 50.000 bit meningkat. Berikut perbandingan peningkatan PSNR:

- Nilai PSNR video akiyo meningkat sebanyak 18,2368 dB dari HS dan meningkat sebanyak 0,3144 dB dari NS.
- Nilai PSNR video coastguard meningkat sebanyak 8,7568 dB dari HS dan meningkat sebanyak 0,9122 dB dari NS.
- Nilai PSNR video foreman meningkat sebanyak 10,3208 dB dari HS dan meningkat sebanyak 2,5384 dB dari NS.

- Nilai PSNR video hall monitor meningkat sebanyak 15,0294 dB dari HS dan meningkat sebanyak 2,2718 dB dari NS.
- Nilai PSNR video mother daughter meningkat sebanyak 11,5642 dB dari HS dan meningkat sebanyak 1,1794 dB dari NS.

Dapat dilihat bahwa rata-rata nilai PSNR meningkat sebanyak 29,07% dari HS dan meningkat sebanyak 8,89% dari NS.

Tabel 5.4 Hasil PSNR dengan *Payload* 100.000 Bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	74,5205	56,3981	74,2443
coastguard	66,5867	57,3269	65,7776
foreman	70,2034	59,8116	67,7878
hall monitor	70,4598	55,4886	68,2671
mother daughter	73,5728	61,7608	72,2969

Dari data yang ditunjukkan oleh Tabel 5.4, dapat dilihat bahwa nilai PSNR untuk *payload* 100.000 bit meningkat. Berikut perbandingan peningkatan PSNR:

- Nilai PSNR video akiyo meningkat sebanyak 18,1225 dB dari HS dan meningkat sebanyak 0,2762 dB dari NS.
- Nilai PSNR video coastguard meningkat sebanyak 9,2598 dB dari HS dan meningkat sebanyak 0,809 dB dari NS.
- Nilai PSNR video foreman meningkat sebanyak 10,3918 dB dari HS dan meningkat sebanyak 2,4156 dB dari NS.

- Nilai PSNR video hall monitor meningkat sebanyak 14,9712 dB dari HS dan meningkat sebanyak 2,1927 dB dari NS.
- Nilai PSNR video mother daughter meningkat sebanyak 11,8121 dB dari HS dan meningkat sebanyak 1,2759 dB dari NS.

Dapat dilihat bahwa rata-rata nilai PSNR meningkat sebanyak 35,73% dari HS dan meningkat sebanyak 13,3% dari NS.

Dari hasil uji coba 1, bisa dilihat bahwa nilai PSNR meningkat. Nilai rata-rata PSNR meningkat sebesar 18,6% dari HS dan meningkat sebesar 2,29% dari NS.

5.2.2 Hasil Uji Coba 2

Uji coba 2 dilakukan dengan cara menyisipkan *payload* sebesar 240.000.000 bit ke dalam 5 video. Uji coba ini dimaksudkan untuk mengetahui jumlah maksimal *payload* yang bisa dimasukkan dalam proses penyisipan. Hasil uji coba ditunjukkan oleh Tabel 5.5.

Tabel 5.5 Jumlah Maksimal Payload

video	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	18.318.152	546.906	9.490.784
coastguard	2.924.303	519.606	1.913.795
foreman	5.096.597	559.344	2.793.449
hall monitor	8.852.393	542.417	4.597.635
mother daughter	13.221.383	584.954	6.954.332

Berdasarkan Tabel 5.5 dapat dilihat bahwa terjadi peningkatan jumlah maksimal *payload* yang bisa dimasukkan.

