



**TESIS - KI142502**

**PENINGKATAN KINERJA METODE  
STEGANOGRAPHY PADA VIDEO DIGITAL  
BERBASIS NEIGHBOURING SIMILARITY**

**DIKSY MEDIA FIRMANSYAH**

**NRP. 5114201004**

**DOSEN PEMBIMBING**

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

**NIP. 19750525 200312 1 002**

**PROGRAM MAGISTER**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA**

**2016**





THESES - KI142502

**AN IMPROVED STEGANOGRAPHY METHOD FOR  
DIGITAL VIDEO BASED ON NEIGHBOURING  
SIMILARITY**

DIKSY MEDIA FIRMANSYAH

NRP. 5114201004

SUPERVISOR

Tohari Ahmad, S.Kom., MIT., Ph.D.

NIP. 19750525 200312 1 002

MASTER PROGRAMS

INFORMATICS ENGINEERING

FACULTY OF INFORMATION TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2016



Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom.)  
di  
Institut Teknologi Sepuluh Nopember

oleh:  
DIKSY MEDIA FIRMANSYAH  
NRP. 5114201004

Dengan Judul:  
PENINGKATAN KINERJA METODE STEGANOGRAPHY PADA  
VIDEO DIGITAL BERBASIS NEIGHBOURING SIMILARITY

Tanggal Ujian : 21-6-2016  
Periode Wisuda : 2015 Genap

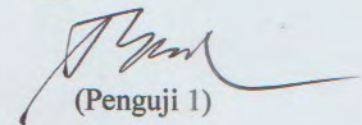
Disetujui oleh:

Tohari Ahmad, S.Kom, MIT, Ph.D  
NIP. 197505252003121002



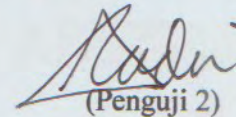
(Pembimbing 1)

Prof.Ir.Supeno Djanali, M.Sc, Ph.D  
NIP. 194806191973011001



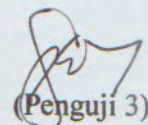
(Penguji 1)

Dr.Eng. Radityo Anggoro, S.Kom, M.Sc  
NIP. 1984101620081210002



(Penguji 2)

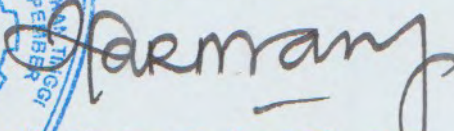
Hudan Studiawan, S.Kom, M.Kom  
NIP. 198705112012121003



(Penguji 3)



Direktur Program Pasca Sarjana,



Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.  
NIP. 196012021987011001



**LEMBAR PERNYATAAN  
PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : DIKSY MEDIA FIRMANSYAH  
Nrp. : 5114201004  
Jurusan / Fak. : Teknik Informatika / Teknologi Informasi  
Alamat kontak : Jl. Rangu III K-13, Jember, Jawa Timur  
a. Email : next.rasmus@gmail.com  
b. Telp/HP : 0822 4557 6268

Menyatakan bahwa semua data yang saya *upload* di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

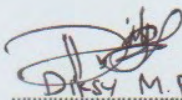
Demikian perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti-Free Right)** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

An Improved Neighbouring Similarity Method for Video Steganography

Dengan Hak Bebas Royalti Non-Eksklusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Surabaya  
Pada tanggal : 27 Juni 2016  
Yang menyatakan,

  
Diksy M.F.

Nrp. 5114201004



**KETERANGAN :**

Tanda tangan pembimbing wajib dibubuhi stempel jurusan.

Form dicetak dan diserahkan di bagian Pengadaan saat mengumpulkan hard copy TA/Tesis/Disertasi.





# Peningkatan Kinerja Metode Steganography pada Video Digital Berbasis Neighbouring Similarity

Nama Mahasiswa : Diksy Media Firmansyah

NRP : 5114201004

Pembimbing : Tohari Ahmad, S.Kom., MIT., Ph.D.

## ABSTRAK

Video digital adalah media yang saat ini mulai digunakan untuk keperluan *steganography*. Salah satu metode *steganography* untuk video digital adalah *neighbouring similarity*. *Neighbouring similarity* merupakan turunan dari metode *histogram shifting* yang mampu menghasilkan kapasitas lebih baik. Alasannya adalah karena metode *neighbouring similarity* mampu memanfaatkan kemiripan antar *frame* untuk membangkitkan suatu media penyisipan, yang kemudian disebut dengan *prediction error*. Tetapi, proses pembangkitan *prediction error* pada metode tersebut masih memiliki banyak kemungkinan untuk dikembangkan.

Penelitian ini mengusulkan tiga buah metode penyisipan yang merupakan pengembangan dari metode *neighbouring similarity*. Pengembangan yang dilakukan adalah dengan memodifikasi arsitektur pembangkitan *prediction error*. Metode pertama menggunakan tiap tiga *frame* untuk menghasilkan dua *prediction error*. Metode kedua menggunakan *prediction error* bertingkat untuk menyisipkan pesan rahasia. Metode ketiga menggunakan skema *overlap* untuk memaksimalkan jumlah *prediction error* yang dibangkitkan.

Berdasarkan uji coba yang dilakukan, dua dari tiga metode yang diusulkan mampu menghasilkan kapasitas yang lebih tinggi dibandingkan metode *neighbouring similarity*. Metode usulan pertama mampu meningkatkan kapasitas hingga 33,56% pada video *grayscale* dan 33,68% pada video berwarna. Metode usulan ketiga mampu meningkatkan kapasitas hingga 100,52% pada video *grayscale* dan 101,14% pada video berwarna. Untuk pengukuran PSNR, ketiga metode usulan menghasilkan nilai di bawah metode *neighbouring similarity*, tetapi masih mampu mempertahankan nilainya diatas 40 dB.

Kata Kunci: *reversible steganography*, *video steganography*, *histogram shifting*, *neighbouring similarity*, *prediction error*

*[Halaman ini sengaja dikosongkan]*

# **An Improved Steganography Method for Digital Video Based On Neighbouring Similarity**

Student Name : Diksy Media Firmansyah  
NRP : 5114201004  
Supervisor : Tohari Ahmad, S.Kom., MIT., Ph.D.

## **ABSTRACT**

Digital video is a media that used for steganography necessities now. One of steganography method for digital video is neighbouring similarity. Neighbouring similarity is a derived method from histogram shifting method that can be increased a capacity. A reason is neighbouring similarity method can utilize a similarity of digital video frames to generates an embedding media, that called prediction error. But, prediction error generation process in neighbouring similarity method has a lot of possibilities to improved.

This research proposes three embedding method that inspired by neighbouring similarity. A contribution is modifying the architecture of prediction error generating process. First proposed method uses three frames to generate two prediction error. Second proposed method uses graded of prediction error to embed a secret message. Third proposed method uses overlapped scheme to maximizing amount of generated prediction error.

From the experiment, two of three proposed methods is able to increased a capacity from neighbouring similarity method. First proposed method increases the capacity by 33.56% in grayscale video and 33.68% in colored video. Second proposed method increases the capacity by 100,52% in grayscale video and 101,14% in colored video. For PSNR measurement, all proposed method are producing value below neighbouring similarity method. But, all proposed method can preserve the value above 40 dB.

Keywords: reversible steganography, video steganography, histogram shifting, neighbouring similarity, prediction error

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Segala puji syukur kepada Allah SWT. atas ridho dan segala pemberianNya sehingga penulis mampu menyelesaikan tesis dengan judul "Peningkatan Kinerja Metode Steganography pada Video Digital Berbasis Neighbouring Similarity" ini. Shalawat serta salam disampaikan juga kepada Rasulullah Muhammad SAW. yang dengan segala ketulusannya bersedia menyampaikan ajaran-ajaran Islam sehingga bisa sampai kepada penulis. Walaupun penulis belum pernah bertemu secara langsung, penulis mencoba menjadikan beliau sebagai rujukan dalam segala tindakan berdasarkan referensi-referensi yang ada dan dapat diterima.

Penulis juga mengucapkan terima kasih kepada seluruh pihak yang telah mendukung proses penyelesaian tesis ini, khususnya kepada:

1. Kedua orang tua, Bapak dan Ibu yang selalu memberikan dukungan moral, doa sampai finansial kepada penulis sehingga proses perkuliahan maupun penyelesaian tesis ini dapat terus dapat terus berjalan. Terima kasih Bapak dan Ibu atas segala dukungan terhadap anakmu ini.
2. Bapak Tohari Ahmad yang telah memberikan bimbingan dan berbagai tantangan dalam menyelesaikan tesis ini sehingga tidak hanya sebagai syarat kelulusan studi S2, tetapi penulis juga mampu memahami bidang keilmuan yang dibahas pada tesis ini.
3. Prof. Supeno Djanali, Bapak Radityo Anggoro, dan Bapak Hudan Studiawan selaku penguji yang telah bersedia memberikan koreksi dan masukan terhadap tesis ini.
4. Teman-teman S2 FTif yang tidak dapat disebutkan satu per satu. Terima kasih atas segala pengalaman dan bantuan yang diberikan terhadap penulis.

Penulis menyadari bahwa tesis ini masih jauh dari kesempurnaan. Oleh karena itu, masukan dan saran yang bersifat membangun selalu dinantikan untuk perbaikan di masa mendatang. Akhirnya, penulis berharap agar tesis ini mampu memberikan kontribusi yang bermanfaat bagi bidang keilmuan di kemudian hari.

Surabaya, Juni 2016

Diksy M. Firmansyah

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xv
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	5
1.3 Hipotesis .....	5
1.4 Batasan Masalah .....	5
1.5 Tujuan Penelitian .....	6
1.6 Manfaat Penelitian .....	6
1.7 Kontribusi Penelitian .....	6
1.8 Sistematika Penulisan .....	7
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA .....	9
2.1 Video Digital .....	9
2.2 Steganography .....	10
2.3 Difference Expansion .....	11
2.3 Histogram Shifting .....	12
2.3.1 Proses Penyisipan .....	13
2.3.2 Proses Ekstraksi dan <i>Recovery</i> .....	15
2.4 Neighbouring Similarity .....	16
2.4.1 Proses Penyisipan .....	16
2.4.2 Proses Ekstraksi dan <i>Recovery</i> .....	19
2.5 Peak to Signal Noise Ratio (PSNR) .....	20
BAB 3 METODE PENELITIAN .....	23
3.1 Tahapan Penelitian .....	23
3.2 Studi Literatur .....	24
3.3 Perancangan Algoritma .....	24
3.3.1 Neighbouring Similarity of Triplets .....	25
3.3.2 Multi-Level Neighbouring Similarity .....	30

3.3.3 Overlapped Neighbouring Similarity .....	35
3.4 Implementasi Penelitian.....	40
3.5 Perancangan Uji Coba dan Analisis Hasil .....	40
BAB 4 HASIL DAN PEMBAHASAN .....	43
4.1 Lingkungan Uji Coba.....	43
4.2 Pelaksanaan Uji Coba .....	43
4.3 Hasil dan Evaluasi .....	46
4.3.1 Skenario 1 .....	46
4.3.2 Skenario 2.....	49
BAB 5 KESIMPULAN DAN SARAN .....	57
5.1 Kesimpulan .....	57
5.2 Saran .....	57
DAFTAR PUSTAKA.....	59
BIODATA PENULIS.....	63



## DAFTAR GAMBAR

Gambar 1.1. Ilustrasi pembagian subtopik <i>data hiding</i> . .....	3
Gambar 2.1. Ilustrasi sampel spasial dan temporal.....	9
Gambar 2.2. Ilustrasi proses <i>steganography</i> . .....	10
Gambar 2.3. Contoh proses pembangkitan <i>histogram</i> . .....	14
Gambar 2.4. Contoh hasil penggeseran <i>histogram</i> . .....	14
Gambar 2.5. Contoh <i>histogram</i> setelah disisipi <i>secret message</i> . .....	15
Gambar 2.6. Ilustrasi proses pembangkitan <i>prediction error</i> . .....	17
Gambar 2.7. Contoh keluaran PSNR menggunakan FFmpeg.....	20
Gambar 3.1. Tahapan Penelitian. ....	23
Gambar 3.2. Ilustrasi pembangkitan <i>histogram</i> pada metode usulan pertama. ....	26
Gambar 3.3. Ilustrasi pembangkitan <i>prediction error</i> pada metode usulan kedua. .....	30
Gambar 3.4. Ilustrasi pembangkitan <i>predicton error</i> pada metode usulan ketiga.	36
Gambar 4.1. Contoh <i>frame</i> pada tiap video uji. ....	44
Gambar 4.2. Hasil pengukuran kapasitas untuk <i>Proposed 1</i> . ....	52
Gambar 4.3. Hasil pengukuran PSNR untuk <i>Proposed 1</i> . ....	52
Gambar 4.4. Hasil pengukuran kapasitas untuk <i>Proposed 2</i> . ....	53
Gambar 4.5. Hasil pengukuran PSNR untuk <i>Proposed 2</i> . ....	53
Gambar 4.6. Hasil pengukuran kapasitas untuk <i>Proposed 3</i> . ....	54
Gambar 4.7. Hasil pengukuran PSNR untuk <i>Proposed 3</i> . ....	54

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 4.1. Hasil pengukuran kapasitas skenario pertama. ....	47
Tabel 4.2. Hasil pengukuran PSNR skenario kedua. ....	48
Tabel 4.3. Hasil pengukuran kapasitas skenario kedua.....	49
Tabel 4.4. Hasil pengukuran PSNR skenario kedua. ....	50

*[Halaman ini sengaja dikosongkan]*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Video digital, atau yang sering disebut dengan istilah *motion picture*, merupakan salah satu media yang sering digunakan di industri hiburan saat ini (Tew & Wong, 2014). Penggunaan istilah *motion picture* adalah karena pada dasarnya, video adalah kumpulan dari citra digital yang ditampilkan secara bergantian dalam kurun waktu tertentu. Citra digital yang saling berdekatan akan memiliki tingkat kemiripan / *similarity* yang tinggi sehingga obyek pada citra digital terlihat seolah-olah bergerak atau berubah secara alami.

Seiring dengan semakin seringnya penggunaan video digital, muncul berbagai isu terkait penggunaan video digital sebagai media komunikasi. Karena video digital merupakan pengembangan dari citra digital, maka banyak isu terkait citra digital yang juga berkembang untuk video digital. Salah satu isu tersebut adalah keamanan sistem. Isu keamanan sistem yang menggunakan citra digital sebagai obyek penelitian kini mulai dikembangkan untuk video digital. Isu keamanan sistem untuk video digital ini kemudian berkembang menjadi sebuah topik penelitian yang cukup populer saat ini.

Topik keamanan sistem dibagi menjadi dua subtopik, yaitu *cryptology* dan *data hiding* (Chanu, et al., 2012). Tujuan dari keduanya sama, yaitu bagaimana menyembunyikan suatu informasi agar hanya dapat dimanfaatkan oleh pihak yang memiliki hak. Namun, pendekatan yang dilakukan keduanya berbeda. *Cryptology* menyembunyikan informasi dengan cara mentransformasikan informasi itu sendiri sehingga informasi yang terlihat seolah-olah merupakan informasi acak yang tidak memiliki arti. *Data hiding* menyembunyikan informasi dengan cara menyisipkannya pada informasi yang lain sehingga informasi utama yang terlihat adalah informasi lain tersebut.

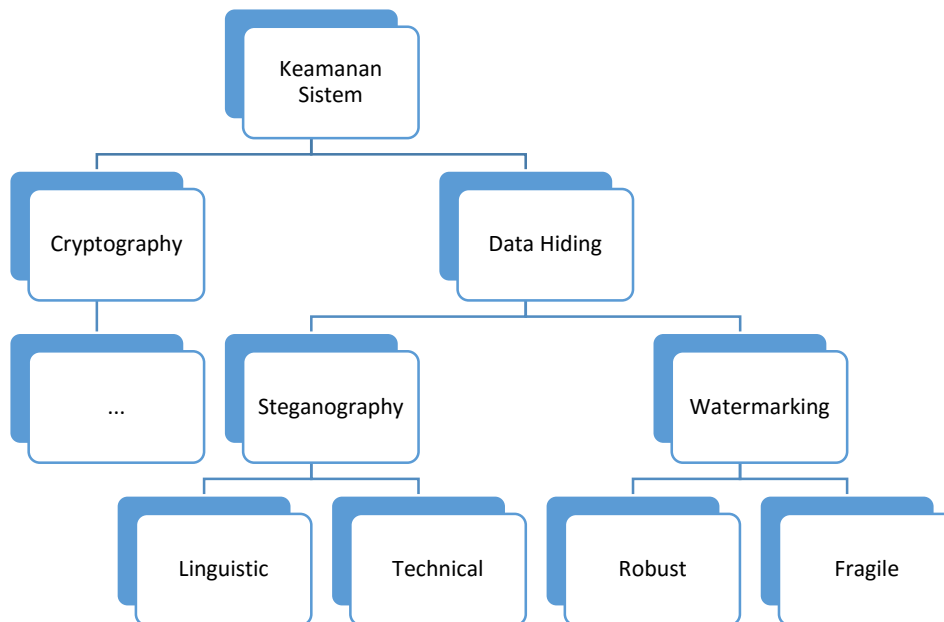
Berdasarkan kemampuan untuk memulihkan informasi lain yang digunakan sebagai media penyisipan, *data hiding* dibagi menjadi dua, yaitu *irreversible data hiding* dan *reversible data hiding* (Bansal & Gupta, 2014). *Irreversible data hiding* adalah metode *data hiding* di mana informasi lain yang digunakan akan mengalami kerusakan dan tidak dapat dipulihkan lagi. *Reversible data hiding* adalah metode *data hiding* di mana informasi lain yang digunakan akan mengalami kerusakan tetapi dapat dipulihkan kembali setelah informasi rahasianya dikeluarkan.

Terdapat empat poin pengujian untuk mengukur kualitas sebuah metode *Data Hiding*, yaitu kemiripan, kapasitas, *auxiliary information*, dan masalah *underflow / overflow*. Penjelasan singkat dari keempat poin tersebut adalah sebagai berikut:

- a. Kemiripan: Cara kerja pendekatan *data hiding* adalah dengan menyisipkan suatu informasi rahasia (*secret message*) ke informasi yang lain (*cover data*). Proses tersebut akan menghasilkan suatu informasi baru (*stego data*) yang mirip dengan *cover data*. Semakin mirip *stego data* yang dihasilkan terhadap *cover data*-nya, maka semakin baik metode tersebut. Banyak metode yang dapat digunakan untuk mengukur tingkat kemiripan tersebut. Salah satu metode yang cukup populer adalah *Peak to Signal Noise Ratio* (PSNR).
- b. Kapasitas: jumlah maksimum *secret message* yang dapat ditampung pada suatu *cover data*. Biasanya disajikan dalam satuan bit. Semakin tinggi nilai kapasitas yang dihasilkan, maka semakin baik suatu metode *data hiding*.
- c. *Auxiliary Information*: informasi tambahan yang harus diketahui pada proses ekstraksi *secret message*. Informasi tambahan ini akan menimbulkan masalah baru pada proses *data hiding*, yaitu bagaimana mendistribusikannya pada penerima pesan. Karena itu, semakin sedikit *auxiliary information* yang dihasilkan, maka semakin baik suatu metode *data hiding*.
- d. *Underflow / Overflow problem*: Setiap informasi digital memiliki batas nilai. Sebagai contoh, nilai piksel pada sebuah citra digital *grayscale* 8 bit akan berada pada rentang 0 hingga 255. Jika sebuah metode *data hiding* mengubah nilai piksel hingga berada di bawah nilai nol, maka masalah ini disebut dengan *underflow*. Jika perubahan nilainya membuat nilai piksel berada di bawah nilai 255, maka disebut dengan *overflow*. Baik *underflow* ataupun

*overflow*, kedua masalah ini tidak boleh sampai terjadi karena akan menyebabkan kegagalan dalam proses penyisipan.

