

# Reversible Video Steganografi Menggunakan Metode Improved Reduced Difference Expansion dan Analisis Gray Level Co-occurrence Matrix

Dicky Irwanto, Tohari Ahmad, dan Hudan Studiawan,  
 Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)  
 Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
 e-mail: tohari@if.its.ac.id, hudan@if.its.ac.id, dicky.doktergigi@gmail.com

**Abstrak**—Saat ini penelitian mengenai *steganografi* telah banyak dilakukan, namun metode yang digunakan hanya berfokus pada media citra. Dimana metode ini memiliki kelemahan yaitu media yang digunakan media penyisipan tidak dapat dikembalikan. Tidak hanya itu berkas gambar juga memiliki kelemahan terbatasnya jumlah *pixel* yang tersedia pada suatu citra. Karena kelemahan tersebut perlunya metode yang bersifat *reversible* serta media selain citra untuk meningkatkan fleksibilitas dari proses *steganografi*.

Untuk meningkatkan fleksibilitas proses *steganografi* salah satunya dapat dilakukan menggunakan media video sebagai *cover* serta menggunakan metode IRDE (*Improved Reduced Difference Expansion*) yang dikombinasikan dengan metode analisis GLCM (*Gray Level Co-occurrence Matrix*) dan metode distribusi pesan secara *intraframe* dan *interframe*. Tujuan penggunaan kombinasi metode tersebut adalah untuk meningkatkan jumlah kapasitas pesan yang disisipkan serta meminimalkan jumlah *frame* yang digunakan pada proses *encode*.

Hasil dari uji coba proses *steganografi* dengan kombinasi metode tersebut dapat meningkatkan jumlah *payload* yang dapat disisipkan hingga 59% dibandingkan dengan metode distribusi *intraframe* dan 146% dibandingkan dengan metode distribusi *interframe* pada media yang diujikan. Fitur pemilihan kandidat yang baik pada metode distribusi *intraframe* dan *interframe* (kombinasi) yaitu fitur pemilihan kandidat bedasarkan jumlah *layer* untuk jumlah *payload*  $\leq 1034408$  bit dan pemilihan bedasarkan nilai *quadratic means* perbedaan *pixel* untuk jumlah *payload*  $> 1034408$  bit.

**Kata Kunci**—*Steganografi, Video, Improved Reduced Difference Expansion, Gray Level Co-occurrence Matrix*.

## I. PENDAHULUAN

BERKEMBANGNYA teknologi keamanan informasi memberikan dampak besar dari distribusi pengiriman berkas digital. Ada kalanya berkas digital yang dikirimkan bersifat rahasia dan *confidential*, seperti data catatan militer, kesehatan pasien, dan data keuangan. Sehingga harus dipastikan data tersebut hanya diketahui oleh pihak yang berhak. Salah satu metode yang dapat digunakan untuk menjaga kerahasiaan adalah *steganografi*.

Penelitian pada bidang *steganografi* telah banyak dilakukan khususnya pada media citra, audio maupun video. Beberapa metode yang telah pernah diajukan yaitu *Low Bit Encoding*, *Echo Hiding*, *Spread Spectrum*, *Phase Coding* [1], dan juga

menggunakan transformasi *wavelet* [2]. Namun, dari metode tersebut memiliki kelemahan yaitu berkas *cover* tidak dapat dikembalikan ke bentuk semula (*irreversible*).

Untuk mendapatkan kembali berkas *cover* dan berkas *payload*, perlunya menggunakan metode-metode yang bersifat *reversible*. Salah satu metode yang dapat digunakan adalah *Difference Expansion* (DE) [3], *Reduced Difference Expansion* (RDE), dan *Improved Reduced Difference Expansion* yang diusulkan oleh Hu Yi [4].

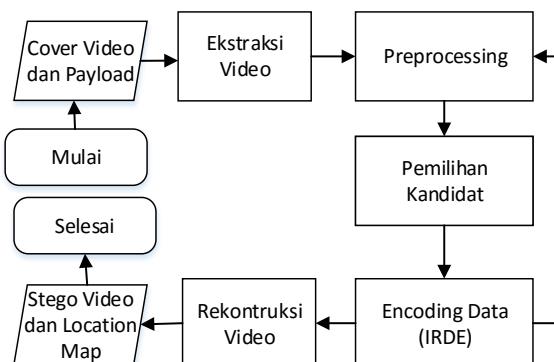
Metode *Improved Reduced Difference Expansion* (IRDE) merupakan metode hasil modifikasi metode *Reduced Difference Expansion* [4]. Dengan menggunakan metode ini sebuah data sisipan (*payload*) dan berkas video (*cover*) dapat dikembalikan seutuhnya tanpa adanya distorsi [4]. Pada *paper* sebelumnya metode *Improved Reduced Difference Expansion* sudah diterapkan pada media citra sebagai *cover*. Penggunaan media citra menggunakan metode IRDE menyebabkan keterbatasan jumlah *pixel* yang akan disisipkan dikarenakan pada metode IRDE terdapat batasan jumlah kandidat yang lebih sempit dibandingkan dengan metode DE atau RDE.