Berikut perbandingan peningkatan jumlah maksimal *payload* yang bisa dimasukkan:

- Jumlah maksimal *payload* video akiyo meningkat sebanyak 32,49 kali dari HS dan meningkat sebanyak 0,93 kali dari NS.
- Jumlah maksimal *payload* video coastguard meningkat sebanyak 4,63 kali dari HS dan meningkat sebanyak 0,53 kali dari NS.
- Jumlah maksimal *payload* video foreman menurun sebanyak 8,11 kali dari HS dan meningkat sebanyak 0,82 kali dari NS.
- Jumlah maksimal *payload* video hall monitor meningkat sebanyak 15,32 kali dari HS dan meningkat sebanyak 0,93 kali dari NS.
- Jumlah maksimal *payload* video mother daughter meningkat sebanyak 21,6 kali dari HS dan meningkat sebanyak 0,9 kali dari NS.

Dapat dilihat bahwa rata-rata jumlah maksimal *payload* yang bisa dimasukkan meningkat sebanyak 16,58 kali dari HS dan meningkat sebanyak 0,88 kali dari NS.

5.2.3 Hasil Uji Coba 3

Uji coba 3 dilakukan dengan cara mengekstraksi stego video memakai location map yang telah dibuat saat proses penyisipan. Uji coba 3 dilakukan untuk mengetahui apakah video hasil ekstraksi bisa kembali seperti sebelum di lakukan penyisipan. Hasil uji coba ditunjukkan oleh Tabel 5.6, Tabel 5.7, Tabel 5.8, Tabel 5.9, Tabel 5.10, Tabel 5.11, Tabel 5.12, dan Tabel 5.13.

Tabel 5.6 PSNR Hasil Ekstraksi dengan Payload 10.000 bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	Inf	Inf	Inf
coastguard	Inf	Inf	Inf
foreman	Inf	Inf	Inf
hall monitor	Inf	Inf	Inf
mother daughter	Inf	Inf	Inf

Tabel 5.7 PSNR Hasil Ekstraksi dengan Payload 50.000 bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	Inf	Inf	Inf
coastguard	Inf	Inf	Inf
foreman	Inf	Inf	Inf
hall monitor	Inf	Inf	Inf
mother daughter	Inf	Inf	Inf

Tabel 5.8 PSNR Hasil Ekstraksi dengan Payload 100.000 bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	Inf	Inf	Inf
coastguard	Inf	Inf	Inf
foreman	Inf	Inf	Inf
hall monitor	Inf	Inf	Inf
mother daughter	Inf	Inf	Inf

Tabel 5.9 PSNR Hasil Ekstraksi dengan Payload 240.000.000 bit

video	PSNR (dB)		
	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	Inf	Inf	Inf
coastguard	Inf	Inf	Inf
foreman	Inf	Inf	Inf
hall monitor	Inf	Inf	Inf
mother daughter	Inf	Inf	Inf

Tabel 5.10 Nilai Korelasi Payload 10.000 bit

video	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	1	1	1
coastguard	1	1	1

foreman	1	1	1
hall monitor	1	1	1
mother daughter	1	1	1

Tabel 5.11 Nilai Korelasi Payload 50.000 bit

video	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	1	1	1
coastguard	1	1	1
foreman	1	1	1
hall monitor	1	1	1
mother daughter	1	1	1

Tabel 5.12 Nilai Korelasi Payload 100.000 bit

video	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	1	1	1
coastguard	1	1	1
foreman	1	1	1
hall monitor	1	1	1
mother daughter	1	1	1

Tabel 5.13 Nilai Korelasi Payload 240.000.000 bit

video	Metode yang diajukan	<i>Histogram Shifting</i> [5]	<i>Neighbour Similarity</i> [6]
akiyo	1	1	1

coastguard	1	1	1
foreman	1	1	1
hall monitor	1	1	1
mother daughter	1	1	1

Dapat dilihat dari Tabel 5.6, Tabel 5.7, Tabel 5.8, Tabel 5.9, Tabel 5.10, Tabel 5.11, Tabel 5.12, dan Tabel 5.13, bahwa video dan payload hasil penyisipan dapat kembali lagi seperti semula setelah dilakukan proses ekstraksi.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh dari Tugas Akhir yang telah dikerjakan dan saran tentang pengembangan dari Tugas Akhir ini yang dapat dilakukan di masa yang akan datang.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil uji coba dan evaluasi Tugas Akhir ini adalah sebagai berikut:

1. Dengan metode modifikasi *histogram shifting* dan *neighbouring similarity* untuk *payload* yang sama, nilai PSNR rata-rata meningkat sebanyak 18,01% dari metode HS dan meningkat sebanyak 2,1% dari NS. Tetapi masih ada penurunan nilai PSNR saat penyisipan video foreman dengan *payload* 10.000 bit.
2. Dengan metode modifikasi *histogram shifting* dan *neighbouring similarity*, jumlah *payload* yang bisa dimasukkan meningkat sangat besar. Rata-rata jumlah *payload* meningkat sebesar 16,58 kali dari HS dan meningkat sebanyak 0,88 kali dari NS.
3. Metode modifikasi ini bersifat *reversible*, sehingga video hasil penyisipan dapat dikembalikan seperti semula.

6.2 Saran

Saran yang diberikan dari hasil uji coba dan evaluasi Tugas Akhir ini adalah sebagai berikut:

1. Pengembangan metode agar *payload* yang bisa dimasukkan tidak hanya teks saja.
2. Perlu adanya peningkatan akurasi pemilihan pergeseran (*shifting*) bit sehingga menghasilkan nilai PSNR yang lebih baik saat jumlah *payload* sangat sedikit.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] M. S. Subhedar dan V. H. Mankar, "Current status and key issues in image steganography: A survey," *Computer Science Review*, 2014.
- [2] "Xiph.org :: Derf's Test Media Collection," [Online]. Available: <https://media.xiph.org/video/derf/>. [Diakses 30 Desember 2017].
- [3] "Random Binary Number Generator - Create Random Bin Digits - Online - Browserling Web Developer Tools," Browserling, [Online]. Available: <https://www.browserling.com/tools/random-bin>. [Diakses 4 July 2018].
- [4] I. E. G. Richardson, H. 264 and MPEG-4 video compression: video coding for, John Wiley & Sons, 2003.
- [5] Z. Ni, Y.-Q. Shi, N. Ansari dan W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354 - 362, 2006.
- [6] H.-L. Yeh, S.-T. Gue, P. Tsai dan W.-K. Shih, "Reversible video data hiding using neighbouring similarity," *IET Signal Processing*, vol. 8, no. 6, pp. 579 - 587, 2014.
- [7] "MATLAB & Simulink," The MathWorks, Inc., [Online]. Available: <https://au.mathworks.com/products/matlab.html>. [Diakses 6 Juny 2018].
- [8] S. Wolf dan M. Pinson, "Reference Algorithm for Computing Peak Signal to Noise Ratio (PSNR) of a Video," 2009.
- [9] D. M. Firmansyah dan T. Ahmad, "An improved neighbouring similarity method for video steganography," dalam *Cyber and IT Service Management, International Conference on*, Bandung, 2016.

- [10] M. H. A. Al-Hooti, S. Djanali dan T. Ahmad, "Audio Data Hiding Based on Sample Value Modification," *Journal Of Informatin Processing Systems*, 2015.

BIODATA PENULIS



Alek Nur Fatman lahir di Trenggalek pada tanggal 29 Agustus 1995. Penulis menempuh pendidikan mulai dari RA Nurudz Dzolam (2000-2002), MI Nurudz Dzolam (2002-2008), SMPN 1 Pogalan (2008-2011), SMAN 1 Trenggalek (2011-2014), dan melanjutkan pendidikan S1 di Departemen Informatika ITS(2014-2018). Penulis mengambil bidang minat Komputasi Berbasis Jaringan (KBJ) saat berkuliah di Departemen Informatika. Penulis pernah kerja praktik di PT. Petrokimia pada Januari 2018. Komunikasi dengan penulis dapat melalui email aleknurfatman95@gmail.com.