*Data hiding* itu sendiri dibagi menjadi dua subtopik lagi, yaitu *steganography* dan *watermarking* (Subhedar & Mankar, 2014). Perbedaan utama dari kedua subtopik tersebut adalah tujuan penyisipan informasi rahasia. *Steganography* menyisipkan informasi rahasia agar hanya *cover data*-nya saja yang dapat terlihat. *Watermarking* menyisipkan informasi rahasia pada *cover data* agar keduanya dapat terlihat. Fokus penelitian ini adalah pada subtopik *steganography*. Ilustrasi pembagian subtopik *data hiding* dapat dilihat pada Gambar 1.1 di bawah ini.



**Gambar 1.1. Ilustrasi pembagian subtopik data hiding.**

Salah satu media yang populer digunakan pada steganography adalah citra digital. Hal ini dikarenakan citra digital memiliki tingkat *redundancy* yang tinggi. Kemungkinan nilai piksel pada citra digital adalah 0 hingga 255, tetapi jumlah piksel yang ada pada sebuah citra digital biasanya lebih banyak dari itu. Untuk citra digital dengan resolusi VGA saja, jumlah piksel yang ada mencapai 307.200 buah. Artinya, kemungkinan besar akan ada banyak piksel yang bernilai sama.

*Difference Expansion* (DE) (Tian, 2003) adalah salah satu metode yang cukup populer di lingkungan *steganography* pada citra digital. Karena kepopulerannya, metode ini telah banyak dikembangkan. Salah satunya adalah pengembangan arsitektur pada pengambilan nilai selisih dari tiap dua piksel menjadi tiap tiga piksel (Alattar, 2003). Kemudian, metode itu dikembangkan lagi dari tiga piksel menjadi empat piksel (Alattar, 2004). Perubahan arsitektur juga dilakukan pada suatu penelitian untuk menghilangkan penggunaan *location map* (Lin, et al., 2008).

Selain arsitektur, pengembangan DE dilakukan terhadap fungsi transformasi yang digunakan (Lou, et al., 2009). Transformasi integer yang digunakan dimodifikasi menggunakan fungsi logaritma, yang kemudian disebut dengan *Reduced Difference Expansion* (RDE). Lebih jauh lagi, pengembangan fungsi transformasi tersebut diimplementasikan pada pengembangan arsitektur yang telah ada (Ahmad, et al., 2013). Pengembangan-pengembangan tersebut telah terbukti mampu meningkatkan kualitas metode DE.

*Histogram Shifting* (Ni, et al., 2006) juga merupakan sebuah metode *steganography* pada citra digital yang cukup populer. Metode ini juga telah banyak berkembang. Contoh perkembangannya adalah penggunaan *framework* untuk menghasilkan kapasitas yang sesuai dengan kebutuhan (Li, et al., 2013). Bahkan, metode ini telah diadaptasikan untuk *cover data* berupa video digital. Salah satu pengembangan *histogram shifting* untuk video digital adalah metode *neighbouring similarity* (Yeh, et al., 2014). Metode *Neighbouring similarity* memodifikasi proses pembangkitan *histogram* pada metode *histogram shifting* sehingga *histogram* yang didapatkan menjadi lebih baik.

Penelitian ini mencoba mengadaptasikan sejarah perkembangan DE untuk mengembangkan metode *neighbouring similarity*. Jika perubahan arsitektur pengambilan nilai selisih mampu meningkatkan kapasitas metode DE, maka penelitian ini mencoba melakukan perubahan terhadap arsitektur pembangkitan *prediction error* pada metode *neighbouring similarity*. Perubahan arsitektur tersebut diharapkan mampu menghasilkan kapasitas yang lebih baik dibandingkan metode *neighbouring similarity*.



## 1.2 Perumusan Masalah

Permasalahan pada penelitian ini dapat dirumuskan menjadi:

- a. Bagaimana mengembangkan metode penyisipan yang memiliki kapasitas yang lebih baik dibandingkan dengan metode *neighbouring similarity*?
- b. Bagaimana mempertahankan tingkat kemiripan / *similarity* metode usulan agar tetap berada dalam kategori kualitas tinggi?
- c. Bagaimana mengatasi permasalahan yang muncul akibat melakukan modifikasi pembangkitan *prediction error*?

## 1.3 Hipotesis

Kapasitas yang dihasilkan oleh metode turunan *histogram shifting* (termasuk metode *neighbouring similarity*) adalah berdasarkan nilai *peak point* yang dihasilkan pada *histogram*. Semakin tinggi nilai *peak point*, maka semakin besar kapasitas yang dihasilkan. Metode *neighbouring similarity* sendiri membangkitkan *histogram* berdasarkan *prediction error* yang dibangkitkan. Oleh karena itu, modifikasi terhadap proses pembangkitan *prediction error* diharapkan mampu meningkatkan kapasitas yang dihasilkan sekaligus mempertahankan tingkat kemiripan pada kategori kualitas tinggi.

## 1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

- a. Data uji berupa contoh video digital yang disediakan oleh [xiph.org](http://xiph.org) pada kategori *derf collection* (xiph.org, 1994).
- b. Jumlah video uji yang digunakan adalah 20 video, di mana sebagian besar dipilih karena digunakan juga pada penelitian sebelumnya. Sisanya dipilih secara acak.

- c. Video uji di-*format* ulang menggunakan codec MJPEG pada perangkat lunak FFmpeg (Bellard, 2000).
- d. Pesan rahasia yang digunakan adalah teks *Lorem Ipsum* (Signale, 2001) dengan panjang 3.971 karakter ASCII (sekitar 27.797 bit jika satu karakter diwakili 7 bit).

### **1.5 Tujuan Penelitian**

Berdasarkan perumusan masalah pada subbab 1.2, tujuan dari penelitian ini adalah sebagai berikut:

- a. Mengembangkan metode *neighbouring similarity* agar menghasilkan kapasitas yang lebih besar.
- b. Mempertahankan tingkat kemiripan / *similarity* yang dihasilkan metode usulan agar tetap berada pada kategori kualitas tinggi.
- c. Mengusulkan solusi untuk permasalahan yang muncul pada usulan proses pembangkitan *prediction error*.

### **1.6 Manfaat Penelitian**

Manfaat dari penelitian ini adalah mengusulkan metode *steganography* untuk video digital yang memiliki kemampuan *reversible*, yaitu kemampuan untuk mengembalikan kondisi video digital seperti sebelum disisipi pesan rahasia. Metode ini dapat digunakan pada video digital untuk berbagai keperluan, seperti pengiriman pesan rahasia, pemberian informasi hak cipta pada suatu video digital, penyisipan informasi yang bersifat *privacy* pada video digital untuk keperluan medis atau militer, dan berbagai keperluan lainnya.

### **1.7 Kontribusi Penelitian**

Kontribusi dari penelitian ini adalah mengusulkan tiga buah metode penyisipan baru yang merupakan pengembangan dari metode *neighbouring*

*similarity*, yaitu *neighbouring similarity of triplet*, *multi-level neighbouring similarity*, dan *overlapped neighbouring similarity*.

## **1.8 Sistematika Penulisan**

Penulisan tesis ini memiliki sistematika sebagai berikut:

- a. BAB 1 PENDAHULUAN. Bab ini berisi pendahuluan yang menjelaskan latar belakang masalah, perumusan masalah, tujuan, manfaat, kontribusi penelitian dan sistematika penulisan terkait dengan penelitian.
- b. BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA. Bab ini berisi tinjauan pustaka yang mengacu pada dasar-dasar teori terkait penelitian, diantaranya *video digital*, *steganography*, *difference expansion*, *histogram shifting*, *neighbouring similarity* dan *peak to signal noise ratio (PSNR)*.
- c. BAB 3 METODOLOGI PENELITIAN. Bab ini berisi tahapan penelitian meliputi studi literatur, perancangan algoritma, implementasi penelitian, perancangan uji coba dan analisis hasil. Bab ini juga menyertakan jadwal pelaksanaan penelitian.
- d. BAB 4 IMPLEMENTASI DAN PEMBAHASAN. Bab ini berisi implementasi penelitian dan analisis terhadap implementasi penelitian tersebut.
- e. BAB 5 PENUTUP. Bab ini berisi kesimpulan dan saran untuk pengembangan tesis.

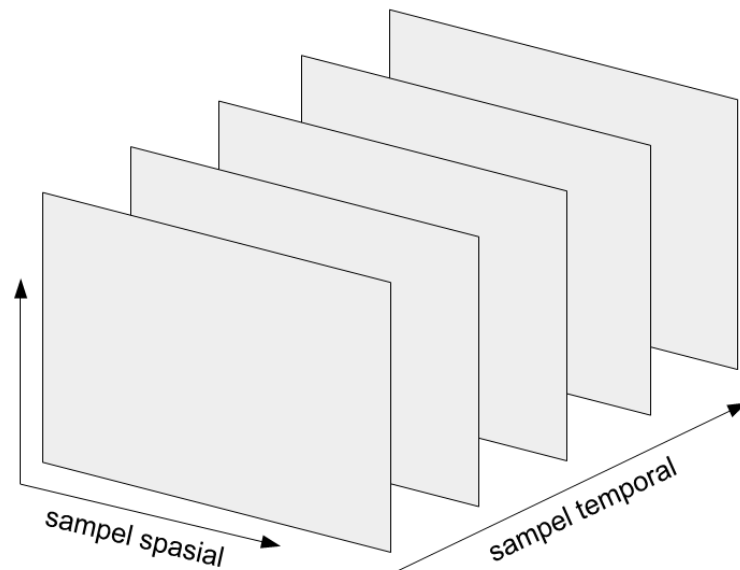
*[Halaman ini sengaja dikosongkan]*

## BAB 2

### DASAR TEORI DAN KAJIAN PUSTAKA

#### 2.1 Video Digital

Video digital adalah representasi dari adegan visual alami (dunia nyata), sampel spasial dan sampel temporal (Richardson, 2003). Adegan visual diambil pada suatu titik waktu tertentu untuk menghasilkan sebuah sampel spasial (*frame*). Proses pengambilan adegan visual secara terus menerus dalam kurun waktu tertentu akan menghasilkan beberapa *frame*, yang kemudian disebut dengan sampel temporal. Ilustrasi dari sampel spasial dan sampel temporal dapat dilihat pada Gambar 2.1.



**Gambar 2.1. Ilustrasi sampel spasial dan temporal.**

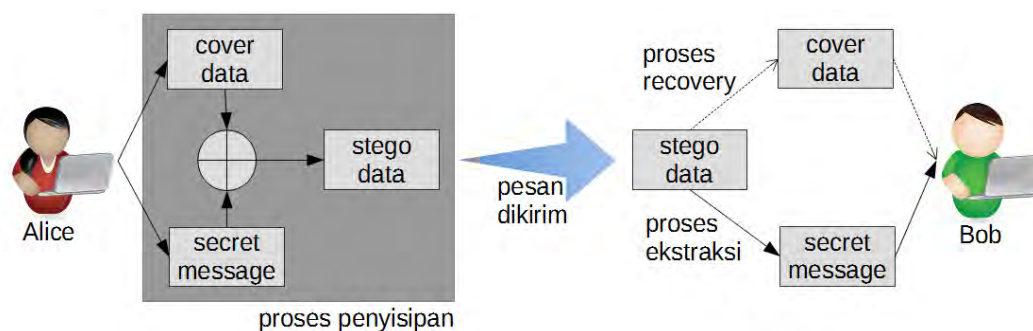
Suatu obyek pada video digital dapat terlihat seolah-olah bergerak karena adanya kemiripan antar *frame* yang berdekatan secara temporal. *Frame* yang mirip tersebut ditampilkan secara bergantian dalam kurun waktu tertentu sehingga obyek tersebut akan terlihat berpindah tempat dari satu titik ke titik yang lain secara alami.

Perubahan yang terlihat alami tersebut tidak hanya untuk posisi benda, tetapi juga bentuk dan warna. Semakin banyak *frame* yang dapat diambil dalam satu waktu, maka semakin terlihat alami sebuah video.

## 2.2 Steganography

*Steganography* adalah salah satu subtopik dari *information hiding* yang merupakan bagian dari keamanan sistem, di mana bagian yang lain adalah *cryptography* (Subhedar & Mankar, 2014). *Cryptography* dan *steganography* memiliki tujuan yang sama, yaitu bagaimana menyembunyikan suatu pesan agar pihak yang memiliki hak saja yang dapat membaca pesan tersebut. Tetapi, *cryptography* dan *steganography* memiliki pendekatan yang berbeda. *Cryptography* mentransformasikan pesan dengan suatu algoritma dan kunci tertentu sehingga pesan tersebut terlihat acak. *Steganography* menggunakan data lain sebagai kamufase sehingga hanya data lain tersebut yang terlihat.

*Steganography* memiliki tiga istilah yang sering digunakan, yaitu *cover data*, *secret message* dan *stego data*. *Cover data* adalah data lain yang digunakan sebagai media kamufase. *Secret message* adalah pesan rahasia yang ingin disimpan atau dikirimkan. *Stego data* adalah *cover data* yang telah disisipi pesan rahasia. Algoritma penyisipan digunakan untuk menyisipkan *secret message* ke dalam *cover data* sehingga menghasilkan *stego data*, sedangkan algoritma ekstraksi digunakan untuk mengeluarkan *secret message* dari *stego data*. Ilustrasi dari proses *steganography* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Ilustrasi proses *steganography*.

Secara umum, *steganography* dapat dibagi menjadi dua sama seperti pembagian *data hiding*, yaitu *irreversible* dan *reversible steganography* (Holil & Ahmad, 2015). *Irreversible steganography* adalah jenis dari *steganography* di mana *cover data* akan rusak setelah melalui proses penyisipan *secret message*. *Reversible steganography* adalah jenis lain dari *steganography* di mana *stego data* dapat dikembalikan menjadi *cover data* secara sempurna setelah melalui proses ekstraksi. Proses pengembalian *stego data* menjadi *cover data* biasa disebut dengan proses *recovery*.

Kualitas sebuah metode *steganography* diukur menggunakan kapasitas dan kemiripan. Kapasitas adalah jumlah maksimal *secret message* yang dapat ditampung pada suatu *cover data*, biasanya dalam satuan bit. Semakin besar kapasitas yang dihasilkan, maka semakin baik sebuah metode *steganography*. Kemiripan adalah tingkat kesamaan antara *stego data* dan *cover data*, biasanya dihitung menggunakan metode PSNR. Untuk metode PSNR, akan dijelaskan lebih lanjut pada subbab 2.5.

### **2.3 Difference Expansion**

*Difference Expansion* (DE) adalah salah satu metode *steganography* untuk *cover data* berupa citra digital yang cukup populer dikalangan para peneliti (Tian, 2003). Kemampuan *reversible*-nya lah yang menjadi salah satu alasan metode ini menjadi populer. DE mampu *recovery* kondisi citra digital setelah melakukan proses ekstraksi dengan memanfaatkan *location map* yang disertakan pada *payload*. Hal ini membuat DE cocok digunakan pada citra digital yang bersifat sensitif, seperti citra medis, peta militer, dan citra digital lainnya yang membutuhkan *cover data* secara utuh.

DE menggunakan nilai selisih antar pasangan piksel untuk menyisipkan *payload*. Proses penyisipan tersebut menggunakan suatu transformasi *integer* yang dapat dibalikkan prosesnya sehingga nilai dari data yang terlibat dapat

dikembalikan seperti semula. Bentuk transformasi *integer* yang dimaksud dapat dilihat pada Persamaan 2.1 di bawah ini.

$$l = \left\lfloor \frac{x + y}{2} \right\rfloor, h = x - y \quad (2.1)$$

Di mana  $l$  adalah nilai rata-rata,  $h$  adalah nilai selisih,  $x$  adalah piksel pertama, dan  $y$  adalah piksel kedua. Sedangkan proses pembalikannya dilakukan menggunakan Persamaan 2.2 di bawah ini.

$$x = l + \left\lfloor \frac{h + 1}{2} \right\rfloor, y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2.2)$$

Karena cukup populer, metode ini telah banyak dikembangkan untuk mendapatkan kualitas *stego data* yang lebih baik. Salah satu pengembangan yang cukup populer digunakan adalah dengan memodifikasi arsitektur pengambilan nilai selisih piksel. DE yang asli menggunakan tiap dua buah piksel *non-overlapped* untuk membangkitkan nilai selisih, dan beberapa pengembangannya menggunakan model *overlapped* untuk membangkitkan nilai selisih lebih dari dua piksel. Modifikasi tersebut terbukti mampu meningkatkan kapasitas yang dihasilkan oleh DE.

### 2.3 Histogram Shifting

Selain *difference expansion*, *histogram shifting* adalah salah satu metode *reversible steganography* untuk *cover data* berupa citra digital (Ni, et al., 2006). *Histogram shifting* memanfaatkan warna piksel dengan frekuensi paling banyak pada citra digital (yang disebut dengan *peak point*) untuk menyisipkan pesan rahasia. Oleh karena itu, semakin tinggi nilai *peak point* pada suatu citra digital, maka semakin banyak jumlah pesan rahasia yang dapat ditampung.

Perubahan nilai warna piksel pada metode *histogram shifting* adalah satu bit. Hal ini membuat tingkat kemiripan *stego data* yang dihasilkan terhadap *cover data* akan cukup bagus. Setiap menemukan nilai warna piksel yang sama dengan nilai *peak point*, maka metode ini akan menambahkan atau mengurangi dengan bit pesan rahasia ke- $n$ . Sedangkan untuk memastikan bahwa pesan rahasia dapat

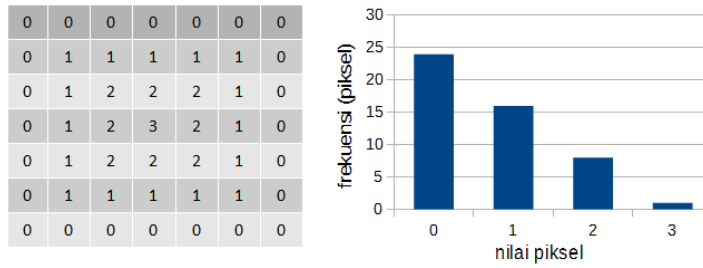


diekstraksi dan *cover data* dapat di-*recovery* dengan sempurna, metode *histogram shifting* harus memastikan bahwa nilai frekuensi warna di sebelah *peak point* harus bernilai nol. Untuk itu, dilakukanlah penggeseran nilai dari sebelah *peak point* ke warna piksel dengan frekuensi paling rendah (yang disebut dengan *zero point*).