Pada tugas akhir ini kami mencoba menerapkan metode *Improved Reduced Difference Expansion* yang sebelumnya diterapkan pada media citra akan dicoba diterapkan pada media video dengan menambah beberapa modifikasi agar dapat diterapkan. Beberapa modifikasi yang harus dilakukan yaitu perubahan struktur *location map* (LM) dan penambahan metode distribusi pesan dengan menggunakan metode jumlah nilai *quadratic means* perbedaan *pixel*, metode *variance*, jumlah *layer* yang digunakan dan jumlah bit *payload* yang dapat disisipkan. Serta untuk meningkatkan jumlah pesan yang disisipkan distribusi pesan akan dilakukan melalui dua cara, yaitu distribusi pesan secara *single frame* (*intraframe*) dan distribusi pesan antar *frame* (*interframe*).

## II. METODE PENELITIAN

*Steganografi* video dengan metode *Improved Reduced Difference Expansion* secara garis besar terdapat dua tahapan pemrosesan yaitu tahap *encoding* dan *decoding*. Perbedaan batasan tahap yang kami gunakan pada *paper* ini dengan *paper* sebelumnya [4] yaitu terletak pada modifikasi nilai *pixel* dalam keadaan nilai *location map* adalah 0 maka tidak adanya modifikasi nilai *pixel* yang dilakukan sedangkan disaat nilai *location map* adalah 1 maka modifikasi *pixel* dilakukan. Hal

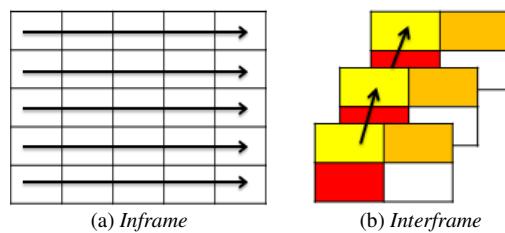
ini dikarenakan reduksi terbaik terletak pada keadaan *location map* adalah 1 [4].



Gambar 1. Skema Proses *Encoding*

Proses pada tahap *encoding* terdiri empat tahapan yaitu: proses *preprocessing*, proses pemilihan kandidat secara *intraframe* atau *interframe*, proses penyisipan *payload* kedalam kandidat menggunakan metode IRDE, dan proses pembuatan *location map* berdasarkan data yang disisipkan. Gambar 1. Untuk proses tahap *decoding* terdiri dari dua tahapan yaitu: proses ekstraksi *location map* dan proses pengembalian *payload* dan *cover* menggunakan metode IRDE. Gambar 3.

Berbeda dengan metode *IRDE* sebelumnya yang menggunakan distribusi secara horizontal dan vertikal Gambar 2, pada metode yang diajukan proses distribusi *payload* dilakukan dua cara yaitu: distribusi *payload* pada satu *frame* (*intraframe*) Gambar 2 (a) dan distribusi *payload* yang melalui *frame* yang berbeda (*interframe*) Gambar 2 (b).



Gambar 2. Distribusi *Payload*

Untuk mendapatkan nilai total distribusi *payload* dapat menggunakan (1).

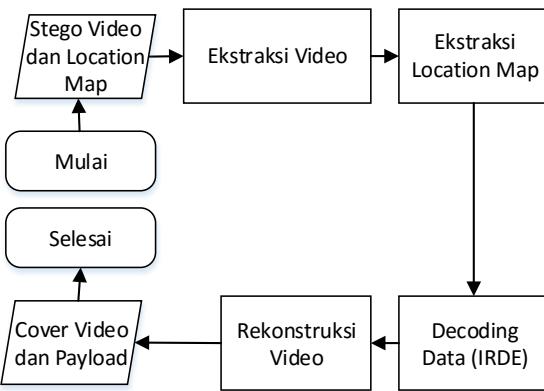
$$s = (\text{width} \times \text{height}) + \text{total frame} \quad (1)$$

Contoh, jika kita menggunakan video dengan panjang 10 detik berukuran 512 x 512 dengan jumlah *frame* perdetik 30 FPS. Maka jumlah distribusi *payload*:

$$s = (512 \times 512) + (30 \times 10) = 262444 \text{ kandidat} \quad (2)$$

Pada (2) terlihat bahwa dengan kombinasi antara *intraframe* dan *interframe* jumlah kandidat lebih banyak dibandingkan menggunakan salah satu metode tersebut. Untuk mendukung kombinasi penggunaan *Intraframe* dan *Interframe* dalam proses *steganografi*, metode yang digunakan dalam proses

*encode* atau *decode* harus mendukung adanya proses *layering*.



Gambar 3. Skema Proses *Decoding*

#### A. *Encode*

1) *Preprocessing*: tahap *preprocessing* merupakan tahap perhitungan nilai fitur pada metode yang diajukan. Perhitungan nilai fitur kandidat pada metode yang diajukan menggunakan empat cara yaitu: perhitungan berdasarkan jumlah *layer* setiap kandidat, perhitungan berdasarkan nilai *variance* GLCM setiap kandidat, perhitungan berdasarkan jumlah data yang dapat disisipkan pada kandidat, dan perhitungan berdasarkan nilai *quadratic means* perbedaan *pixel* kandidat. Perhitungan nilai fitur pertama yaitu berdasarkan jumlah *layer* merupakan proses perhitungan jumlah *layer* yang dimiliki pada setiap kandidat. Jumlah *layer* setiap kandidat akan bertambah jika kandidat tersebut digunakan untuk proses *encoding*. Skema proses penyisipan berdasarkan jumlah *layer* terdapat pada Gambar 3. Untuk perhitungan kedua yaitu berdasarkan nilai *variance* merupakan proses penetapan kandidat terbaik dan perhitungan *variance* setiap kandidatnya. Indikator kandidat terbaik pada proses *encoding* yaitu kandidat memiliki jumlah *payload* yang dapat disisipkan (*Total kandidat*) terbanyak dan memiliki total perbedaan nilai modifikasi *pixel* terendah. Nilai *Total kandidat* dan pada proses *preprocessing* berdasarkan nilai *variance* dapat dihitung melalui (3) hingga (8).

$$h_i = |pixel_i - pixel_{i+1}| \quad (3)$$

$$n = \lfloor \log_2 h_i \rfloor \quad (4)$$

$$flag_i = \begin{cases} 0 & \text{if } 2 \times 2^{n-1} \leq h_i \leq 3 \times 2^{n-1} - 1 \\ 1 & \text{if } 3 \times 2^{n-1} \leq h_i \leq 4 \times 2^{n-1} - 1 \end{cases} \quad (5)$$

$$Total_{\text{kandidat}} = \sum_{i=1}^{n_{\text{kandidat}}} flag_i \quad (6)$$

$$R_{\text{kandidat}} = \sum_{i=1}^{n_{\text{kandidat}}} x_i - (l_i + h'_i) \text{ if } flag_i = 1 \quad (7)$$

$$L_{\text{kandidat}} = \sum_{i=1}^{n_{\text{kandidat}}} y_i - (l_i - h'_i) \text{ if } flag_i = 1 \quad (8)$$

Nilai  $n_{kandidat}$  merupakan nilai jumlah blok pada kandidat *interframe* atau *intraframe*. Sedangkan untuk nilai  $x$  merupakan nilai terbesar dari blok *pixel* dan nilai  $y$  merupakan nilai terendah dari blok *pixel*. Untuk nilai  $l_i$  merupakan nilai rata-rata dari *pixel* dalam blok. Untuk nilai  $h_i'$  merupakan nilai hasil reduksi pada metode IRDE. Untuk  $R_{kandidat}$  dan  $L_{kandidat}$  merupakan nilai total kenaikan atau penurunan *pixel* setelah dimodifikasi. Kemudian setelah mendapatkan nilai  $Total_{kandidat}$ ,  $L_{kandidat}$ , dan  $R_{kandidat}$  tahap selanjutnya yaitu menghitung nilai rata-rata setiap kandidat dan pilih kandidat yang memiliki rata-rata terendah. Pemilihan ini bertujuan untuk memilih kandidat yang terbaik dengan memiliki nilai  $h$  yang kecil. Rata-rata setiap kandidat dapat dihitung dengan (9).

$$\mu_{kandidat} = \frac{R_{kandidat} + L_{kandidat}}{Total_{kandidat}} \quad (9)$$