Kelemahan dari metode *histogram shifting* adalah adanya *auxiliary information* yang dihasilkan dari proses penyisipan. *Auxiliary information* adalah informasi tambahan yang dibutuhkan untuk melakukan proses ekstraksi maupun *recovery*. Akibatnya, harus ada suatu prosedur pendistribusian *auxiliary information* agar dapat diketahui oleh penerima pesan. *Auxiliary information* pada metode *histogram shifting* berupa nilai *peak point* dan *zero point*. Tanpa kedua informasi tersebut, pesan rahasia yang tersimpan pada *stego data* tidak dapat diekstraksi dan *cover data* juga tidak dapat di-*recovery*.

### **2.3.1 Proses Penyisipan**

Pertama, seluruh piksel pada citra digital dipindai secara vertikal, horizontal, atau menggunakan algoritma tertentu. Nilai frekuensi warna piksel pada citra digital yang dibaca pada proses pemindaian digunakan sebagai referensi untuk membangkitkan *histogram*. Sumbu-x pada *histogram* mewakili nilai piksel pada citra digital (umumnya 0 sampai 255), sedangkan sumbu-y mewakili frekuensi nilai piksel. Nilai piksel yang digunakan dapat berupa komponen warna merah, hijau, biru, *grayscale*, atau berdasarkan algoritma tertentu. Setelah *histogram* dibuat, pilih nilai dengan frekuensi paling tinggi (*peak point*) dan nilai dengan frekuensi paling rendah (*zero point*). Contoh proses pembangkitan *histogram* dapat dilihat pada Gambar 2.3.

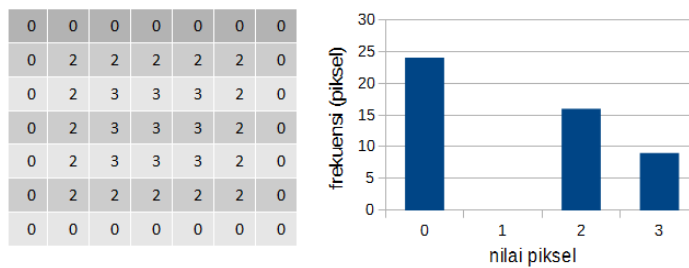


Gambar 2.3. Contoh proses pembangkitan *histogram*.

Kedua, nilai piksel yang berada di antara *peak point* dan *zero point* digeser ke arah *zero point*. Tujuan dari penggeseran ini adalah untuk mengosongkan nilai piksel di sebelah nilai *peak point*. Proses penggeseran dilakukan menggunakan Persamaan 2.3 di bawah ini.

$$\begin{aligned}
 \alpha_{x,y} &= \alpha_{x,y} + 1, \text{ jika } p < \theta, \alpha_{x,y} > p, \alpha_{x,y} < \theta \\
 \alpha_{x,y} &= \alpha_{x,y} - 1, \text{ jika } p > \theta, \alpha_{x,y} < p, \alpha_{x,y} > \theta
 \end{aligned}
 \tag{2.3}$$

Di mana  $\alpha$  adalah piksel,  $x$  dan  $y$  adalah koordinat dari piksel  $\alpha$ ,  $p$  adalah *peak point* dan  $\theta$  adalah *zero point*. Untuk memastikan *cover data* dapat di-*recovery*, maka dibuatlah sebuah informasi *overhead* untuk membedakan lokasi piksel yang bernilai *zero point* dan yang bernilai di sebelah *zero point*. Hal ini dikarenakan nilai piksel di sebelah *zero point* akan menjadi sama dengan *zero point* setelah proses penggeseran pada tahap ini. Jika piksel ke- $(x, y)$  adalah *zero point*, maka *overhead* ke- $m$  adalah 0. Sedangkan jika piksel ke- $(x, y)$  adalah nilai di sebelah *zero point*, maka *overhead* ke- $m$  adalah 1. Contoh hasil penggeseran *histogram* dapat dilihat pada Gambar 2.4.

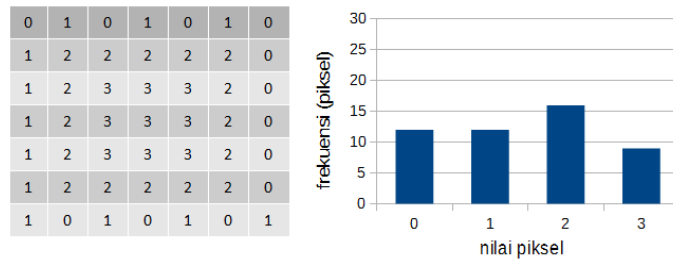


Gambar 2.4. Contoh hasil penggeseran *histogram*.

Ketiga, pesan rahasia disisipkan pada nilai piksel yang sama dengan *peak point*. Proses penyisipan dilakukan menggunakan Persamaan 2.4 di bawah ini.

$$\begin{aligned} \alpha_{x,y} &= \alpha_{x,y} + \beta_n, \text{ jika } p < z, \alpha_{x,y} = p \\ \alpha_{x,y} &= \alpha_{x,y} - \beta_n, \text{ jika } p > z, \alpha_{x,y} = p \end{aligned} \quad (2.4)$$

Di mana  $\beta$  adalah bit pesan rahasia ke- $n$ . Panjang bit pesan rahasia harus sama dengan nilai *peak point*. Jika panjang bit pesan rahasia kurang dari *peak point*, maka dapat ditambahkan bit 0 di depan pesan rahasia sebanyak *peak point* dikurang panjang bit pesan rahasia. Sedangkan jika panjang bit pesan rahasia lebih dari *peak point*, artinya *cover data* tidak dapat menampung seluruh bit pesan rahasia yang ada. Misalkan pesan rahasia yang akan disisipkan adalah 0101010101, maka contoh *histogram* hasil penyisipan pesan dapat dilihat pada Gambar 2.5.



Gambar 2.5. Contoh *histogram* setelah disisipi *secret message*.

### 2.3.2 Proses Ekstraksi dan *Recovery*

Pertama, nilai *peak point* dan *zero point* dari proses penyisipan harus diketahui terlebih dahulu. Kemudian, pindai seluruh piksel pada citra digital dengan pola yang sama seperti pada proses penyisipan. Proses ekstraksi pesan rahasia diselesaikan menggunakan Persamaan 2.5 di bawah ini.

$$\begin{aligned} \beta_n &= 0, \text{ jika } \alpha_{x,y} = p \\ \beta_n &= 1, \text{ jika } (\alpha_{x,y} = p + 1 \text{ dan } p < \theta) \text{ atau } (\alpha_{x,y} = p - 1 \text{ dan } p > \theta) \end{aligned} \quad (2.5)$$

Setelah pesan rahasia berhasil diekstraksi dengan sempurna, *histogram* dapat digeser ke bentuk semula menggunakan Persamaan 2.6 di bawah ini.

$$\begin{aligned} \alpha_{x,y} &= \alpha_{x,y} - 1, \text{ jika } p < \alpha_{x,y} < z \text{ atau } (\alpha_{x,y} = z \text{ dan } \delta_m = 1) \\ \alpha_{x,y} &= \alpha_{x,y} + 1, \text{ jika } p > \alpha_{x,y} > z \text{ atau } (\alpha_{x,y} = z \text{ dan } \delta_m = 0) \end{aligned} \quad (2.6)$$

Di mana  $\delta$  adalah informasi *overhead* ke- $m$ .

## 2.4 Neighbouring Similarity

*Neighbouring Similarity* adalah metode *reversible steganography* untuk *cover data* berupa video digital (Yeh, et al., 2014). Metode ini merupakan pengembangan dari metode *histogram shifting*. Perbedaan utama dari *neighbouring similarity* dan *histogram shifting* adalah pada proses pembangkitan *histogram*. *Histogram shifting* membangkitkan *histogram* berdasarkan nilai warna piksel pada citra digital atau pada *frame* jika diterapkan pada video digital, sedangkan *neighbouring similarity* membangkitkan *histogram* berdasarkan nilai selisih antar *frame* pada video digital yang disebut dengan *prediction error*.

Metode *neighbouring similarity* mampu menghasilkan *histogram* dengan nilai *peak point* yang lebih besar jika dibandingkan implementasi *histogram shifting* secara langsung pada video. Hal ini karena sebuah *frame* pada video akan mirip dengan *frame* lain pada video tersebut yang saling berdekatan. Sebagai contoh, *frame* pertama akan mirip dengan *frame* kedua, *frame* ketiga akan mirip dengan *frame* keempat, dst. Akibatnya, nilai *prediction error* yang dihasilkan akan memiliki sebaran data dengan variasi yang kecil. Apalagi *prediction error* yang digunakan merupakan nilai absolut dari selisih piksel yang sebenarnya.

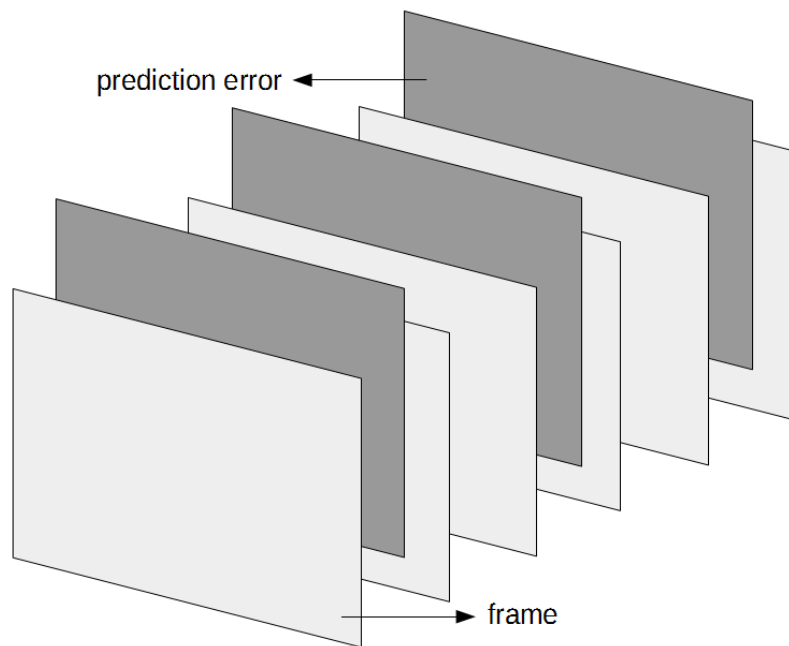
### 2.4.1 Proses Penyisipan

Pertama, *prediction error* dibangkitkan berdasarkan nilai piksel pada *frame*. *Prediction error* adalah nilai absolut dari selisih antar piksel pada *frame* yang saling berdekatan tetapi tidak saling *overlap*. Pada beberapa proses *video encoding*, *prediction error* akan dibangkitkan saat video dibuat dan disertakan pada video sehingga bisa langsung dikumpulkan untuk diproses. Tetapi, jika proses *encoding*

tidak melibatkan pembangkitan *prediction error*, maka pembangkitan *prediction error* dapat dilakukan menggunakan Persamaan 2.7 di bawah ini.

$$PE_{x,y,z} = |\alpha_{x,y,2z} - \alpha_{x,y,2z+1}| \quad (2.7)$$

Di mana PE adalah *prediction error*,  $\alpha$  adalah nilai piksel pada frame dan  $x$ - $y$ - $z$  adalah koordinat piksel pada *frame*. Ilustrasi *prediction error* dapat dilihat pada Gambar 2.6.



**Gambar 2.6.** Ilustrasi proses pembangkitan *prediction error*.

Penggunaan Persamaan 2.5 untuk membangkitkan *prediction error* akan menimbulkan masalah *underflow / overflow* saat diimplementasikan pada *frame*. Oleh karena itu, perlu dilakukan suatu prosedur pencegahan sebelum membangkitkan *prediction error*. Caranya adalah dengan melakukan penggeseran nilai piksel pada *frame* video digital ke dua buah *zero point* seperti yang diusulkan oleh (Tsai, et al., 2009). Proses penggeseran nilai piksel diawali dengan membangkitkan *histogram* berdasarkan nilai piksel pada seluruh *frame* video digital, memilih dua buah *zero point*, kemudian menggeser nilai piksel menggunakan Persamaan 2.8 di bawah ini.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} + 1, \text{ untuk } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} - 1, \text{ untuk } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (2.8)$$

Di mana  $\theta_1$  adalah *zero point* pertama,  $\theta_2$  adalah *zero point* kedua, dan  $0 < \theta_1 < \theta_2 < 255$ . Setelah itu barulah *prediction error* dibangkitkan.

Kedua, seluruh nilai *prediction error* dipindai secara vertikal, horizontal, ataupun berdasarkan algoritma tertentu. Frekuensi nilai *prediction error* pada proses pemindaian digunakan sebagai referensi untuk membangkitkan *histogram*. Sumbu-x pada *histogram* mewakili seluruh kemungkinan nilai *prediction error* (0 sampai 255), sedangkan sumbu-y mewakili frekuensi nilai-nilai *prediction error*.

Ketiga, nilai piksel yang berada di sebelah kanan *peak point* digeser ke kanan sebanyak satu poin. Tujuan dari penggeseran ini adalah untuk mengosongkan nilai piksel di sebelah kanan *peak point* agar dapat digunakan untuk menyisipkan payload. Proses penggeseran dapat digambarkan menggunakan Persamaan 2.9 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} + 1, \text{ untuk } PE_{x,y,z} > p \quad (2.9)$$

Perbedaan metode *neighbouring similarity* dengan metode *histogram shifting* adalah tidak digunakannya *zero point* sebagai tujuan penggeseran. Metode *neighbouring similarity* mengasumsikan *zero point* selalu berada di paling kanan *histogram* (nilai 255). Ini dikarenakan data-data *prediction error* akan terdistribusi secara tidak normal dan berada pada nilai-nilai yang kecil (sebelah kiri *histogram*). Selain itu, jika *prediction error* dibangkitkan sendiri, proses penggeseran nilai-nilai pada *frame* menggunakan Persamaan 2.6 akan menjamin frekuensi untuk nilai 255 akan kosong.

Keempat, *payload* disisipkan pada nilai *prediction error* yang nilainya sama dengan *peak point*. Proses penyisipan pesan rahasia dilakukan menggunakan Persamaan 2.10 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} + \beta_i, \text{ untuk } PE_{x,y,z} = p, \beta = \delta + m \quad (2.10)$$

Di mana  $\beta_i$  adalah *payload* ke- $i$ ,  $\delta$  adalah informasi *overhead* dan  $m$  adalah *secret message*. Jika *prediction error* dibangkitkan sendiri, maka perubahan nilai

*prediction error* harus diimplementasikan pada *frame* video digital. Proses implementasi dapat dilakukan menggunakan Persamaan 2.11 di bawah ini.

$$\begin{aligned} \alpha_{x,y,2z} &= \alpha_{x,y,2z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z} &= \alpha_{x,y,2z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (2.11)$$

Di mana  $\gamma_{x,y,z}$  adalah nilai *prediction error* ke- $(x,y,z)$  sebelum disisipi *payload*.

#### 2.4.2 Proses Ekstraksi dan Recovery

Pertama, nilai *peak point* dari proses penyisipan harus diketahui terlebih dahulu. Kemudian, pindai seluruh nilai *prediction error*. Jika *prediction error* pada proses penyisipan dibangkitkan sendiri, maka bangkitkan *prediction error* terlebih dahulu seperti pada proses penyisipan, yaitu dengan menggunakan Persamaan 2.7. Ekstrak pesan rahasia dari *prediction error* menggunakan Persamaan 2.12 pada saat proses pemindaian.

$$\begin{aligned} \beta_i &= 0, \text{ jika } PE_{x,y,z} = p \\ \beta_i &= 1, \text{ jika } PE_{x,y,z} = p + 1 \end{aligned} \quad (2.12)$$

Kedua, proses *recovery* dilakukan dengan menggeser *prediction error* yang bernilai lebih dari *peak point* ke arah *peak point*. Proses *recovery* tersebut digambarkan pada Persamaan 2.13 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} - 1, \text{ untuk } PE_{x,y,z} > p \quad (2.13)$$

Jika proses pembangkitan *prediction error* dilakukan sendiri menggunakan Persamaan 2.5, maka implementasikan perubahan *prediction error* pada *frame*. Proses implementasi dapat dilakukan menggunakan Persamaan 2.14 di bawah ini.

$$\begin{aligned} \alpha_{x,y,2z} &= \alpha_{x,y,2z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z} &= \alpha_{x,y,2z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (2.14)$$

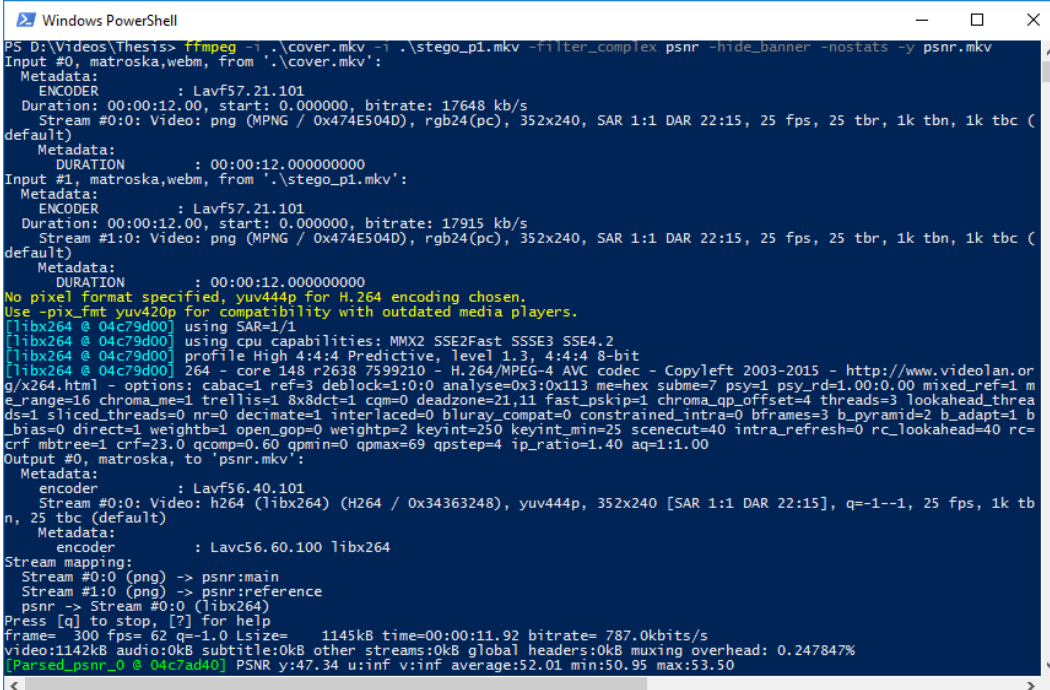
Kemudian kembalikan kondisi *frame* seperti sebelum dilakukan penggeseran (sebagai pencegahan masalah *underflow* / *overflow*) menggunakan Persamaan 2.15 di bawah ini.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} - 1, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} + 1, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (2.15)$$

Sedangkan untuk  $\alpha_{x,y,z} = \theta_1$  atau  $\alpha_{x,y,z} = \theta_2$ , digunakan informasi *overhead* untuk *recovery* posisinya.