Tahap selanjutnya yaitu menghitung nilai *variance* setiap kandidat dengan metode GLCM. Metode analisis GLCM yang digunakan adalah *variance*. Untuk kerapatan dan sudut yang digunakan yaitu kerapatan satu *pixel* dan sudut yang digunakan adalah  $0^\circ$ . Pencarian nilai *variance* dapat menggunakan (11).

$$\mu = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N P_{ij} \quad (10)$$

$$Variance = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^2 \quad (11)$$

Nilai  $P_{ij}$  pada (11) merupakan nilai dari matrik GLCM, untuk nilai  $M$  dan  $N$  merupakan nilai ukuran dari matrik GLCM. Perhitungan nilai *variance* setiap kandidat nantinya digunakan untuk membandingkan fitur kandidat yang sebelumnya terdipilih dengan kandidat lain. Tahap selanjutnya yaitu memilih kandidat yang memiliki nilai  $\mu_{kandidat}$  terendah, kandidat tersebut merupakan kandidat terbaik yang nantinya akan dibandingkan nilai *variancanya* dengan kandidat lain.

Perhitungan nilai fitur yang ketiga yaitu perhitungan *preprocessing* berdasarkan nilai *root mean square* (RMS) kenaikan dan penurunan nilai modifikasi *pixel*. Sebelum menghitung nilai RMS perlunya mengetahui nilai kuadarat jumlah kenaikan atau penurunan nilai *pixel* dan jumlah total kapasitas, hal ini dilakukan untuk menghitung kemungkinan jumlah *noise* pada *pixel frame* setelah disisipkan dan jumlah *total payload* yang dapat disisipkan pada setiap kandidat. Persamaan yang digunakan untuk menentukan nilai kuadrat jumlah kenaikan atau penurunan dapat dengan menggunakan (12), (13), dan untuk menentukan jumlah *total payload* yang dapat disisipkan menggunakan (6).

$$R_{kandidat}^2 = \sum_{i=1}^{n_{kandidat}} (x_i - (l_i + h_i'))^2 \text{ if } flag_i = 1 \quad (12)$$

$$L_{kandidat}^2 = \sum_{i=1}^{n_{kandidat}} (y_i - (l_i - h_i'))^2 \text{ if } flag_i = 1 \quad (13)$$

Setelah nilai  $R_{kandidat}^2$ ,  $L_{kandidat}^2$  dan nilai  $Total_{kandidat}$  didapatkan, selanjutnya menghitung nilai RMS pada setiap kandidat. Perhitungan RMS dapat menggunakan (14).

$$Q_{kandidat} = \sqrt{\frac{R_{kandidat}^2 + L_{kandidat}^2}{Total_{kandidat}}} \quad (14)$$

Perhitungan nilai fitur yang keempat yaitu perhitungan *preprocessing* berdasarkan jumlah kapasitas *payload* yang dapat disisipkan. kenaikan dan penurunan nilai modifikasi *pixel*. Perhitungan jumlah kapasitas *payload* dapat dihitung menggunakan (6).

Setelah menghitung fitur yang digunakan tahap selanjutnya yaitu pemilihan kandidat. Pemilihan kandidat merupakan proses pemilihan kandidat yang sesuai dengan batasan-batasan pengujian dan bedasarkan fitur yang digunakan. Batasan-batasan pemilihan setiap kandidat sebagai berikut:

1. Nilai PSNR dari setiap *frame*  $> 35$  dB. 35 dB merupakan nilai tengah Dikarenakan jika nilai PSNR  $< 30$  termasuk *frame* yang bersifat *low quality* dan jika nilai PSNR  $> 40$  termasuk *frame* yang bersifat *high quality* [5].
2. Batasan jumlah *payload* yang disimpan pada setiap kandidatnya adalah 5%.

Jika setiap kandidat memenuhi batasan-batasan tersebut selanjutnya yaitu setiap kandidat akan dipilih sesuai dengan keadaan fiturnya. Detail pemilihan kandidat bedasarkan fitur sebagai berikut:

1. Fitur bedasarkan jumlah *layer*, kandidat yang dipilih merupakan kandidat yang memiliki jumlah *layer* terendah.
2. Fitur bedasarkan nilai *variance*, kandidat yang dipilih merupakan kandidat yang memiliki *difference variance* terhadap kandidat yang dipilih pada tahap *preprocessing*.
3. Fitur bedasarkan nilai *Root Mean Square*, kandidat yang dipilih merupakan kandidat yang memiliki nilai RMS terendah.
4. Fitur bedasarkan jumlah kapasitas *payload*, kandidat yang dipilih merupakan kandidat yang memiliki jumlah kapasitas *payload* terbanyak.

2) *Encode payload*: Tahap *encode payload* merupakan tahap penyisipan *payload* ke dalam *cover*. Metode yang digunakan pada tahap ini adalah metode IRDE karena metode tersebut mendukung adanya *layering frame* [4] serta dapat mengembalikan data *cover* dan *payload* secara utuh tanpa distorsi [4]. Cara kerja metode IRDE adalah membandingkan dua buah *pixel* yang saling bertetangga.

3) *Pembuatan location map*: *location map* merupakan data yang menyimpan lokasi peletakan *payload* ke dalam *cover*. Pembuatan *location map* bertujuan untuk mengembalikan data *payload* dan *cover* ke bentuk semula. Setiap metode *steganografi* biasanya mendefenisikan *location map* berbeda-beda. Pada *Location Map* metode yang kami ajukan, terdapat beberapa perbedaan dengan metode IRDE yang sebelumnya. *location map* pada metode IRDE sebelumnya mendefinisikan

1 untuk keadaan *pixel* disisipkan dan 0 untuk *pixel* yang tidak disisipkan, namun untuk metode yang kami ajukan terdapat bagian yang merepresentasikan *inframe* atau *interframe*, lokasi penyisipan data serta keadaan *pixel* yang disisipkan. Perhitungan tersebut dapat dilihat pada (15) dan (16).

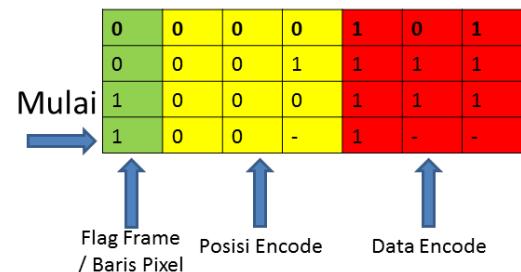
$$m = \begin{cases} \text{length} * \text{width}, \text{if } \text{length} * \text{width} > \text{total frame} \\ \text{total frame}, \text{if } \text{length} * \text{width} \leq \text{total frame} \end{cases} \quad (15)$$

Persamaan (15) merupakan persamaan yang digunakan untuk mendapatkan panjang bit lokasi penyisipan *payload*. Kemudian hasil dari (15) akan dikombinasikan dengan bagian yang merepresentasikan *inframe* atau *interframe* dan panjang keadaan perubahan *pixel*, total panjang *location map* direpresentasikan pada (16).

$$LM_{kandidat} = 1 + \lceil \log_2 m \rceil + \text{Length Data Encode} \quad (16)$$

#### B. Decode

1) *Ekstraksi Stego Video dan Location Map*: Sama seperti tahap *encode*, proses ekstraksi video dilakukan untuk mengambil informasi dan mengekstraksi video menjadi *frame*. Untuk ekstraksi *location map* dapat dilihat pada Gambar 4.



Gambar 4. Skema Proses Decoding

Pada proses ekstraksi *location map* pembacaan data harus dilakukan pada baris yang paling bawah. Hal ini dilakukan karena penggunaan konsep *layering* pada proses *encode*. Setelah tahap pembacaan data, setiap baris *location map* akan diekstraksi menjadi tiga bagian yaitu bagian jenis distribusi, posisi kandidat, dan informasi penyisipan.

2) *Decode Stego Video*: Tahap *decode stego* merupakan tahap pengembalian *payload* dan *cover* ke bentuk semula. Metode yang digunakan pada tahap ini adalah Metode IRDE yang diimplementasikan oleh Hu Yi [4]. Sebelum melakukan *decode* perlunya informasi terkait posisi penyisipan dan teknik penyisipan secara *inframe* atau *interframe*.

### III. HASIL UJI COBA

Data yang digunakan untuk uji coba implementasi sistem *steganografi* ini adalah berkas video dengan ekstensi .avi dengan batasan kompresi yang digunakan pada video adalah HuffyUV dan bertipe *grayscale* dan untuk *payload* yang digunakan berformat .TXT. Detail uji coba ditunjukkan pada Tabel 1 dan Tabel 2.

Data uji video Tabel 1 merupakan data uji standar pada bidang kompresi dan *steganografi* yang dapat diunduh pada halaman <http://media.xiph.org/video/derf/>.