## 2.5 Peak to Signal Noise Ratio (PSNR)

*Peak to Signal Noise Ratio* (PSNR) adalah salah satu metode untuk mengukur distorsi suatu citra digital (Cheddad, et al., 2010). Karena video digital adalah kumpulan citra digital (*frame*), maka metode ini dapat diadaptasikan pada video digital. Salah satu *tool* yang menyediakan fasilitas untuk menghitung nilai PSNR suatu video digital adalah FFmpeg (Bellard, 2000). Contoh hasil keluaran nilai PSNR menggunakan FFmpeg dapat dilihat pada Gambar 2.7 di bawah ini.



```
PS D:\Videos\Thesis> ffmpeg -i .\cover.mkv -i .\stego_p1.mkv -filter_complex psnr -hide_banner -nostats -y psnr.mkv
Input #0, matroska,webm, from '.\cover.mkv':
  Metadata:
    ENCODER           : Lavf57.21.101
    Duration: 00:00:12.00, start: 0.000000, bitrate: 17648 kb/s
    Stream #0:0: Video: png (MPNG / 0x474E504D), rgb24(pc), 352x240, SAR 1:1 DAR 22:15, 25 fps, 25 tbr, 1k tbn, 1k tbc (
default)
  Metadata:
    DURATION          : 00:00:12.000000000
Input #1, matroska,webm, from '.\stego_p1.mkv':
  Metadata:
    ENCODER           : Lavf57.21.101
    Duration: 00:00:12.00, start: 0.000000, bitrate: 17915 kb/s
    Stream #1:0: Video: png (MPNG / 0x474E504D), rgb24(pc), 352x240, SAR 1:1 DAR 22:15, 25 fps, 25 tbr, 1k tbn, 1k tbc (
default)
  Metadata:
    DURATION          : 00:00:12.000000000
No pixel format specified, yuv444p for H.264 encoding chosen.
Use -pix_fmt yuv420p for compatibility with outdated media players.
[libx264 @ 04c79d00] using SAR=1/1
[libx264 @ 04c79d00] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2
[libx264 @ 04c79d00] profile High 4:4:4 Predictive, level 1.3, 4:4:4 8-bit
[libx264 @ 04c79d00] 264 - core 148 r2638 7599210 - H.264/MPEG-4 AVC codec - Copyleft 2003-2015 - http://www.videoan.or
g/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed_ref=1 m
e_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=4 threads=3 lookahead_threa
ds=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b
bias=0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0 rc_lookahead=40 rc=
crf mbtree=1 crf=23.0 qcomp=0.60 qpmin=0 qpmx=69 qpstep=4 ip_ratio=1.40 aq=1:1.00
Output #0, matroska, to 'psnr.mkv':
  Metadata:
    encoder           : Lavf56.40.101
    Stream #0:0: Video: h264 (libx264) (H264 / 0x34363248), yuv444p, 352x240 [SAR 1:1 DAR 22:15], q=-1--1, 25 fps, 1k t
b n, 25 tbc (default)
  Metadata:
    encoder           : Lavc56.60.100 libx264
Stream mapping:
  Stream #0:0 (png) -> psnr:main
  Stream #1:0 (png) -> psnr:reference
  psnr -> Stream #0:0 (libx264)
Press [q] to stop, [?] for help
frame= 300 fps= 62 q=1.0 lsize= 1145kB time=00:00:11.92 bitrate= 787.0kbits/s
video:1142kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.247847%
[Parsed_psnr_0 @ 04c7ad40] PSNR y:47.34 u:inf v:inf average:52.01 min:50.95 max:53.50
```

Gambar 2.7. Contoh keluaran PSNR menggunakan FFmpeg.

Pada dasarnya, PSNR menggunakan formula *Mean Square Error* (MSE) untuk membandingkan nilai piksel pada *cover data* dengan *stego data*. Proses perhitungan MSE dapat dilihat pada Persamaan 2.16 di bawah ini.



$$MSE = \frac{1}{wh} \sum_{x=1}^w \sum_{y=1}^h (s_{xy} - c_{xy})^2 \quad (2.16)$$

Di mana  $w$  dan  $h$  adalah dimensi *frame*,  $x$  dan  $y$  adalah koordinat nilai piksel,  $s$  adalah *stego data*, dan  $c$  adalah *cover data*. Kemudian, persamaan MSE tersebut dioperasikan terhadap fungsi logaritma sedemikian rupa sehingga hasilnya dapat disajikan dalam bentuk decibels (dB). Adapun proses perhitungan PSNR dapat dilihat pada Persamaan 2.17 di bawah ini.

$$PSNR = 10 \log_{10} \left( \frac{C_{max}^2}{MSE} \right) \quad (2.17)$$

Di mana  $C_{max}$  adalah nilai maksimum dari piksel pada citra digital.

Karena disajikan dalam satuan dB, maka sebuah *stego data* dapat diklasifikasikan sebagai *high quality* jika nilainya berada di atas 40 dB. Sedangkan jika berada di bawah 30 dB, maka *stego data* diklasifikasikan sebagai *low quality*. Semakin besar nilai PSNR yang dihasilkan, maka semakin baik suatu *stego data* karena mengindikasikan bahwa *stego data* semakin mirip dengan *cover data*.

*[Halaman ini sengaja dikosongkan]*

## BAB 3

### METODE PENELITIAN

Pada bab ini dijelaskan metode penelitian yang diusulkan. Metode penelitian dibagi menjadi enam bagian, yaitu studi literatur, perancangan algoritma, implementasi penelitian, uji coba, analisis hasil dan penyusunan laporan akhir. Jadwal pengerjaan penelitian juga akan dijabarkan pada penelitian ini.

#### 3.1 Tahapan Penelitian

Penelitian ini memiliki beberapa tahapan penelitian. Tahapan-tahapan tersebut dapat dilihat pada Gambar 3.1. Secara umum, luaran yang diharapkan dari penelitian ini adalah metode *reversible steganography* baru untuk cover data berupa video digital yang merupakan pengembangan dari metode *neighbouring similarity*. Penjelasan dari tiap tahapan yang dilakukan pada penelitian ini akan dijabarkan pada subbab-subbab berikutnya.



**Gambar 3.1. Tahapan Penelitian.**

### 3.2 Studi Literatur

Tahapan ini merupakan tahapan pengumpulan referensi pendukung yang sesuai dan dapat dipertanggungjawabkan secara ilmiah. Referensi yang digunakan dapat berupa jurnal, artikel, buku, atau sumber lainnya. Referensi-referensi yang ditemukan akan digunakan sebagai dasar pemikiran perancangan algoritma untuk memecahkan masalah yang sedang diteliti. Topik-topik referensi yang dikumpulkan diantaranya adalah metode *histogram shifting* beserta pengembangannya, survey pada topik *steganography*, dan metode-metode *steganography* pendukung.

Topik *histogram shifting* beserta pengembangannya digunakan sebagai dasar pemikiran dalam mencari sekaligus memecahkan masalah. Survey pada topik *steganography* digunakan untuk mengumpulkan informasi mengenai perkembangan metode-metode *steganography* yang ada saat ini, sekaligus untuk mencari tahu penelitian-penelitian serupa yang telah ada. Metode *steganography* pendukung digunakan sebagai metode bantuan dalam menyelesaikan masalah pada metode utama yang digunakan.

### 3.3 Perancangan Algoritma

Kapasitas *secret message* yang dihasilkan oleh metode *neighbouring similarity* bergantung pada nilai *peak point*. Semakin tinggi nilai *peak point*, maka kapasitas yang dihasilkan pun memiliki kemungkinan untuk semakin tinggi. Nilai *peak point* itu sendiri didapatkan dari *histogram* yang dibangkitkan berdasarkan nilai *prediction error*. Artinya, pembangkitan *prediction error* menjadi salah satu faktor penting untuk menentukan kapasitas *secret message* pada metode *neighbouring similarity*.

Ide utama dari penelitian ini adalah memodifikasi proses pembangkitan *prediction error* pada metode *neighbouring similarity*. Tiga metode pembangkitan *prediction error* diusulkan pada penelitian ini untuk mencoba meningkatkan kapasitas *secret message* yang dihasilkan. Pertama, *prediction error* dibangkitkan

di tiap tiga *frame*. Kedua, *prediction error* dibangkitkan berdasarkan *prediction error* yang telah ada. Ketiga, pembangkitan *prediction error* yang awalnya *non-overlapped* diubah menjadi *overlapped*. Detail dari ketiga usulan tersebut akan dijelaskan lebih lanjut pada subbab-subbab berikutnya.

Perubahan proses pembangkitan *prediction error* diharapkan mampu menghasilkan *histogram* yang lebih baik dibandingkan dengan metode *neighbouring similarity*. Pernyataan “lebih baik” yang dimaksud di sini adalah menghasilkan nilai *peak point* yang lebih tinggi dibandingkan *histogram* pada metode *neighbouring similarity*. Ketiga metode yang diusulkan diperkirakan mampu memperkecil sebaran nilai *prediction error*, sehingga nilai *peak point* dimungkinkan menjadi lebih tinggi.

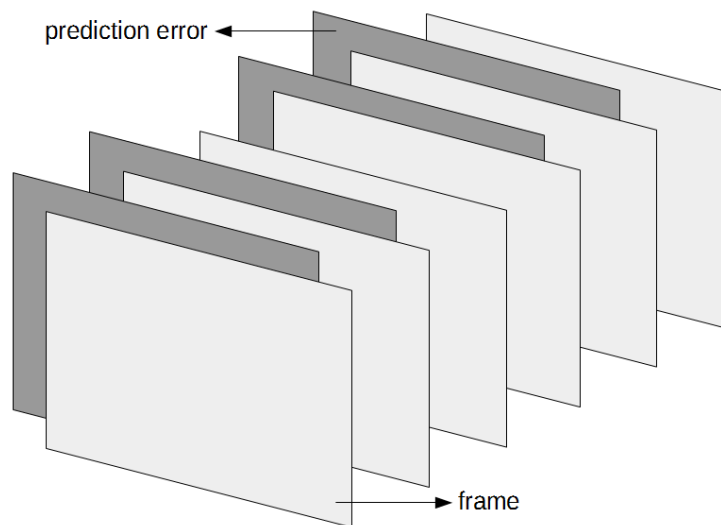
Selain dalam hal kapasitas, metode-metode usulan pada penelitian ini juga diharapkan mampu mempertahankan tingkat kemiripan *stego data* terhadap *cover data*-nya. Walaupun peningkatan kapasitas seringkali berbanding terbalik dengan tingkat kemiripan, tetapi setidaknya metode-metode yang diusulkan diharapkan mampu mempertahankan nilai PSNR di atas 40 dB. Lebih baik lagi, jika metode-metode usulan mampu menghasilkan nilai PSNR yang perbedaannya hanya sedikit dibandingkan metode yang telah ada sebelumnya, yaitu metode *neighbouring similarity*.

### 3.3.1 Neighbouring Similarity of Triplets

Metode *neighbouring similarity* menggunakan tiap dua buah *frame* yang bersifat *non-overlapped* untuk menghasilkan sebuah *prediction error*. *Frame* pertama dan kedua akan menghasilkan *prediction error* pertama, *frame* ketiga dan keempat akan menghasilkan *prediction error* kedua, dan seterusnya hingga akhir *frame*. Artinya, jika terdapat  $n$ -buah *frame* pada sebuah video, maka jumlah *prediction error* yang dihasilkan adalah  $\lfloor \frac{n}{2} \rfloor$ .

Ide dari *neighbouring similarity of triplets* ini adalah menggunakan tiga buah *frame* untuk menghasilkan dua buah *prediction error*. Dengan begitu, jumlah *prediction error* yang dihasilkan pada sebuah video digital akan lebih banyak

dibandingkan dengan metode *neighbouring similarity*. Jika terdapat  $n$ -buah *frame* pada sebuah video digital, maka jumlah *prediction error* yang dihasilkan adalah  $2 \lfloor \frac{n}{3} \rfloor$ . Peningkatan jumlah *prediction error* ini diharapkan mampu meningkatkan nilai *peak point*. Ilustrasi dari usulan proses pembangkitan *prediction error* pada metode usulan ini dapat dilihat pada Gambar 3.1.



**Gambar 3.2. Ilustrasi pembangkitan histogram pada metode usulan pertama.**

Alasan digunakannya tiga buah *frame* adalah untuk menjaga sifat mirip (*similarity*) antar *frame* yang saling berdekatan. *Frame* pertama mirip dengan *frame* kedua, tetapi *frame* kedua juga mirip dengan *frame* ketiga. Metode ini memanfaatkan *frame* tengah sebagai *reference macro block* dari *frame* sebelum dan sesudahnya. Dengan begitu, selain *prediction error* yang dihasilkan lebih banyak, *similarity* yang merupakan keuntungan metode *neighbouring similarity* juga tetap dapat dipertahankan.

### 3.3.1.1 Proses Penyisipan

Pertama, pencegahan terhadap masalah *underflow* / *overflow* dilakukan. Caranya adalah dengan membangkitkan *histogram* berdasarkan seluruh *frame* pada

video digital, kemudian pilih dua buah *zero point* (nilai dengan frekuensi terendah) di sebelah paling kiri dan paling kanan. Nilai-nilai di sebelah kiri dan kanan *zero point* tersebut kemudian digeser ke arah *zero point* sebanyak satu poin sehingga nilai 0 dan 255 akan menjadi kosong. Proses penggeseran ini digambarkan menggunakan Persamaan 3.1.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} + 1, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} - 1, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (3.1)$$

Di mana  $\alpha_{x,y,z}$  adalah nilai piksel ke- $x,y,z$ ,  $\theta_1$  adalah *zero point* pertama,  $\theta_2$  adalah *zero point* kedua dan  $\theta_1 < \theta_2$ . Jika *zero point* bernilai lebih dari 0, baik  $\theta_1$  ataupun  $\theta_2$ , maka perlu dibuat suatu informasi *overhead* untuk menandai lokasi piksel di *zero point* dan di sebelah *zero point*.

Kedua, *prediction error* dibangkitkan berdasarkan nilai-nilai piksel pada seluruh *frame*. Jika *PE* adalah *prediction error*,  $\alpha$  adalah nilai piksel,  $x-y-z$  adalah koordinat, maka proses pembangkitan *prediction error* pada metode ini dilakukan menggunakan Persamaan 3.2.

$$PE_{x,y,z} = \left| \alpha_{x,y,z+\lfloor \frac{z}{2} \rfloor} - \alpha_{x,y,z+\lfloor \frac{z}{2} \rfloor+1} \right| \quad (3.2)$$

Setelah seluruh *prediction error* selesai dibangkitkan (sebanyak  $2 \lfloor \frac{n}{3} \rfloor$ ), di mana  $n$  adalah jumlah seluruh *frame* pada video digital, maka *histogram* dibangkitkan berdasarkan seluruh *prediction error* yang telah terbentuk. *Peak point* (nilai dengan frekuensi tertinggi) dipilih dari *histogram* tersebut.

Ketiga, *histogram* digeser untuk mengosongkan frekuensi nilai piksel di sebelah *peak point*. Jika  $p$  adalah *peak point*, maka proses penggeseran dilakukan menggunakan Persamaan 3.3 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} + 1, \text{ untuk } PE_{x,y,z} > p \quad (3.3)$$

Proses ini akan menggeser nilai-nilai dengan frekuensi yang lebih tinggi dari *peak point*. Ruang kosong di sebelah *peak point* ini nantinya akan digunakan sebagai ruang untuk menggeser *peak point* pada proses penyisipan *payload*.

Keempat, *payload* disisipkan pada *peak point*. *Payload* adalah kombinasi dari informasi *overhead* dan *secret message*. Proses penyisipan ini dilakukan menggunakan Persamaan 3.4.

$$PE_{x,y,z} = PE_{x,y,z} + \beta_i, \text{ dimana } \beta = \delta + m, PE_{x,y,z} = p \quad (3.4)$$

Di mana  $\beta_i$  adalah *payload* ke- $i$ ,  $\delta$  adalah informasi *overhead* dan  $m$  adalah *secret message*. Informasi *overhead* didapatkan dari proses pertama, yaitu informasi berupa penanda piksel yang bernilai *zero point* dan yang bernilai di sebelah *zero point*. Informasi *overhead* disertakan agar setelah proses ekstraksi, *frame* pada video digital dapat di-*recovery*.

Kelima, perubahan *prediction error* diimplementasikan pada *frame* video digital. Proses ini dilakukan dua kali, yaitu terhadap urutan *prediction error* genap dan terhadap urutan *prediction error* ganjil. Untuk urutan *prediction error* genap, proses implementasi dilakukan menggunakan Persamaan 3.5.

$$\begin{aligned} \alpha_{x,y,2z} &= \alpha_{x,y,2z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z} &= \alpha_{x,y,2z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (3.5)$$

Sedangkan untuk *prediction error* ganjil, proses implementasi dilakukan menggunakan Persamaan 3.6 di bawah ini.

$$\begin{aligned} \alpha_{x,y,2z+1} &= \alpha_{x,y,2z+1} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z+1} &= \alpha_{x,y,2z+1} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (3.6)$$

Di mana  $\gamma_{x,y,z}$  adalah nilai *prediction error* ke- $x,y,z$  sebelum disisipi *payload*.

### 3.3.1.2 Proses Ekstraksi dan Recovery

Pertama, *peak point* harus diketahui terlebih dahulu. Kemudian, *prediction error* dibangkitkan berdasarkan nilai-nilai piksel pada *frame* seperti pada proses penyisipan. Karena sama dengan proses penyisipan, proses pembangkitan *prediction error* menggunakan Persamaan 3.2.

Kedua, *payload* diekstraksi dari seluruh *prediction error*. Caranya adalah dengan melakukan pemindaian terhadap seluruh nilai *prediction error*. Jika ditemukan nilai yang sama dengan *peak point*, maka *payload* ke- $i$  adalah 0. Sedangkan jika ditemukan nilai yang sama dengan *peak point* ditambah satu, maka



*payload* ke-*i* adalah 1. Proses ekstraksi *payload* dari *prediction error* dapat digambarkan menggunakan Persamaan 3.7 di bawah ini.

$$\begin{aligned} \beta_i &= 0, \text{ jika } PE_{x,y} = p \\ \beta_i &= 1, \text{ jika } PE_{x,y} = p + 1 \end{aligned} \quad (3.7)$$

Ketiga, seluruh *prediction error* di-*recovery* seperti sebelum disisipi *payload*. Proses *recovery* dilakukan dengan menggeser nilai *prediction error* di sebelah kanan *peak point* ke sebelah kiri sebanyak satu poin. Dengan begitu, bentuk *histogram* dari *prediction error* akan kembali seperti sebelum disisipi *payload*. Proses *recovery* tiap *prediction error* dapat digambarkan menggunakan Persamaan 3.8 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} - 1, \text{ untuk } PE_{x,y,z} > p \quad (3.8)$$

Keempat, perubahan pada *prediction error* diimplementasikan pada *frame*. Proses implementasi ini dilakukan dua kali, yaitu terhadap urutan *prediction error* genap dan urutan *prediction error* ganjil. Proses implementasi terhadap urutan genap dilakukan menggunakan Persamaan 3.9.

$$\begin{aligned} \alpha_{x,y,2z} &= \alpha_{x,y,2z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z} &= \alpha_{x,y,2z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (3.9)$$

Sedangkan implementasi terhadap urutan *prediction error* ganjil dilakukan menggunakan Persamaan 3.10 di bawah ini.

$$\begin{aligned} \alpha_{x,y,2z+1} &= \alpha_{x,y,2z+1} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z+1} &= \alpha_{x,y,2z+1} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (3.10)$$

Kelima, setelah seluruh *prediction error* selesai diproses, seluruh *frame* di-*recovery* seperti sebelum dilakukan penggeseran untuk menghindari kondisi *underflow* / *overflow*. Proses *recovery frame* dilakukan menggunakan menggunakan Persamaan 3.11 di bawah ini.

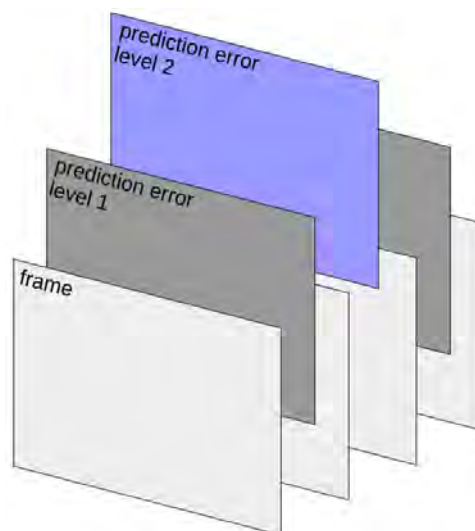
$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} - 1, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} + 1, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (3.11)$$

Sedangkan untuk  $\alpha_{x,y,z} = \theta_1$  atau  $\alpha_{x,y,z} = \theta_2$ , proses *recovery* dilakukan berdasarkan informasi *overhead* yang didapatkan dari *payload*.

### 3.3.2 Multi-Level Neighbouring Similarity

*Prediction error* mampu menghasilkan bentuk *histogram* yang lebih baik karena *prediction error* memanfaatkan *similarity* yang dimiliki dua buah *frame* yang saling berdekatan. *Similarity* membuat sebaran nilai yang ada pada *prediction error* memiliki variasi yang lebih sedikit dibandingkan dengan sebaran nilai pada *frame* itu sendiri. Sebaran nilai yang kecil tersebut diharapkan mampu menghasilkan nilai *peak point* yang lebih tinggi dibandingkan dengan penggunaan nilai *frame* secara langsung.

Ide dari *multi-level neighbouring similarity* adalah menggunakan *prediction error* yang dihasilkan oleh metode *neighbouring similarity* untuk membangkitkan *prediction error* yang lain. Jika dua buah *frame* yang saling berdekatan menghasilkan matriks dengan sebaran data yang kecil (yaitu *prediction error*), maka dua buah *prediction error* yang saling berdekatan kemungkinan mampu menghasilkan matriks dengan sebaran data yang lebih kecil lagi. Sebaran data yang lebih kecil memungkinkan terbentuknya nilai *peak point* yang lebih tinggi. Ilustrasi dari usulan proses pembangkitan *prediction error* pada metode usulan ini dapat dilihat pada Gambar 3.2 di bawah ini.



**Gambar 3.3. Ilustrasi pembangkitan prediction error pada metode usulan kedua.**

Metode usulan ini akan menimbulkan sebuah masalah baru, yaitu kemungkinan terjadinya *underflow* / *overflow* pada *prediction error* tingkat satu. Padahal, keberadaan *prediction error* tingkat satu itu sendiri sudah memungkinkan terjadinya *underflow* / *overflow* saat implementasi terhadap *frame*. Jika  $\gamma$  adalah *prediction error* tingkat dua,  $\beta$  adalah *prediction error* tingkat satu, dan  $x$ - $y$ - $z$  adalah koordinat, maka proses pembangkitan *prediction error* tingkat dua dapat diwakilkan oleh Persamaan 3.12 di bawah ini.

$$\gamma_{x,y,z} = \beta_{x,y,2z} - \beta_{x,y,2z+1}, \text{ dimana } \gamma = \{0 - 255\}, \beta = \{0 - 255\} \quad (3.12)$$

Karena kemungkinan nilai  $\beta$  adalah 0 hingga 255, maka perubahan pada  $\gamma$  akan mengakibatkan kemungkinan salah satu nilai  $\beta$  menjadi -1 hingga 256. Di sisi lain, nilai  $\beta$  didapatkan dari Persamaan 3.13.

$$\beta_{x,y,z} = \alpha_{x,y,2z} - \alpha_{x,y,2z+1}, \text{ dimana } \alpha = \{0 - 255\}, \beta = \{0 - 255\} \quad (3.13)$$

Di mana  $\alpha$  adalah nilai piksel pada *frame*. Jika kemungkinan nilai  $\beta$  berubah menjadi -1 hingga 256, maka kemungkinan salah satu nilai  $\alpha$  akan menjadi -2 hingga 257. Untuk mengatasi masalah ini, penelitian ini mengusulkan untuk melakukan penggeseran sebanyak dua nilai sebelum melakukan penyisipan pesan.

### 3.3.2.1 Proses Penyisipan

Pertama, pencegahan terhadap masalah *underflow* / *overflow* pada tingkat *frame* dilakukan. Caranya adalah dengan membangkitkan *histogram* berdasarkan seluruh *frame* pada video digital, kemudian pilih dua buah *zero point* (nilai dengan frekuensi terendah) di sebelah paling kiri dan paling kanan. Nilai-nilai di sebelah kiri dan kanan *zero point* tersebut kemudian digeser ke arah *zero point* sebanyak dua poin sehingga nilai 0, 1, 254 dan 255 akan menjadi kosong. Proses penggeseran ini digambarkan menggunakan Persamaan 3.14.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} + 2, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} - 2, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (3.14)$$

Di mana  $\alpha_{x,y,z}$  adalah nilai piksel ke- $x,y,z$ ,  $\theta_1$  adalah *zero point* pertama,  $\theta_2$  adalah *zero point* kedua dan  $\theta_1 < \theta_2$ . Jika *zero point* bernilai lebih dari 0, baik  $\theta_1$  ataupun

$\theta_2$ , maka perlu dibuat suatu informasi *overhead* untuk menandai lokasi piksel di *zero point* dan di sebelah *zero point*.

Kedua, *prediction error level 1* dibangkitkan berdasarkan nilai-nilai piksel pada seluruh *frame*. Jika  $PE$  adalah *prediction error level 1*,  $\alpha$  adalah nilai piksel,  $x$ - $y$ - $z$  adalah koordinat, maka proses pembangkitan *prediction error* pada metode ini dilakukan menggunakan Persamaan 3.15.

$$PE_{x,y,z} = |\alpha_{x,y,2z} - \alpha_{x,y,2z+1}| \quad (3.15)$$

Setelah seluruh *prediction error level 1* selesai dibangkitkan (sebanyak  $\lfloor \frac{n}{2} \rfloor$ ), di mana  $n$  adalah jumlah seluruh *frame* pada video digital, maka dilakukan pencegahan masalah *underflow / overflow* pada tingkat *prediction error level 1*. Caranya adalah dengan menggeser nilai *prediction error level 1* ke kanan sebanyak 1 poin. Proses penggeseran tersebut digambarkan menggunakan Persamaan 3.16 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} + 1 \quad (3.16)$$

Ketiga, *prediction error level 2* dibangkitkan berdasarkan *prediction error level 1*. Jika  $PE'$  adalah *prediction error level 2*, maka proses pembangkitan *prediction error level 2* dilakukan menggunakan Persamaan 3.17.

$$PE'_{x,y,z} = |PE_{x,y,2z} - PE_{x,y,2z+1}| \quad (3.17)$$

Setelah seluruh *prediction error* selesai dibangkitkan (sebanyak  $\lfloor \frac{n}{4} \rfloor$ ), maka *histogram* dibangkitkan berdasarkan seluruh *prediction error level 2* yang telah terbentuk. *Peak point* (nilai dengan frekuensi tertinggi) dipilih dari *histogram* tersebut.

Keempat, *histogram* digeser untuk mengosongkan frekuensi nilai piksel di sebelah *peak point*. Jika  $p$  adalah *peak point*, maka proses penggeseran dilakukan menggunakan Persamaan 3.18 di bawah ini.

$$PE'_{x,y,z} = PE'_{x,y,z} + 1, \text{ untuk } PE'_{x,y,z} > p \quad (3.18)$$

Proses ini akan menggeser nilai-nilai dengan frekuensi yang lebih tinggi dari *peak point*. Ruang kosong di sebelah *peak point* ini nantinya akan digunakan sebagai ruang untuk menggeser *peak point* pada proses penyisipan *payload*.

Kelima, *payload* disisipkan pada *peak point*. *Payload* adalah kombinasi dari informasi *overhead* dan *secret message*. Proses penyisipan ini dilakukan menggunakan Persamaan 3.19.

$$PE'_{x,y,z} = PE'_{x,y,z} + \beta_i, \text{ dimana } \beta = \delta + m, PE'_{x,y,z} = p \quad (3.19)$$

Di mana  $\beta_i$  adalah *payload* ke- $i$ ,  $\delta$  adalah informasi *overhead* dan  $m$  adalah *secret message*. Informasi *overhead* didapatkan dari proses pertama, yaitu informasi berupa penanda piksel yang bernilai *zero point* dan yang bernilai di sebelah *zero point*. Informasi *overhead* disertakan agar setelah proses ekstraksi, *frame* pada video digital dapat di-*recovery*.

Keenam, perubahan *prediction error* diimplementasikan pada *prediction error level 1*, baru kemudian pada *frame* video digital. Proses ini dilakukan karena *prediction error* yang digunakan untuk menyisipkan pesan pada metode ini tidak dibangkitkan langsung dari *frame* video digital. Oleh karena itu, perlu dilakukan perubahan secara bertahap. Implementasi yang pertama dilakukan menggunakan Persamaan 3.20 di bawah ini.

$$\begin{aligned} PE_{x,y,2z} &= PE_{x,y,2z} + (PE'_{x,y,z} - \gamma'_{x,y,z}), \text{ jika } PE_{x,y,2z} > PE_{x,y,2z+1} \\ PE_{x,y,2z} &= PE_{x,y,2z} - (PE'_{x,y,z} - \gamma'_{x,y,z}), \text{ jika } PE_{x,y,2z} < PE_{x,y,2z+1} \end{aligned} \quad (3.20)$$

Di mana  $\gamma_{x,y,z}$  adalah *prediction error level 2* yang ke- $x,y,z$  sebelum disisipi *payload*. Setelah implementasi yang pertama selesai dilakukan pada seluruh *prediction error level 1*, barulah perubahan diimplementasikan pada *frame* video digital. Proses implementasi pada *frame* dilakukan menggunakan Persamaan 3.21 di bawah ini.

$$\begin{aligned} \alpha_{x,y,2z} &= \alpha_{x,y,2z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z} &= \alpha_{x,y,2z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (3.21)$$

### 3.3.2.2 Proses Ekstraksi dan Recovery

Pertama, nilai peak point harus diketahui terlebih dahulu. Kemudian, *prediction error level 1* dan *level 2* dibangkitkan seperti pada proses penyisipan. Proses pembangkitan *prediction error level 1* dilakukan menggunakan Persamaan 3.15, sedangkan proses pembangkitan *prediction error level 2* dilakukan menggunakan Persamaan 3.17.

Kedua, *payload* diekstraksi dari *prediction error level 2*. Caranya adalah dengan melakukan pemindaian terhadap seluruh nilai *prediction error level 2*. Jika ditemukan nilai yang sama dengan *peak point*, maka *payload* ke-*i* adalah 0. Sedangkan jika ditemukan nilai yang sama dengan *peak point* ditambah satu, maka *payload* ke-*i* adalah 1. Proses ekstraksi *payload* dari *prediction error level 2* dapat digambarkan menggunakan Persamaan 3.22 di bawah ini.

$$\begin{aligned} \beta_i &= 0, \text{ jika } PE_{x,y} = p \\ \beta_i &= 1, \text{ jika } PE_{x,y} = p + 1 \end{aligned} \quad (3.22)$$

Ketiga, *prediction error level 2* di-recovery seperti sebelum disisipi *payload*. Proses *recovery* dilakukan dengan menggeser nilai *prediction error level 2* di sebelah kanan *peak point* ke sebelah kiri sebanyak satu poin. Dengan begitu, bentuk *histogram* dari *prediction error level 2* akan kembali seperti sebelum disisipi *payload*. Proses *recovery* dari *prediction error level 2* dapat digambarkan menggunakan Persamaan 3.23.

$$PE'_{x,y,z} = PE'_{x,y,z} - 1, \text{ untuk } PE'_{x,y,z} > p \quad (3.23)$$

Setelah seluruh *prediction error level 2* di-recovery, perubahan tersebut diimplementasikan pada *prediction error level 1*. Proses implementasi dilakukan menggunakan Persamaan 3.24 di bawah ini.

$$\begin{aligned} PE_{x,y,2z} &= PE_{x,y,2z} - (PE'_{x,y,z} - \gamma'_{x,y,z}), \text{ jika } PE_{x,y,2z} > PE_{x,y,2z+1} \\ PE_{x,y,2z} &= PE_{x,y,2z} + (PE'_{x,y,z} - \gamma'_{x,y,z}), \text{ jika } PE_{x,y,2z} < PE_{x,y,2z+1} \end{aligned} \quad (3.24)$$

Keempat, *prediction error level 1* di-recovery seperti sebelum dilakukan penggeseran untuk menghindari kondisi *underflow* / *overflow*. Proses *recovery* dilakukan dengan menggeser seluruh *prediction error level 1* sebanyak satu poin

ke sebelah kiri. Proses penggeseran ini dapat digambarkan menggunakan Persamaan 3.25.

$$PE_{x,y,z} = PE_{x,y,z} - 1 \quad (3.25)$$

Dari proses penggeseran ini, *prediction error level* 1 akan kembali ke bentuk awal dibangkitkannya. Kemudian, perubahan ini diimplementasikan pada *frame* video digital. Proses implementasi dilakukan menggunakan Persamaan 3.26 di bawah ini.

$$\begin{aligned} \alpha_{x,y,2z} &= \alpha_{x,y,2z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} > \alpha_{x,y,2z+1} \\ \alpha_{x,y,2z} &= \alpha_{x,y,2z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,2z} < \alpha_{x,y,2z+1} \end{aligned} \quad (3.26)$$

Kelima, setelah seluruh *prediction error* selesai diproses, seluruh *frame* di-*recovery* seperti sebelum dilakukan penggeseran untuk menghindari kondisi *underflow* / *overflow*. Proses *recovery frame* dilakukan menggunakan menggunakan Persamaan 3.27 di bawah ini.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} - 2, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} + 2, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (3.27)$$

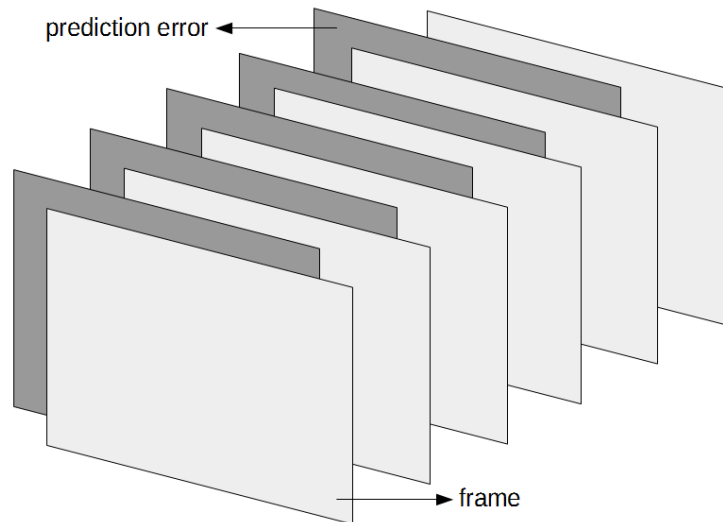
Sedangkan untuk  $\alpha_{x,y,z} = \theta_1$  atau  $\alpha_{x,y,z} = \theta_2$ , proses *recovery* dilakukan berdasarkan informasi *overhead* yang didapatkan dari *payload*.

### 3.3.3 Overlapped Neighbouring Similarity

Pada metode *neighbouring similarity*, proses pembangkitan *prediction error* dilakukan secara *non-overlapped*. *Frame* pertama akan berpasangan dengan *frame* kedua, *frame* ketiga akan berpasangan dengan *frame* keempat, dan seterusnya sehingga tidak ada *frame* yang digunakan dua kali. Hal ini dikarenakan salah satu *frame* harus digunakan sebagai acuan (*reference macro block*) untuk membangkitkan *prediction error*.

Ide dari *overlapped neighbouring similarity* ini adalah mengubah proses pembangkitan *prediction error* dari yang *non-overlapped* menjadi *overlapped*. *Frame* yang telah digunakan untuk membangkitkan sebuah *prediction error*, akan digunakan lagi untuk membangkitkan *prediction error* berikutnya. *Frame* pertama akan berpasangan dengan *frame* kedua, *frame* kedua dengan *frame* ketiga, dan seterusnya. Dengan begitu, jumlah *prediction error* yang didapatkan bisa

bertambah hampir dua kali lipat dari metode *neighbouring similarity*, dari yang awalnya  $\lfloor \frac{n}{2} \rfloor$  menjadi  $n - 1$ . Ilustrasi dari usulan proses pembangkitan *prediction error* pada metode ini dapat dilihat pada Gambar 3.3 di bawah ini.



**Gambar 3.4. Ilustrasi pembangkitan prediction error pada metode usulan ketiga.**

Penggunaan skema *overlapped* tentu akan menimbulkan masalah tambahan. Permasalahan tersebut adalah proses ekstraksi harus dilakukan satu per satu dari *frame* paling akhir ke *frame* paling awal. Setiap ekstraksi selesai dilakukan pada sebuah pasangan *frame*, *frame* yang berubah (*encoded macro block*) harus di-*recovery* terlebih dahulu agar kembali ke kondisi semula. Kemudian, *frame* yang telah di-*recovery* tersebut baru dapat digunakan untuk membangkitkan *prediction error* berikutnya. Jika tidak di-*recovery* ke kondisi semula, maka *prediction error* yang dihasilkan tidak akan sama dengan *prediction error* yang dihasilkan pada proses penyisipan.