Tabel 1  
Dataset Video

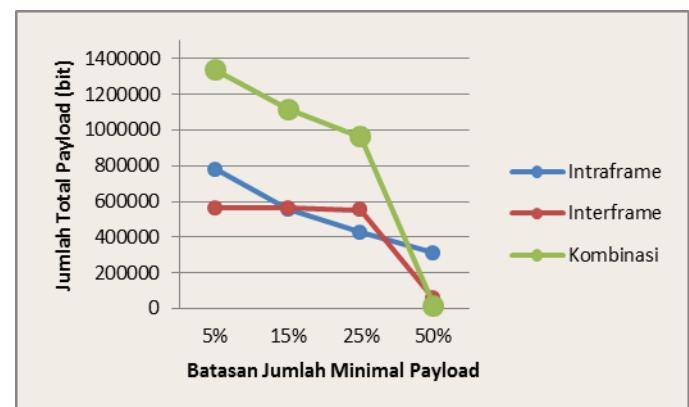
Nama Cover Video (.avi)	Resolusi Frame	Total Frame	Durasi Cover Video (s)
Garden	352 x 240	115	3
Football	176 x 144	130	8
Bus	352 x 288	150	5
Tennis	352 x 240	150	5

Tabel 2  
Dataset Payload

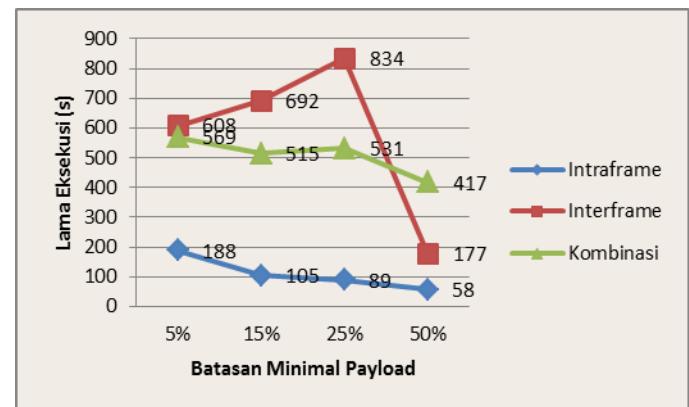
Nama Payload (.txt)	Panjang Data (bit)
asyoulik	1034408
asyoulik3	3103224
asyoulik4	4137632

#### A. Skenario Uji Coba 1

Skenario uji coba 1 merupakan proses pengujian perbandingan nilai batasan kandidat untuk proses pemilihan kandidat. Proses penentapan batasan kandidat dilakukan melalui dua cara yaitu penetapan batasan kandidat bedasarkan jumlah *payload* yang dapat disisipkan pada kandidat dan bedasarkan waktu eksekusi proses uji coba penyisipan. Batasan yang digunakan pada uji coba ini sebesar 5%, 10%, 15%, 25%, dan 50%. Hasil uji coba ditunjukkan pada Gambar 5 dan Gambar 6.



Gambar 5. Jumlah Kapasitas Payload Pada Uji Coba 1



Gambar 6. Lama Waktu Eksekusi Pada Uji Coba 1

Hasil uji coba pertama yakni bedasarkan pada jumlah *payload*. Hasil uji coba pertama terlihat bahwa peningkatan batasan minimal *payload* berpengaruh dalam kapasitas jumlah *payload*. Semakin tingginya batasan yang digunakan maka

semakin sedikitnya *payload* yang dapat disisipkan pada setiap kandidat. Pada Gambar 5 dan Gambar 6 terlihat juga bahwa penentuan batasan minimal *payload* berpengaruh pada lama eksekusi pada program. Hal ini berkaitan dengan semakin tingginya batasan minimal *payload* maka semakin banyaknya kandidat yang dibutuhkan. Dan semakin banyaknya kandidat yang digunakan maka berpengaruh pada lama eksekusi.

#### B. Skenario Uji Coba 2

Skenario uji coba 2 merupakan proses perbandingan kapasitas *encoding* secara *intraframe*, *interframe*, dan kombinasinya. Hasil perbandingan kapasitas *encoding* ditunjukkan pada Tabel 3 hingga Tabel 6.

Tabel 3.

Kapasitas *Payload* Berdasarkan Fitur Quadratic Means

Nama Video	Kapasitas <i>Payload</i> (Fitur Quadratic Means)		
	Distribusi <i>Intraframe</i>	Distribusi <i>Interframe</i>	Distribusi Kombinasi
<i>Football</i>	778360	561941	1235111
<i>Bus</i>	4033592	2685272	4137632
<i>Garden</i>	2675822	1633604	3908237
<i>Tennis</i>	3346781	1685061	4137632

Tabel 4.