### 3.3.3.1 Proses Penyisipan

Pertama, pencegahan terhadap masalah *underflow* / *overflow* dilakukan. Caranya adalah dengan membangkitkan *histogram* berdasarkan seluruh *frame* pada video digital, kemudian pilih dua buah *zero point* (nilai dengan frekuensi terendah)



di sebelah paling kiri dan paling kanan. Nilai-nilai di sebelah kiri dan kanan *zero point* tersebut kemudian digeser ke arah *zero point* sebanyak satu poin sehingga nilai 0 dan 255 akan menjadi kosong. Proses penggeseran ini digambarkan menggunakan Persamaan 3.28.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} + 1, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} - 1, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (3.28)$$

Di mana  $\alpha_{x,y,z}$  adalah nilai piksel ke- $x,y,z$ ,  $\theta_1$  adalah *zero point* pertama,  $\theta_2$  adalah *zero point* kedua dan  $\theta_1 < \theta_2$ . Jika *zero point* bernilai lebih dari 0, baik  $\theta_1$  ataupun  $\theta_2$ , maka perlu dibuat suatu informasi *overhead* untuk menandai lokasi piksel di *zero point* dan di sebelah *zero point*.

Kedua, *prediction error* dibangkitkan berdasarkan nilai-nilai piksel pada seluruh *frame*. Jika *PE* adalah *prediction error*,  $\alpha$  adalah nilai piksel,  $x-y-z$  adalah koordinat, maka proses pembangkitan *prediction error* pada metode ini dilakukan menggunakan Persamaan 3.29.

$$PE_{x,y,z} = |\alpha_{x,y,z} - \alpha_{x,y,z+1}| \quad (3.29)$$

Setelah seluruh *prediction error* selesai dibangkitkan (sebanyak  $n-1$ ), di mana  $n$  adalah jumlah seluruh *frame* pada video digital, maka *histogram* dibangkitkan berdasarkan seluruh *prediction error* yang telah terbentuk. *Peak point* (nilai dengan frekuensi tertinggi) dipilih dari *histogram* tersebut.

Ketiga, *histogram* digeser untuk mengosongkan frekuensi nilai piksel di sebelah *peak point*. Jika  $p$  adalah *peak point*, maka proses penggeseran dilakukan menggunakan Persamaan 3.30 di bawah ini.

$$PE_{x,y,z} = PE_{x,y,z} + 1, \text{ untuk } PE_{x,y,z} > p \quad (3.30)$$

Proses ini akan menggeser nilai-nilai dengan frekuensi yang lebih tinggi dari *peak point*. Ruang kosong di sebelah *peak point* ini nantinya akan digunakan sebagai ruang untuk menggeser *peak point* pada proses penyisipan *payload*.

Keempat, *payload* disisipkan pada *peak point*. *Payload* adalah kombinasi dari informasi *overhead* dan *secret message*. Proses penyisipan ini dilakukan menggunakan Persamaan 3.31.

$$PE_{x,y,z} = PE_{x,y,z} + \beta_i, \text{ dimana } \beta = \delta + m, PE_{x,y,z} = p \quad (3.31)$$

Di mana  $\beta_i$  adalah payload ke- $i$ ,  $\delta$  adalah informasi *overhead* dan  $m$  adalah *secret message*. Informasi *overhead* didapatkan dari proses pertama, yaitu informasi berupa penanda piksel yang bernilai *zero point* dan yang bernilai di sebelah *zero point*. Informasi *overhead* disertakan agar setelah proses ekstraksi, *frame* pada video digital dapat di-*recovery*.

Kelima, perubahan *prediction error* diimplementasikan pada *frame* video digital. Proses ini dilakukan karena *prediction error* pada metode ini merupakan matriks yang dibangkitkan, bukan bagian dari video digital. Oleh karena itu, perlu dilakukan perubahan pada komponen pembentuk video digital (yaitu *frame*). Jika  $\gamma_{x,y,z}$  adalah nilai *prediction error* ke- $x,y,z$  sebelum disisipi *payload*, maka proses implementasi dilakukan menggunakan Persamaan 3.32.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,z} > \alpha_{x,y,z+1} \\ \alpha_{x,y,z} &= \alpha_{x,y,z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,z} < \alpha_{x,y,z+1} \end{aligned} \quad (3.32)$$

### 3.3.3.2 Proses Ekstraksi dan *Recovery*

Pertama, *prediction error* dibangkitkan berdasarkan nilai-nilai piksel pada *frame*. Berbeda dengan proses penyisipan, pembangkitan *prediction error* pada saat ekstraksi dilakukan dari *frame* akhir ke *frame* awal. Selain itu, setiap *prediction error* akan diproses secara terpisah satu per satu, bukan keseluruhan secara langsung. Jadi, *prediction error* paling akhir diproses dulu hingga tahap keempat selesai, kemudian kembali ke tahap pertama lagi untuk *prediction error* berikutnya. Begitu seterusnya hingga seluruh *prediction error* selesai diproses dari akhir hingga awal, barulah tahap kelima dijalankan. Pembangkitan *prediction error* di tiap pasangan *frame* dilakukan menggunakan Persamaan 3.29.

Kedua, *payload* diekstraksi dari tiap *prediction error*. Caranya adalah dengan melakukan pemindaian terhadap seluruh nilai di tiap *prediction error*. Jika ditemukan nilai yang sama dengan *peak point*, maka *payload* ke- $i$  adalah 0. Sedangkan jika ditemukan nilai yang sama dengan *peak point* ditambah satu, maka

*payload* ke-*i* adalah 1. Proses ekstraksi *payload* dari tiap *prediction error* dapat digambarkan menggunakan Persamaan 3.30 di bawah ini.

$$\begin{aligned} \beta_i &= 0, \text{ jika } PE_{x,y} = p \\ \beta_i &= 1, \text{ jika } PE_{x,y} = p + 1 \end{aligned} \quad (3.30)$$

Ketiga, tiap *prediction error* di-*recovery* seperti sebelum disisipi *payload*. Proses *recovery* dilakukan dengan menggeser nilai *prediction error* di sebelah kanan *peak point* ke sebelah kiri sebanyak satu poin. Dengan begitu, bentuk *histogram* dari *prediction error* akan kembali seperti sebelum disisipi *payload*. Proses *recovery* tiap *prediction error* dapat digambarkan menggunakan Persamaan 3.31 di bawah ini.

$$PE_{x,y} = PE_{x,y} - 1, \text{ untuk } PE_{x,y} > p \quad (3.31)$$

Keempat, perubahan pada tiap *prediction error* diimplementasikan pada *frame*. Proses implementasi dilakukan menggunakan Persamaan 3.32.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} - (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,z} > \alpha_{x,y,z+1} \\ \alpha_{x,y,z} &= \alpha_{x,y,z} + (PE_{x,y,z} - \gamma_{x,y,z}), \text{ jika } \alpha_{x,y,z} < \alpha_{x,y,z+1} \end{aligned} \quad (3.32)$$

Dari proses implementasi ini, *frame* ke-*z* akan kembali seperti sebelum disisipi *payload*. Oleh karena itu, *frame* tersebut bisa digunakan sebagai acuan untuk membangkitkan *prediction error* selanjutnya.

Kelima, setelah seluruh *prediction error* selesai diproses, seluruh *frame* di-*recovery* seperti sebelum dilakukan penggeseran untuk menghindari kondisi *underflow* / *overflow*. Proses *recovery frame* dilakukan menggunakan menggunakan Persamaan 3.33 di bawah ini.

$$\begin{aligned} \alpha_{x,y,z} &= \alpha_{x,y,z} - 1, \text{ jika } \alpha_{x,y,z} < \theta_1 \\ \alpha_{x,y,z} &= \alpha_{x,y,z} + 1, \text{ jika } \alpha_{x,y,z} > \theta_2 \end{aligned} \quad (3.33)$$

Sedangkan untuk  $\alpha_{x,y,z} = \theta_1$  atau  $\alpha_{x,y,z} = \theta_2$ , proses *recovery* dilakukan berdasarkan informasi *overhead* yang didapatkan dari *payload*.

### 3.4 Implementasi Penelitian

Tahapan implementasi merupakan tahapan realisasi terhadap algoritma yang telah dirancang. Realisasi yang dimaksud adalah dengan menerapkan algoritma yang telah dirancang ke sebuah bahasa pemrograman sehingga algoritma tersebut dapat benar-benar digunakan untuk melakukan penyisipan. Implementasi pada penelitian ini menggunakan *hardware* berupa PC dengan spesifikasi Intel Celeron Processor 1000M (1,8 GHz) dan 2 GB DDR3 SDRAM. Sedangkan bahasa pemrograman yang digunakan adalah Java dengan *Integrated Development Environment* (IDE) berupa NetBeans 8.1.

Tahapan implementasi meliputi penulisan kode program, *compiling*, *debugging* sampai dihasilkan kode program yang siap dieksekusi. Setelah seluruh tahapan implementasi terhadap metode yang diusulkan selesai, barulah metode usulan dapat diuji. Selain itu, penelitian ini juga mengimplementasikan metode *histogram shifting* (Ni et al., 2006) yang diadaptasikan terhadap video dan metode *neighbouring similarity* (Yeh et al., 2014) sebagai pembanding.

### 3.5 Perancangan Uji Coba dan Analisis Hasil

Uji coba dilakukan untuk membuktikan hipotesis yang telah dijelaskan pada Bab I. Pengujian dilakukan terhadap dua hal, yaitu tingkat kemiripan / *similarity* dan kapasitas pesan rahasia yang dapat ditampung. Penjelasan dari keduanya adalah sebagai berikut:

- a. Tingkat kemiripan / *similarity*: Metode *Peak to Signal Noise Ratio* (PSNR) digunakan untuk menilai tingkat kemiripan video hasil penyisipan (*stego data*) dengan video asli sebelum disisipi pesan rahasia (*cover data*). Semakin tinggi nilai PSNR, maka semakin mirip *stego data* terhadap *cover data*, yang berarti semakin bagus metode tersebut. Nilai yang dikeluarkan PSNR juga dapat diklasifikasikan menjadi tiga kategori, yaitu kualitas rendah (di bawah 30 dB), kualitas sedang (di atas 30 dB tetapi di bawah 40 dB) dan kualitas tinggi (di atas 40 dB) (Cheddad, et al., 2010).

- b. Kapasitas: Cara mengukur kapasitas adalah dengan mengurangi jumlah bit maksimum yang dapat ditampung dengan jumlah bit informasi *overhead* yang harus disertakan pada saat penyisipan. Dalam metode *histogram shifting* beserta turunannya, jumlah bit maksimum diwakili oleh nilai *peak point*. Sedangkan jumlah bit informasi *overhead* didapatkan saat proses penggeseran sebelum penyisipan.

Uji coba dilakukan terhadap 20 contoh video digital yang disediakan oleh [xiph.org](http://xiph.org). Video-video yang tersedia pada *website* xiph.org merupakan contoh video digital yang sudah sering digunakan pada penelitian-penelitian yang berhubungan dengan video digital. Referensi utama yang digunakan pada tesis ini, yaitu metode *neighbouring similarity*, juga menggunakan video digital sejenis.

Data uji yang berupa video digital disajikan dalam ekstensi y4m. Tetapi, penelitian ini memformat ulang video uji tersebut menggunakan codec mjpeg pada program FFmpeg sebelum digunakan. Tujuannya adalah untuk mempermudah pemrosesan *frame* pada video uji. Ekstensi yang digunakan setelah format ulang adalah Matroska Video (.mkv). Ekstensi ini digunakan karena bersifat *open source*, sehingga penggunaannya lebih bebas dibandingkan ekstensi-ekstensi *proprietary*.

Pesan rahasia yang digunakan adalah teks *lorem ipsum* dengan panjang 3971 karakter (sekitar 27.797 bit jika 1 karakter adalah 7 bit). Teks *lorem ipsum* merupakan teks pengganti yang sering digunakan sebagai data sementara, seperti pada *template website*, rancangan aplikasi, dan berbagai keperluan IT lainnya. Oleh karena itu, penggunaan teks ini dianggap dapat mewakili karakteristik pesan rahasia.

Uji coba dilakukan dengan dua skenario. Pemilihan skenario ini berdasarkan jenis warna pada video uji. Skenario pertama, *secret message* disisipkan pada komponen warna *grayscale* sehingga video uji diformat ulang ke skema warna *grayscale* sebelum disisipi pesan rahasia. Skenario kedua, *secret message* disisipkan pada komponen warna merah. Penggunaan kedua skenario ini bertujuan untuk menentukan kondisi seperti apa yang membuat metode-metode usulan bekerja paling optimal, apakah pada video *grayscale* atau video berwarna.

*[Halaman ini sengaja dikosongkan]*

## **BAB 4**

### **HASIL DAN PEMBAHASAN**

Pada bab ini dijelaskan hasil eksperimen beserta pembahasannya. Lingkungan uji coba menjelaskan detail dari spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan. Pelaksanaan uji coba menjelaskan secara detail bagaimana pengujian dilakukan. Terakhir, hasil dan evaluasi menjelaskan output yang didapatkan dari eksperimen penelitian ini.

#### **4.1 Lingkungan Uji Coba**

Spesifikasi perangkat keras yang digunakan pada penelitian ini adalah sebuah Notebook PC dengan processor Intel Celeron 1000M (1,8 GHz) dan memori 2 GB DDR3 SDRAM. Sedangkan spesifikasi perangkat lunak yang digunakan adalah bahasa pemrograman Java SE (*Standart Edition*) versi 1.8 dengan IDE (*Integrated Development Environment*) berupa NetBeans 8.1. Selain itu, codec yang digunakan untuk memproses video adalah MJPEG pada aplikasi FFmpeg versi 20160110-git-42c54d4.

#### **4.2 Pelaksanaan Uji Coba**

Selain dari tiga metode yang diusulkan, penelitian ini juga mengimplementasikan dua metode yang telah ada sebagai pembanding, yaitu metode *histogram shifting* dan metode *neighbouring similarity*. Karena metode *histogram shifting* pada penelitian sebelumnya digunakan pada citra digital, maka metode tersebut diadaptasikan terhadap video digital. Caranya adalah dengan menggunakan *frame* pada video untuk menggantikan citra digital sebagai media penyisipan.

Data uji yang digunakan adalah contoh video digital yang didapatkan dari [xiph.org](http://xiph.org). *Proposed 1*, *Proposed 2* dan *Proposed 3* secara berturut-turut adalah

metode usulan pertama (*Neighbouring Similarity of Triplets*), metode usulan kedua (*Multi-Level Neighbouring Similarity*) dan metode usulan ketiga (*Overlapped Neighbouring Similarity*) seperti yang telah dijelaskan pada subbab 3.3.1 sampai 3.3.3. Sedangkan HS adalah singkatan untuk *Histogram Shifting* dan NS adalah singkatan untuk *Neighbouring Similarity*. Contoh *frame* dari tiap video uji yang digunakan dapat dilihat pada Gambar 4.1.



**Gambar 4.1. Contoh *frame* pada tiap video uji.**



Skenario pertama adalah menyisipkan teks *lorem ipsum* pada video uji yang memiliki skema warna *grayscale*. Seluruh *frame* pada video uji diekstrak menggunakan FFmpeg ke citra digital dengan format *Portable Network Graphics* (PNG). Kemudian, setiap citra digital dikonversi kedalam skema warna *grayscale* menggunakan Persamaan 4.1 di bawah ini.

$$\alpha_{x,y,z} = \left\lfloor \frac{R_{x,y,z} + G_{x,y,z} + B_{x,y,z}}{3} \right\rfloor \quad (4.1)$$

Di mana  $\alpha$  adalah nilai piksel,  $R$  adalah nilai warna merah,  $G$  adalah nilai warna hijau,  $B$  adalah nilai warna biru, dan  $x$ - $y$ - $z$  adalah koordinat. Setelah seluruh *frame* selesai dikonversi, barulah *frame* yang ada diproses ulang menjadi video berformat Matroska Video (.mkv) menggunakan codec MJPEG pada aplikasi FFmpeg.

Skenario kedua adalah menyisipkan teks *lorem ipsum* pada komponen warna merah dari video uji. Seluruh *frame* pada video uji diekstrak menggunakan FFmpeg ke citra digital dengan format PNG tanpa dikonversi kedalam skema warna *grayscale*. *Secret message* yang berupa teks *lorem ipsum* disisipkan pada komponen warna merah dari video uji, di mana komponen warna hijau dan warna biru tidak ikut diubah. Proses konversi pada skenario kedua ini hanya bertujuan mengubah format dan ekstensi saja tanpa mengubah skema warna yang digunakan.

Setelah didapatkan video uji dengan ekstensi Matroska Video, proses penyisipan *secret message* dijalankan menggunakan masing-masing metode yang telah dibahas sebelumnya. Metode-metode tersebut adalah dua buah metode yang telah ada (yaitu *Histogram Shifting* dan *Neighbouring Similarity*) dan tiga buah metode usulan (yaitu *Proposed 1*, *Proposed 2* dan *Proposed 3*). Pesan yang disisipkan adalah teks *lorem ipsum* dengan panjang sekitar 27.797 bit. Pada saat proses penyisipan pesan juga ditentukan berapa kapasitas pesan yang dapat ditampung pada video uji terkait. Penentuan kapasitas diwakilkan oleh Persamaan 4.2.

$$C = p - \theta \quad (4.2)$$

Di mana  $C$  adalah kapasitas *secret message* (dalam satuan bit),  $p$  adalah nilai *peak point* dan  $\theta$  adalah panjang informasi *overhead*. Sebuah *stego data* berupa video digital akan didapatkan setelah proses penyisipan selesai.

Metode PSNR digunakan untuk menentukan tingkat kemiripan / *similarity* dari *stego data* terhadap *cover data*-nya seperti yang telah dijelaskan pada subbab 2.5. Tetapi, perhitungannya tidak dilakukan secara manual. Penentuan nilai PSNR pada penelitian ini menggunakan bantuan aplikasi FFmpeg. Nilai keluaran FFmpeg yang digunakan adalah nilai rata-rata (*average*).

### 4.3 Hasil dan Evaluasi

Evaluasi terhadap kualitas metode-metode usulan dilakukan terhadap dua parameter, yaitu kapasitas (dalam satuan bit) dan tingkat kemiripan / *similarity* (menggunakan metode PSNR). Semakin tinggi keduanya, baik kapasitas maupun PSNR, mengindikasikan bahwa semakin baik kualitas suatu metode *steganography*.

#### 4.3.1 Skenario 1

Hasil pengukuran kapasitas untuk skenario pertama dapat dilihat pada Tabel 4.1. Sedangkan hasil pengukuran PSNR untuk skenario pertama dapat dilihat pada Tabel 4.2. Berdasarkan nilai rata-rata dari hasil pengukuran kapasitas, *Proposed 3* terlihat menghasilkan kapasitas tertinggi diantara kelima metode yang diuji. *Proposed 3* mampu menyediakan kapasitas dua kali lipat metode NS dan tiga kali lipat metode HS. Sedangkan metode yang memiliki kapasitas terendah adalah *Proposed 2*. Kapasitas yang dihasilkan oleh *Proposed 2* hanya setengah dari kapasitas yang dihasilkan metode HS.