Kapasitas *Payload* Berdasarkan Fitur Variance

Nama Video	Kapasitas <i>Payload</i> (Fitur Variance)		
	Distribusi <i>Intraframe</i>	Distribusi <i>Interframe</i>	Distribusi Kombinasi
<i>Football</i>	778481	561941	1235421
<i>Bus</i>	4033450	2685272	4137632
<i>Garden</i>	2676439	1633604	3924446
<i>Tennis</i>	3348283	1685061	4137632

Tabel 5.

Kapasitas *Payload* Fitur Kapasitas *Payload*

Nama Video	Kapasitas <i>Payload</i> (Fitur Jumlah <i>Payload</i> )		
	Distribusi <i>Intraframe</i>	Distribusi <i>Interframe</i>	Distribusi Kombinasi
<i>Football</i>	778360	561941	1234728
<i>Bus</i>	4033450	2685272	4137632
<i>Garden</i>	2676733	1633604	3908237
<i>Tennis</i>	3346781	1685061	4137632

Tabel 6.

Kapasitas *Payload* Fitur Jumlah *Layer*

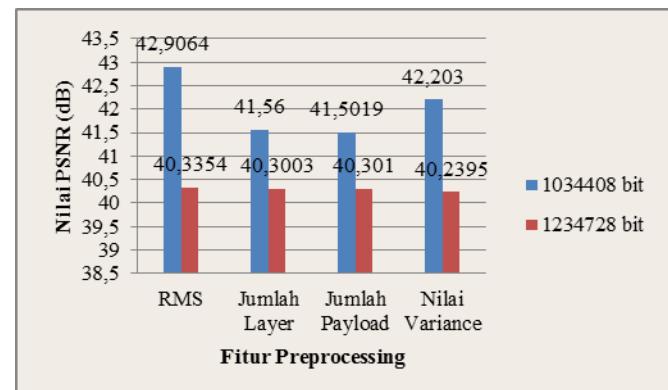
Nama Video	Kapasitas <i>Payload</i> (Fitur <i>Layer</i> )		
	Distribusi <i>Intraframe</i>	Distribusi <i>Interframe</i>	Distribusi Kombinasi
<i>Football</i>	778972	561941	1235736
<i>Bus</i>	4033550	2685272	4137632
<i>Garden</i>	2676428	1633604	3924532
<i>Tennis</i>	3348798	1685061	4137632

Pada Tabel 3 hingga Tabel 6 terlihat bahwa jumlah *payload* yang dapat disisipkan pada metode distribusi kombinasi lebih besar dibandingkan dengan metode distribusi secara *interframe* dan *intraframe*.

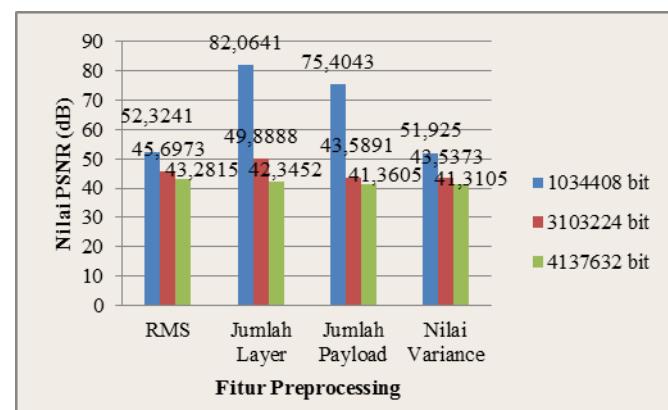
#### C. Skenario Uji Coba 3

Skenario uji coba 3 merupakan proses uji coba perbandingan nilai PSNR dengan menggunakan metode distribusi *payload* secara *intraframe* dan *interframe* (kombinasi). Uji coba akan dilakukan pada empat fitur yaitu fitur pemilihan kandidat bedasarkan nilai *quadratic means*,