Peningkatan pada *Proposed 3* terjadi karena metode ini menggunakan *prediction error* sebagai acuan untuk membangkitkan *histogram*, tetapi mampu menggunakan jumlah *prediction error* yang hampir sama dengan jumlah *frame*. Sedangkan metode NS hanya menggunakan setengah dari jumlah *frame* yang ada. *Proposed 2* menjadi metode dengan kapasitas paling rendah karena jumlah

Tabel 4.1. Hasil pengukuran kapasitas skenario pertama.

Video	HS	NS	Proposed 1	Proposed 2	Proposed 3
bridge_close_qcif	9.327.587	6.793.050	8.880.817	3.718.327	13.315.463
bus_qcif_15fps	34.248	62.560	85.172	28.542	126.462
coastguard_cif	334.572	1.312.405	1.764.642	702.417	2.657.565
flower_cif	9.238.102	5.114.564	6.811.966	2.317.520	10.213.826
football_qcif_15fps	99.548	145.820	192.077	65.885	281.355
foreman_cif	360.135	2.654.193	3.517.979	1.352.069	5.264.724
garden_sif	177.335	388.677	530.526	116.710	812.684
hall_monitor_qcif	337.109	1.019.100	1.359.403	567.980	2.034.821
highway_qcif	7.837.221	5.295.392	7.356.402	2.811.120	11.066.445
mobile_sif	223.347	1.681.086	2.270.429	690.488	3.439.687
mother_daughter_qcif	296.938	1.406.423	1.876.268	229.004	2.803.877
news_qcif	385.407	2.297.302	3.069.020	1.069.832	4.592.935
paris_qcif	338.802	5.249.391	7.009.323	2.724.783	10.511.189
silent_qcif	113.984	2.332.370	3.103.193	919.809	4.663.855
stefan_sif	585.387	2.106.974	2.809.940	1.050.122	4.200.381
suzie_qcif	89.233	533.125	712.816	293.444	1.059.754
tempete_cif	411.648	2.528.778	3.349.973	1.380.695	5.040.716
tt_sif	207.948	818.847	1.099.629	412.916	1.673.661
washdc_422_qcif	126.681	1.730.772	2.301.881	960.210	3.446.598
waterfall_cif	468.339	2.774.982	3.665.094	1.589.773	5.527.889
rata-rata	1.549.679	2.312.291	3.088.328	1.150.082	4.636.694

*prediction error* yang digunakan jauh lebih sedikit dibandingkan metode NS. Sebaran data yang lebih baik ternyata belum mampu mengakomodasi jumlah *prediction error* level 2 yang jauh lebih sedikit dibandingkan *prediction error* level 1 (yang digunakan pada metode NS).

Video	HS	NS	Proposed 1	Proposed 2	Proposed 3
bridge_close_qcif	9.327.587	6.793.050	8.880.817	3.718.327	13.315.463
bus_qcif_15fps	34.248	62.560	85.172	28.542	126.462
coastguard_cif	334.572	1.312.405	1.764.642	702.417	2.657.565
flower_cif	9.238.102	5.114.564	6.811.966	2.317.520	10.213.826
football_qcif_15fps	99.548	145.820	192.077	65.885	281.355
foreman_cif	360.135	2.654.193	3.517.979	1.352.069	5.264.724
garden_sif	177.335	388.677	530.526	116.710	812.684
hall_monitor_qcif	337.109	1.019.100	1.359.403	567.980	2.034.821
highway_qcif	7.837.221	5.295.392	7.356.402	2.811.120	11.066.445
mobile_sif	223.347	1.681.086	2.270.429	690.488	3.439.687
mother_daughter_qcif	296.938	1.406.423	1.876.268	229.004	2.803.877
news_qcif	385.407	2.297.302	3.069.020	1.069.832	4.592.935
paris_qcif	338.802	5.249.391	7.009.323	2.724.783	10.511.189
silent_qcif	113.984	2.332.370	3.103.193	919.809	4.663.855
stefan_sif	585.387	2.106.974	2.809.940	1.050.122	4.200.381
suzie_qcif	89.233	533.125	712.816	293.444	1.059.754
tempete_cif	411.648	2.528.778	3.349.973	1.380.695	5.040.716

Walaupun bukan yang terbaik, *Proposed 1* merupakan metode yang lebih baik dibandingkan dengan metode HS, NS, atau *Proposed 2* dan *Proposed 3* kapasitas *Proposed 1* juga memiliki kelebihan lain jika dibandingkan dengan *Proposed 3*, yaitu proses ekstraksi yang lebih sederhana. *Proposed 1* mampu menggambil skema yang sama dengan NS untuk proses ekstraksi dan *Proposed 1* tidak perlu diproses satu per satu tiap *frame* seperti pada *Proposed 3*. Hal ini menyebabkan kecepatan ekstraksi *Proposed 1* kemungkinan besar akan lebih cepat dibandingkan dengan *Proposed 3*.

Tabel 4.224. Hasil pengukuran PSNR skenario kedua.

Video	HS	NS	Proposed 1	Proposed 2	Proposed 3
bridge_close_qcif	54,36	52,21	51,93	47,42	51,36
bus_qcif_15fps	54,21	52,11	51,69	47,22	51,13
coastguard_cif	59,70	52,24	51,86	47,32	51,28
flower_cif	55,48	52,32	52,03	47,68	51,58
football_qcif_15fps	57,22	52,21	51,85	47,19	51,23
foreman_cif	54,73	52,38	52,06	47,32	51,51
garden_sif	53,56	52,24	51,83	47,33	51,29
hall_monitor_qcif	53,64	52,52	52,29	47,32	51,81
highway_qcif	54,19	52,17	51,85	47,43	51,27
mobile_sif	65,00	52,29	51,95	47,21	51,38
mother_daughter_qcif	59,45	60,44	59,41	55,09	57,91
news_qcif	55,19	52,77	52,56	48,09	52,22
paris_qcif	59,21	52,49	52,21	47,06	51,66
silent_qcif	68,20	58,65	57,42	47,38	55,69
stefan_sif	55,71	52,34	52,01	47,28	51,45
suzie_qcif	56,33	52,68	52,42	47,10	52,04
tempete_cif	53,72	52,36	52,09	47,16	51,50
tt_sif	56,59	52,38	52,12	47,25	51,52
washdc_422_qcif	59,53	62,29	61,20	55,97	59,31
waterfall_cif	73,58	52,45	52,12	47,14	51,64
rata-rata	57,98	53,58	53,15	48,15	52,44

Jika dilihat dari rata-rata nilai PSNR, metode HS masih merupakan metode dengan nilai PSNR terbaik dibandingkan keempat metode yang diuji lainnya. Ini dikarenakan metode HS tidak memiliki kemungkinan *underflow* / *overflow* sehingga tidak perlu melakukan penggeseran sebelum penyisipan (yang merupakan langkah pencegahan *underflow* / *overflow*). Rata-rata nilai PSNR terendah adalah *Proposed 2*. Ini dikarenakan *Proposed 2* harus melakukan penggeseran sebanyak dua poin untuk mencegah *underflow* / *overflow* pada *prediction error level 2*. Walaupun menghasilkan PSNR yang lebih rendah, ketiga metode usulan mempertahankan rata-rata nilai PSNR di atas 40 dB. Oleh karena itu, ketiga data yang dihasilkan masih tergolong sebagai kualitas tinggi. Selain itu, *Proposed 3* masih lebih baik dibandingkan metode NS jika dilihat dari perubahan rata-rata nilai PSNR. Rata-rata NS diusulkan sebesar 50,44 dB, 50,09 dB, dan 51,50 dB.

Video	HS	NS	Proposed 1	Proposed 2	Proposed 3
bridge_close_qcif	54,36	52,21	51,93	47,42	51,36
bus_qcif_15fps	54,21	52,11	51,69	47,22	51,13
coastguard_cif	59,70	52,24	51,86	47,32	51,28
flower_cif	55,48	52,32	52,03	47,68	51,58
football_qcif_15fps	57,22	52,21	51,85	47,19	51,23
foreman_cif	54,73	52,38	52,06	47,32	51,51
garden_sif	53,56	52,24	51,83	47,33	51,29
hall_monitor_qcif	53,64	52,52	52,29	47,32	51,81
highway_qcif	54,19	52,17	51,85	47,43	51,27
mobile_sif	65,00	52,29	51,95	47,21	51,38
mother_daughter_qcif	59,45	60,44	59,41	55,09	57,91
news_qcif	55,19	52,77	52,56	48,09	52,22
paris_qcif	59,21	52,49	52,21	47,06	51,66
silent_qcif	68,20	58,65	57,42	47,38	55,69
stefan_sif	55,71	52,34	52,01	47,28	51,45
suzie_qcif	56,33	52,68	52,42	47,10	52,04
tempete_cif	53,72	52,36	52,09	47,16	51,50

kapasitas sekitar 49,21% dari metode sebelumnya (yaitu HS), sedangkan *Proposed 3* hanya perlu mengorbankan 1,14 dB untuk menambahkan kapasitas sekitar 100,52% dari metode NS.

#### 4.3.2 Skenario 2

Hasil pengukuran kapasitas untuk skenario kedua dapat dilihat pada Tabel 4.3. Sedangkan hasil pengukuran PSNR untuk skenario kedua dapat dilihat pada Tabel 4.4. Berdasarkan nilai rata-rata dari hasil pengukuran kapasitas, *Proposed 3* terlihat menghasilkan kapasitas tertinggi diantara kelima metode yang diuji. Seperti pada skenario pertama, *Proposed 3* mampu menyediakan kapasitas dua kali lipat metode NS dan tiga kali lipat metode HS. Sedangkan metode yang memiliki kapasitas terendah adalah *Proposed 2*. Kapasitas yang dihasilkan oleh *Proposed 2* hanya setengah dari kapasitas yang dihasilkan metode HS.

**Tabel 4.385. Hasil pengukuran kapasitas skenario kedua.**

Video	HS	NS	Proposed 1	Proposed 2	Proposed 3
bridge_close_qcif	9,488,588	6,042,364	8,057,965	3,171,076	12,082,398
bus_qcif_15fps	31,052	54,511	74,140	25,644	110,185
coastguard_cif	337,165	1,265,973	1,701,298	700,861	2,563,551
flower_cif	11,327,728	5,281,056	7,026,244	2,272,800	10,534,211
football_qcif_15fps	77,593	130,668	171,316	62,996	251,241
foreman_cif	745,721	2,317,030	3,070,584	1,261,961	4,594,487
garden_sif	610,849	330,195	459,246	43,976	717,216
hall_monitor_qcif	583,674	693,187	923,020	425,529	1,380,456
highway_qcif	9,988,000	5,933,139	7,944,934	2,874,796	11,948,125
mobile_sif	686,497	1,169,797	1,675,511	612,636	2,676,711
mother_daughter_qcif	258,985	1,125,504	1,500,468	138,247	2,243,152
news_qcif	399,558	2,273,522	3,040,076	1,037,697	4,550,939
paris_qcif	1,341,370	4,946,412	6,603,630	2,283,371	9,903,002
silent_qcif	153,951	2,196,953	2,925,139	822,192	4,395,569
stefan_sif	731,874	1,813,750	2,419,974	948,283	3,618,704
suzie_qcif	87,590	469,914	629,489	272,889	934,666
tempete_cif	418,405	2,033,339	2,709,503	1,091,888	4,098,200
tt_sif	223,450	731,018	980,142	324,493	1,492,956
washdc_422_qcif	151,023	1,512,967	2,012,169	885,997	3,013,443
waterfall_cif	434,715	2,324,761	3,083,861	1,351,305	4,669,640
<b>average</b>	<b>1,903,889</b>	<b>2,132,303</b>	<b>2,850,435</b>	<b>1,030,432</b>	<b>4,288,943</b>

Tabel 4.448. Hasil pengukuran PSNR skenario kedua.

Video	HS	NS	Proposed 1	Proposed 2	Proposed 3
bridge_close_qcif	56.01	54.69	54.38	50.25	53.76
bus_qcif_15fps	53.28	53.33	53.48	50.02	53.89
coastguard_cif	58.98	53.49	53.55	50.05	53.75
flower_cif	58.46	56.37	56.03	51.08	55.48
football_qcif_15fps	59.42	54.79	54.39	50.39	53.74
foreman_cif	57.06	55.49	55.07	50.76	54.40
garden_sif	56.30	54.77	54.36	50.50	53.79
hall_monitor_qcif	56.80	55.28	54.96	50.69	54.35
highway_qcif	57.27	55.62	55.04	50.82	54.35
mobile_sif	53.12	53.36	53.48	50.09	53.74
mother_daughter_qcif	95.38	56.24	55.64	51.06	55.13
news_qcif	54.64	52.64	52.70	50.39	52.85
paris_qcif	56.54	55.07	54.76	50.68	54.19
silent_qcif	57.25	55.92	55.56	51.26	54.96
stefan_sif	56.82	55.32	54.94	50.75	54.29
suzie_qcif	62.14	56.55	56.13	51.09	55.53
tempete_cif	56.59	53.24	53.37	50.11	53.61
tt_sif	57.01	55.50	55.15	50.82	54.49
washdc_422_qcif	59.19	56.68	56.41	51.49	55.96
waterfall_cif	52.97	53.27	53.41	50.18	53.60
<b>average</b>	<b>58.76</b>	<b>54.88</b>	<b>54.64</b>	<b>50.62</b>	<b>54.29</b>

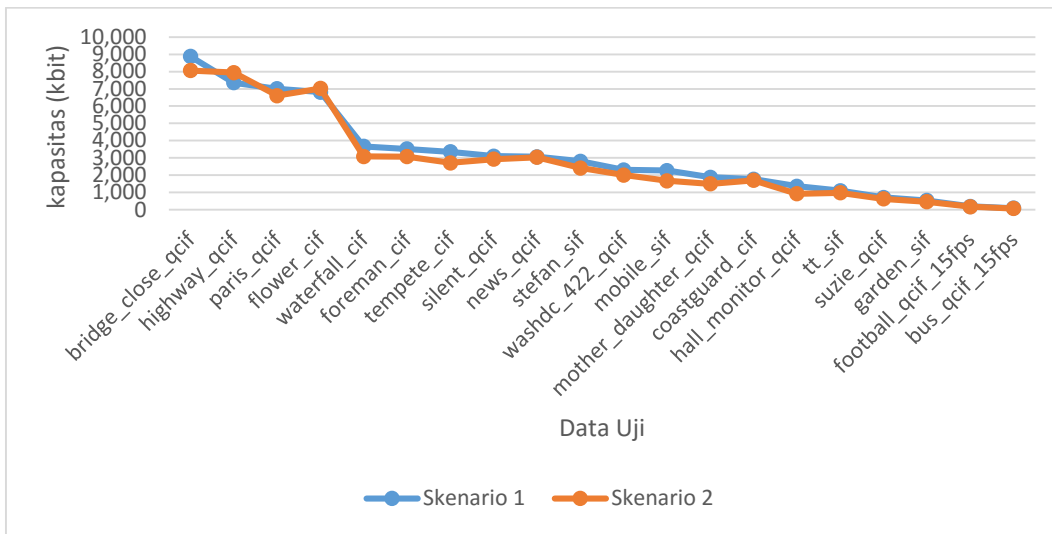
Jika dilihat dari rata-rata nilai PSNR, metode HS masih merupakan metode dengan nilai PSNR terbaik dibandingkan keempat metode yang diuji lainnya. Rata-rata nilai PSNR terendah adalah *Proposed 2* sama seperti pada skenario pertama. Walaupun menghasilkan PSNR yang lebih rendah, ketiga metode usulan masih mampu mempertahankan rata-rata nilai PSNR di atas 40 dB. Oleh karena itu, *stego data* yang dihasilkan masih tergolong sebagai kualitas tinggi. Selain itu, *Proposed 3* masih lebih baik dibandingkan metode NS jika dilihat dari perubahan rata-rata nilai PSNR. Metode NS harus mengorbankan 3,88 dB untuk menambahkan kapasitas sekitar 12,00% dari metode sebelumnya (yaitu HS), sedangkan *Proposed 3* hanya perlu mengorbankan 0,59 dB untuk menambahkan kapasitas sekitar 101,14% dari metode NS.

*Proposed 1* menghasilkan nilai PSNR yang lebih rendah dibandingkan metode yang telah ada sebelumnya (HS dan NS), tetapi menghasilkan nilai yang paling tinggi diantara ketiga metode usulan. Hal ini karena *Proposed 1* menghasilkan lebih banyak *frame* yang tidak diubah (*reference macro block*) dibandingkan *Proposed 3*, di mana *Proposed 1* menghasilkan 1 buah *reference macro block* dan *Proposed 3* hanya menghasilkan sebuah *reference macro block*, yaitu *frame* paling akhir. *Proposed 1* juga menggeser nilai *frame* sebanyak 1 poin untuk mencegah masalah *underflow / overflow*, di mana *Proposed 2* menggeser sebanyak 2 poin.

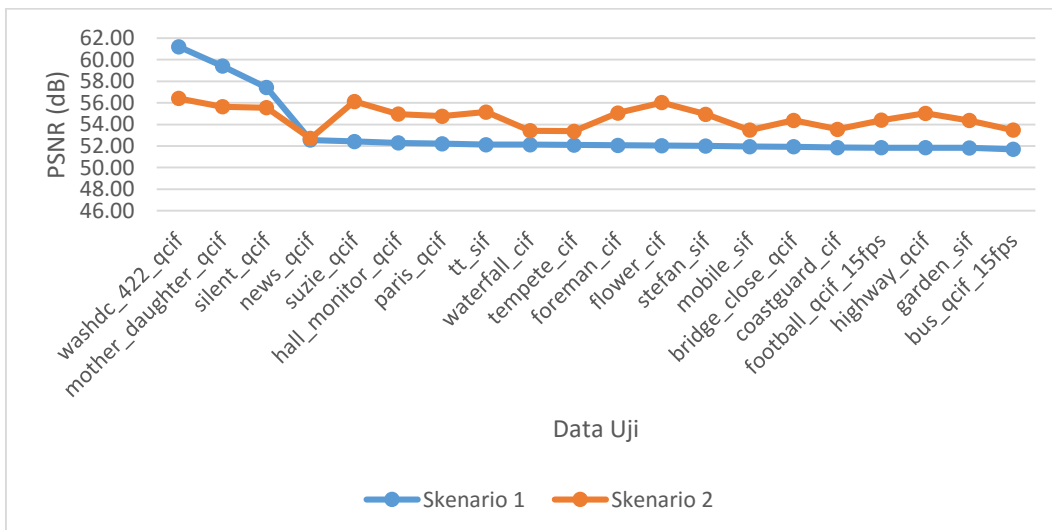
Untuk mempermudah proses analisis, setiap metode usulan pada kedua skenario disajikan dalam bentuk grafik. Setiap grafik mewakili hasil eksperimen untuk setiap metode pada sebuah penilaian. Sumbu-*x* pada grafik menunjukkan video uji yang digunakan, sedangkan sumbu-*y* mewakili nilai yang dihasilkan untuk setiap video uji. Warna biru mewakili video *grayscale* karena menunjukkan nilai yang dihasilkan untuk skenario pertama yang melibatkan komponen warna *grayscale* untuk penyisipan pesan. Warna jingga mewakili video berwarna karena menunjukkan nilai yang dihasilkan untuk skenario kedua, di mana skenario kedua menggunakan komponen warna merah pada video berwarna.