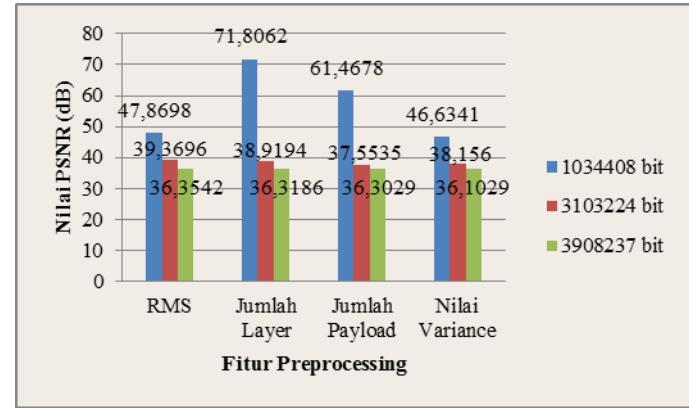
nilai *variance*, kapasitas *payload* yang dapat disisipkan, dan jumlah *layer*. Hasil uji coba skenario 3 ditunjukkan pada Gambar 7 hingga Gambar 10.



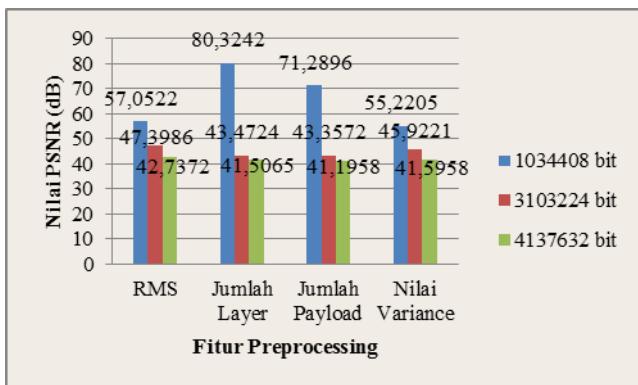
Gambar 7. Hasil PSNR Pada Video Football



Gambar 8. Hasil PSNR Pada Video Bus



Gambar 9. Hasil PSNR Pada Video Garden



Gambar 10. Hasil PSNR Pada Video Tennis

Berdasarkan hasil uji coba skenario 3 fitur yang paling terbaik dalam melakukan *preprocessing* adalah fitur penyisipan menggunakan metode pemilihan kandidat bedasarkan jumlah *layer* setiap kandidat pada jumlah *payload*  $\leq 1034408$  bit dan untuk jumlah *payload*  $> 1034408$  fitur terbaik dalam pemilihan kandidat menggunakan *preprocessing* bedasarkan nilai *Root Mean Square (Quadratic Means)*.

#### IV. KESIMPULAN

Berdasarkan hasil uji coba *steganografi* pada video dengan metode IRDE dan dikombinasikan dengan distribusi secara *interframe* dan *intraframe* dan pemilihan kandidat sesuai fitur didapatkan kesimpulan sebagai berikut:

1. Uji coba proses *encode* menggunakan skema distribusi secara *interframe* dan *intraframe* memperlihatkan peningkatan jumlah kapasitas yang signifikan.
2. Bedasarkan hasil uji coba, fitur pemilihan kandidat yang menghasilkan nilai PSNR dan jumlah *payload* terbaik pada distribusi *payload* secara kombinasi adalah fitur pemilihan kandidat bedasarkan jumlah *layer* pada jumlah *payload* 1034408 bit dengan nilai PSNR maksimal 82,0641 dB yang diujikan pada *cover Bus* dan fitur pemilihan kandidat bedasarkan nilai *quadratic means* perbedaan *pixel* pada jumlah *payload*  $> 1034408$  bit dengan nilai PSNR maksimal 47,3986 dB yang diujikan pada *cover Tennis*.

#### DAFTAR PUSTAKA

- [1] Bilal, I., Roj, M.S., Kumar, R., Mishra, P.K., 11-13 Dec. 2014, "Recent advancement in audio steganography", Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference, 402-405.
- [2] Santosa, R.A., Bao, P., 8-10 Juni 2005, "Audio-to-image wavelet transform based audio steganography", ELMAR, 2005. 47th International Symposium, 209-212.
- [3] Tian, J., Aug. 2003, "Reversible data embedding using a difference expansion", Circuits and Systems for Video Technology, IEEE Transactions, 13, 8: 890-896.
- [4] Yi, H., Wei, S., Hou, J., Aug. 2009, "Improved Reduced Difference Expansion Based Reversible Data Hiding Scheme for Digital Images", Electronic Measurement & Instruments, ICEM'09. 9th International Conference, 4: 315-318.
- [5] Cheddad, A., Condell, J., Curran, K., Kevitt, P.Mc., 2009, "Digital Image Steganography : Survey and Analysis of Current Methods", Signal Processing 90, 2010, 727-752.