Tampilan video uji pada tiap grafik diurutkan berdasarkan jumlah kapasitas yang dihasilkan pada skenario pertama. Urutan yang digunakan adalah *descending*, yaitu data uji dengan hasil kapasitas tertinggi ditampilkan terlebih dahulu. Tujuan dari proses ini adalah untuk menyederhanakan visualisasi perbandingan skenario, sehingga lebih mudah dilihat secara langsung.

Gambar 4.2 merupakan grafik untuk kapasitas metode usulan pertama, sedangkan Gambar 4.3 merupakan grafik untuk PSNR pada metode usulan pertama. Gambar 4.4 adalah grafik untuk kapasitas metode usulan pertama dan Gambar 4.5 adalah grafik untuk tingkat PSNR pada metode usulan kedua. Gambar 4.6 dan Gambar 4.7 masing-masing adalah grafik untuk kapasitas dan PSNR pada metode usulan ketiga. Untuk mempermudah pembacaan, data uji yang ditampilkan pada seluruh grafik diurutkan berdasarkan nilai skenario pertama secara *ascending*.



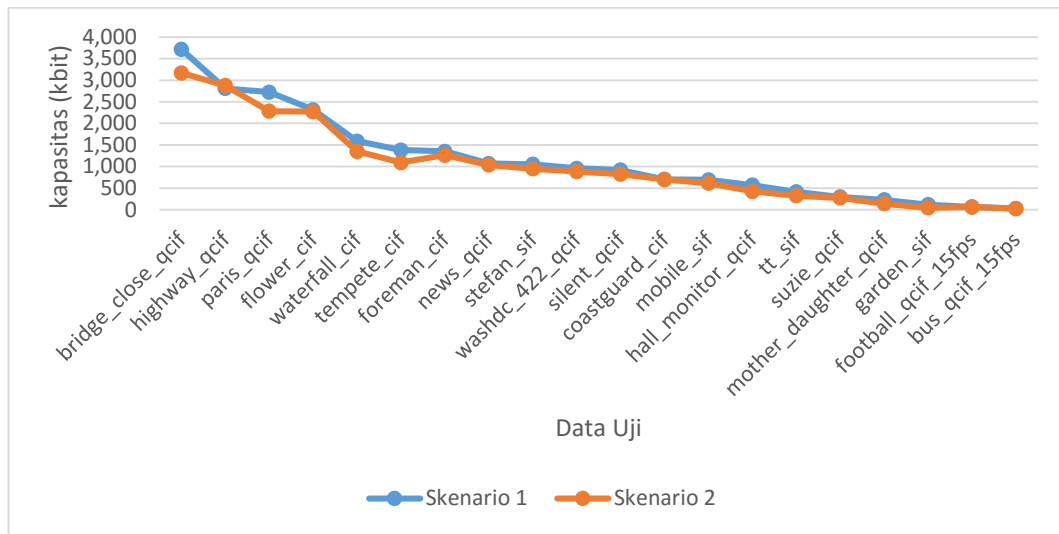
Gambar 4.2. Hasil pengukuran kapasitas untuk *Proposed 1*.



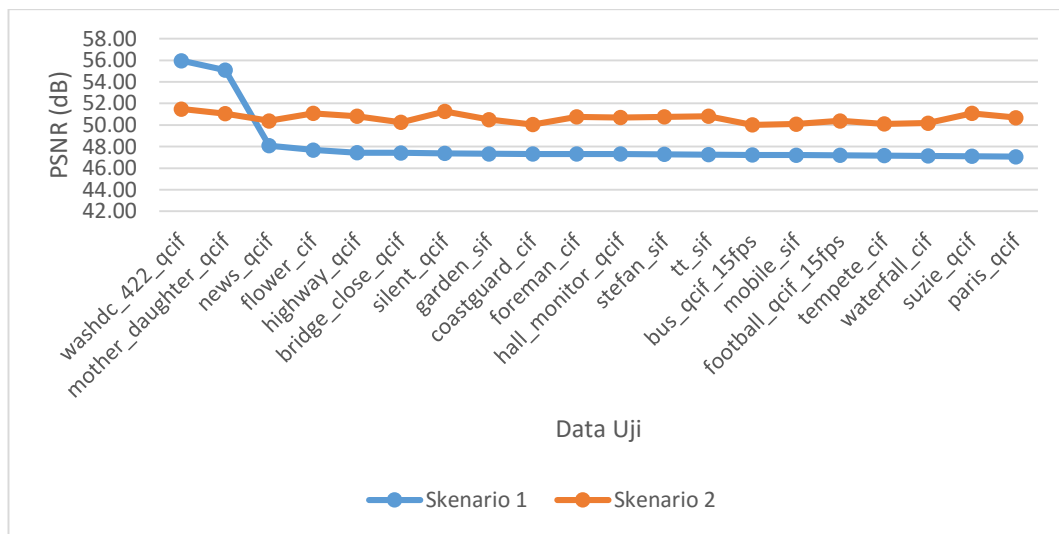
Gambar 4.3. Hasil pengukuran PSNR untuk *Proposed 1*.

Gambar 4.2 menunjukkan bahwa pada *Proposed 1*, sebagian besar nilai skenario pertama lebih tinggi dibandingkan skenario kedua. Ini berbanding terbalik dengan Gambar 4.3 yang menunjukkan lebih banyak nilai-nilai pada skenario kedua yang berada di atas skenario pertama. Kondisi ini menunjukkan bahwa *Proposed 1* mampu menghasilkan kapasitas yang lebih baik terhadap video dengan skema warna *grayscale* dibandingkan video berwarna.



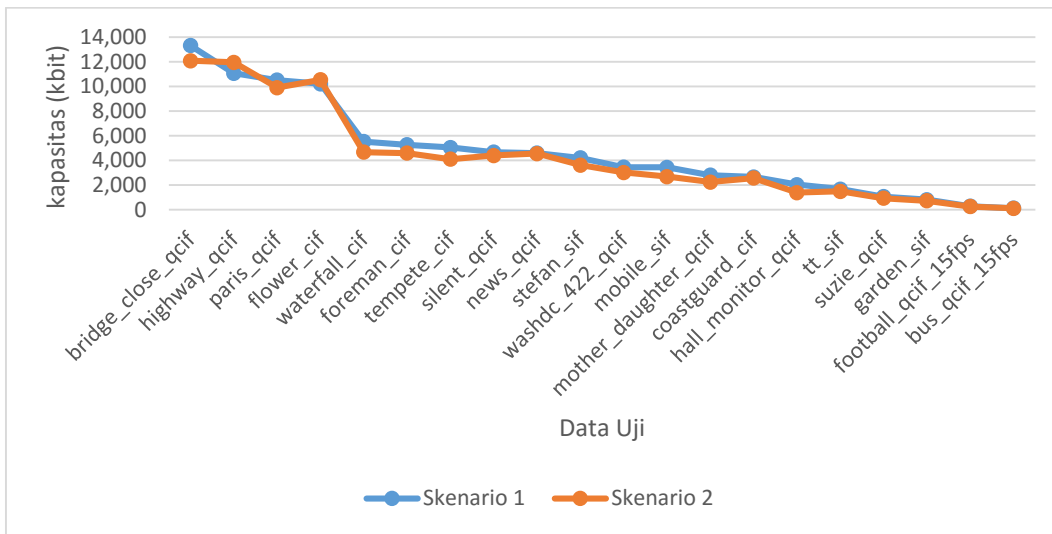


Gambar 4.4. Hasil pengukuran kapasitas untuk *Proposed 2*.

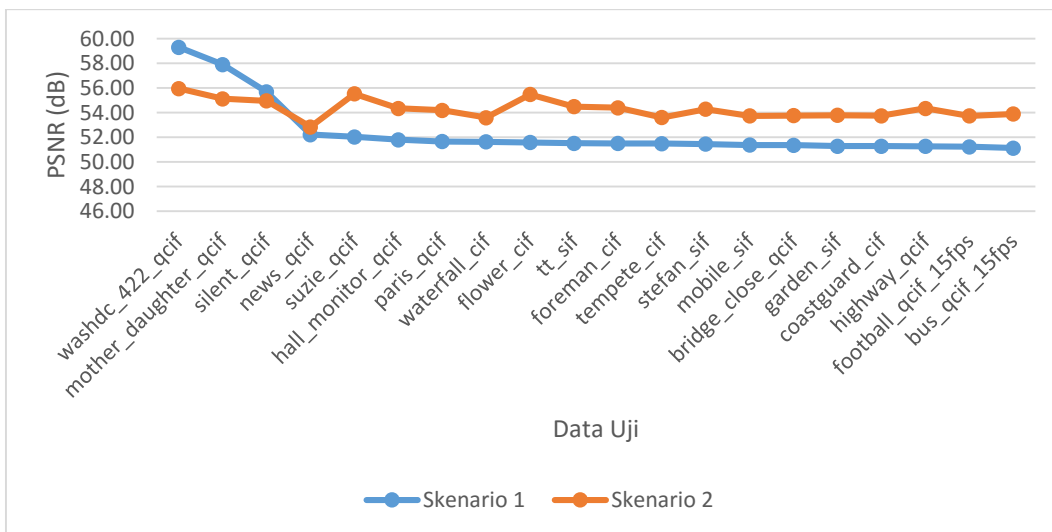


Gambar 4.5. Hasil pengukuran PSNR untuk *Proposed 2*.

Gambar 4.4 dan Gambar 4.5 juga menunjukkan bahwa skenario pertama lebih unggul dalam hal kapasitas, tetapi skenario kedua lebih unggul dalam hal PSNR. Sebagian besar nilai kapasitas skenario pertama menghasilkan nilai di atas skenario kedua, dan sebaliknya untuk nilai PSNR. Artinya, *Proposed 2* mampu menghasilkan kapasitas yang baik pada video dengan skema warna *grayscale* dibandingkan dengan video berwarna, tetapi tingkat kesamaan / *similarity* yang dihasilkan akan lebih baik pada video berwarna.



Gambar 4.6. Hasil pengukuran kapasitas untuk *Proposed 3*.



Gambar 4.7. Hasil pengukuran PSNR untuk *Proposed 3*.

Kondisi yang sama terlihat juga pada *Proposed 3*. Sebagian besar skenario pertama terlihat menghasilkan nilai yang lebih tinggi untuk pengukuran kapasitas. Sedangkan untuk pengukuran PSNR, sebagian besar skenario kedua lah yang menghasilkan nilai di atas skenario pertama. Video dengan skema warna *grayscale* mampu menghasilkan kapasitas yang lebih baik dibandingkan video berwarna, tetapi video berwarna mampu menghasilkan tingkat kesamaan / *similarity* yang lebih baik dibandingkan video *grayscale*.

Semakin tinggi nilai kapasitas pada video *grayscale* adalah karena video *grayscale* memiliki sebaran data pada komponen warna yang lebih kecil daripada video berwarna. Kemungkinan nilai piksel video *grayscale* adalah 0 hingga 255, sedangkan video warna adalah  $255^3$  karena sebuah piksel pada video berwarna terdiri dari tiga buah komponen warna (merah, hijau dan biru). Untuk nilai PSNR, video berwarna lebih unggul karena komponen yang berubah warna hanya salah satu saja, di mana pada penelitian ini digunakan komponen warna merah. Kedua komponen sisanya memiliki nilai yang sama dengan sebelum disisipi *secret message*.

*[Halaman ini sengaja dikosongkan]*

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Pada bab ini diuraikan mengenai kesimpulan yang dapat diambil dari penelitian ini. Selain itu diuraikan juga beberapa saran yang dapat diperhatikan untuk pengembangan selanjutnya.

#### **5.1 Kesimpulan**

Berdasarkan eksperimen yang telah dilakukan, kesimpulan yang dapat ditarik dari penelitian ini adalah:

- a. Penelitian ini mengusulkan tiga buah metode penyisipan, di mana dua diantaranya (yaitu *Neighbouring Similarity of Triplets* dan *Overlapped Neighbouring Similarity*) mampu menghasilkan kapasitas yang lebih baik dibandingkan kedua metode pendahulunya (yaitu *Histogram Shifting* dan *Neighbouring Similarity*).
- b. Dari ketiga metode yang diusulkan pada penelitian ini, *overlapped neighbouring similarity* merupakan metode yang menghasilkan kapasitas paling baik diantara metode usulan lainnya maupun metode pendahulunya.
- c. Nilai PSNR yang dihasilkan dari ketiga metode usulan lebih rendah dibandingkan metode yang telah ada (*Neighbouring Similarity* dan *Histogram Shifting*). Tetapi, nilai yang dihasilkan masih di atas 40 dB sehingga pengurangan tersebut masih dapat ditoleransi. *Stego data* yang dihasilkan masih dapat dikategorikan sebagai *high quality* jika nilainya di atas 40 dB.

#### **5.2 Saran**

Metode usulan kedua (yaitu *Multi-Level Neighbouring Similarity*) pada penelitian ini menghasilkan kapasitas dan PSNR yang paling rendah. Bahkan

kapasitas dan PSNR tersebut juga lebih rendah dibandingkan metode pendahulunya. Padahal, *prediction error level 2* yang dihasilkan seharusnya lebih baik seperti yang dijelaskan pada metode penelitian. Kemungkinan permasalahannya terletak pada jumlah *prediction error* yang lebih sedikit. Jika *prediction error* yang dihasilkan mampu dimodifikasi agar menghasilkan jumlah yang lebih tinggi dari usulan pada penelitian ini, maka kapasitas yang dihasilkan pun mungkin dapat menjadi lebih besar.

Permasalahan lain yang bisa dikembangkan adalah cara mengatasi *underflow / overflow*. Walaupun masalah tersebut sudah teratasi dengan melakukan penggeseran sebelum penyisipan pesan untuk mengosongkan nilai 0 dan 255, tetapi cara tersebut membuat nilai PSNR menjadi berkurang. Hal ini karena jumlah piksel yang harus diubah nilainya lebih banyak dibandingkan metode *histogram shifting*. Jika masalah *underflow / overflow* dapat ditangani tanpa harus melakukan penggeseran banyak piksel, mungkin PSNR yang dihasilkan akan lebih baik.

## DAFTAR PUSTAKA

- Ahmad, T., Holil, M., Wibisono, W. & Muslim, I. R., 2013. *An improved Quad and RDE-based medical data hiding method*. Yogyakarta, Computational Intelligence and Cybernetics (CYBERNETICSCOM), 2013 IEEE International Conference on.
- Alattar, A. M., 2003. *Reversible watermark using difference expansion of triplets*. Gurgaon, Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on.
- Alattar, A. M., 2004. *Reversible watermark using difference expansion of quads*. Montreal, Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on.
- Bansal, C. & Gupta, P., 2014. *A survey on histogram shifting techniques in reversible data hiding*. Gurgaon, Advance Computing Conference (IACC), 2014 IEEE International.
- Bellard, F., 2000. *FFmpeg*. [Online] Available at: <https://ffmpeg.org/> [Accessed 10 January 2016].
- Chanu, Y. J., Tuithung, T. & Manglem Singh, K., 2012. *A short survey on image steganography and steganalysis techniques*. Shillong, Emerging Trends and Applications in Computer Science (NCETACS), 2012 3rd National Conference on.
- Cheddad, A., Condell, J., Curran, K. & Mc Kevitt, P., 2010. Digital image steganography: Survey and analysis of current methods. *Signal processing*, 90(3), pp. 727-752.
- Holil, M. & Ahmad, T., 2015. Secret Data Hiding by Optimizing General Smoothness Difference Expansion-Based Method. *Journal of Theoretical and Applied Information Technology*, 72(2), pp. 155-163.

- Lin, C.-C., Yang, S.-P. & Hsueh, N.-L., 2008. *Lossless data hiding based on difference expansion without a location map*. Sanya, Image and Signal Processing, 2008. CISP'08. Congress on.
- Li, X., Li, B., Yang, B. & Zeng, T., 2013. General framework to histogram-shifting-based reversible data hiding. *Image Processing, IEEE Transactions on*, 22(6), pp. 2181-2191.
- Lou, D.-C., Hu, M.-C. & Liu, J.-L., 2009. Multiple layer data hiding scheme for medical images. *Computer Standards & Interfaces*, 31(2), pp. 329-335.
- Ni, Z., Shi, Y.-Q., Ansari, N. & Su, W., 2006. Reversible data hiding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(3), pp. 354-362.
- Richardson, I. E., 2003. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. s.l.:John Wiley & Sons.
- Signale, 2001. *Lorem ipsum - download text and word document examples*. [Online] Available at: [view-source:http://www.loremipsum.de/download\\_lorem\\_ipsum.html](http://www.loremipsum.de/download_lorem_ipsum.html) [Accessed 17 February 2016].
- Subhedar, M. S. & Mankar, V. H., 2014. Current status and key issues in image steganography: A survey. *Computer Science Review*, Volume 13, pp. 95-113.
- Tew, Y. & Wong, K., 2014. An overview of information hiding in H. 264/AVC compressed video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(2), pp. 305-319.
- Tian, J., 2003. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Techn.*, 13(8), pp. 890-896.
- Tsai, P., Hu, Y.-C. & Yeh, H.-L., 2009. Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Processing*, 89(6), pp. 1129-1143.
- xiph.org, 1994. *Xiph.org :: Derf's Test Media Collection*. [Online] Available at: <https://media.xiph.org/video/derf/> [Accessed 10 January 2016].



Yeh, H.-L., Gue, S.-T., Tsai, P. & Shih, W.-K., 2014. Reversible video data hiding using neighbouring similarity. *Signal Processing, IET*, 8(6), pp. 579-587.

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Penulis lahir di Banyuwangi tanggal 13 Desember 1991, merupakan anak pertama dari tiga bersaudara dari pasangan Budi Setyono dan Sohibah. Penulis menempuh pendidikan formal dari SD Negeri Jember Lor 1 (1998-2004), SMP Negeri 2 Jember (2004-2006), SMA Negeri 2 Jember (2006-2009), dan S1 di P.S. Sistem Informasi Universitas Jember (2009-2014). Selama perkuliahan S2, penulis mengambil bidang minat Komputasi Berbasis Jaringan (KBJ) karena tertarik dengan topik penelitian Keamanan Sistem. Penulis dapat dihubungi melalui email: [next.rasmus@gmail.com](mailto:next.rasmus@gmail.com).