



TESIS - TE092099

**PENGURANGAN *FLICKER* PADA VIDEO
ANIMASI KUNO MENGGUNAKAN *WAVELET*
*TRESHOLDING***

**DESITA RIA YUSIAN TB
2211205703**

**Dosen Pembimbing:
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng
Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc**

**PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2014**

THESIS TE 09 2099

**REDUCTION FLICKER IN OLD VIDEO ANIMATION
USING WAVELET TRESHOLDING**

DESITA RIA YUSIAN TB
2211 205 703

SUPERVISOR

Prof. Ir. Mauridhi Hery Purnomo., M.Eng, Ph.D
Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc

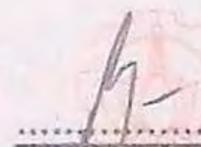
MASTER PROGRAM
DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2014

Tesis disusun untuk memenuhi salah satu syarat untuk memperoleh gelar
Magister Teknik (MT)
di Institut Teknologi Sepuluh Nopember Surabaya

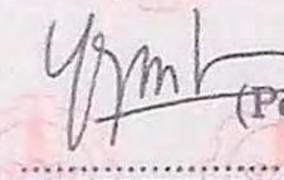
Oleh :
DESITA RIA YUSIAN TB
NRP. 2211205703

Tanggal Ujian : 12 Juni 2014
Periode Wisuda : September 2014

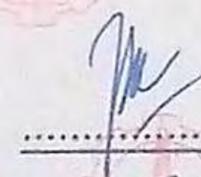
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
NIP. 19580916 198601 1 001


.....
(Pembimbing I)

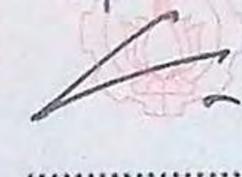
Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc.
NIP. 19540925 197803 1 001


.....
(Pembimbing II)

Dr. I Ketut Eddy Purnama, ST., MT.
NIP. 19690730 199512 1 001

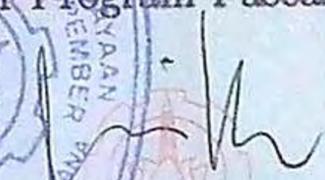

.....
(Penguji I)

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP. 19680601 199512 1 009


.....
(Penguji II)



Mengetahui,
Direktur Program Pascasarjana ITS


Prof. Dr. Ir. Adi Soeprijanto, MT
NIP. 19640405 199002 1 001

PENGURANGAN *FLICKER* PADA VIDEO ANIMASI KUNO MENGGUNAKAN *WAVELET TRESHOLDING*

Nama Mahasiswa : Desita Ria Yusian TB
NRP : 2211205703
Pembimbing I : Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Pembimbing II : Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc.

Abstrak

Sampai saat ini video animasi masih menjadi tontonan yang menarik untuk anak-anak atau orang dewasa, jika film animasi tua memiliki kualitas audio-visual yang modern seperti yang diinginkan penonton maka penonton akan tertarik pada film. Film animasi tua memiliki beberapa jenis kerusakan seperti flicker, noise, bercak dan pergerakan benda-benda divideo yang kurang halus. Pada penelitian ini tahap awal yang dilakukan adalah pembagian video ke dalam 30 frame. Transformasi wavelet dilakukan pada frame dengan melakukan transformasi pada semua baris yang menghasilkan matrik, dimana sisi kiri berisi koefisien low pass down sample dari setiap baris, dan sisi kanan berisi koefisien high pass dengan memilih tipe wavelet Haar dengan dekomposisi 5 level. Flicker pada setiap frame yang telah dikenali sebagai masalah Gaussian noise. Pada nilai koefisien-koefisien wavelet dilakukan proses tresholding pada koefisien detail menggunakan Bayesshrink. Hasil penelitian yang telah dilakukan dinyatakan berdasarkan data hasil yang didapat dengan menghitung PSNR hasil denoising pada citra menggunakan transformasi tipe wavelet Haar memberikan hasil yang baik dan video yang dihasilkan memiliki kualitas gambar yang lebih halus dengan diperolehnya nilai MSE dan PSNR setiap citra inputan hasil denoised pada citra masukan Popaye memiliki nilai MSE = 0.02392 dan PSNR = 45.4977, Steamboat dengan nilai MSE = 0.0173 dan PSNR = 45.9974, dan Felix dengan nilai MSE = 0.01908 dan PSNR = 45.4822.

Kata-kunci: *Video animasi kuno, wavelet, Thresholding, MSE, PSNR*

PENGURANGAN *FLICKER* PADA VIDEO ANIMASI KUNO MENGGUNAKAN WAVELET TRESHOLDING

Nama Mahasiswa : Desita Ria Yusian TB
NRP : 2211205703
Pembimbing I : Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Pembimbing II : Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc.

Abstract

Since first time old video animation was created, children and adult still interest to watch. however, the old video animation still have poor visual quality such as flicker, noise (blotces), and less smooth movement. Various research try to make the old video animation much better in visual technical side with various method. Reduce flicker is one way to make old vidoe animation increase visual side. To reduce flicker in old video animation probelm, in this research used wavelet tresholding method. Wavelet transform are performed on frame by performing transformations on all the lines that generate the matrix. The type of Haar wavelet decomposition level 5 are to decomposition process. Flicker in each frame has been recognized as a Gaussian noise. The results of this research has been done based on video "Popaye", "Steamboat" and "Felix" are obtained MSE and PSNR value. For Popaye input image has $MSE = 0.02392$ and $PSNR = 45.4977$, Steamboat with $MSE = 0.0173$ and $PSNR = 45.9974$, and Felix with $MSE = 0.01908$ and $PSNR = 45.4822$.

Key-words: *Old video animation, wavelet, Thresholding, MSE, PSNR*

KATA PENGANTAR

Segala puji kehadiran Allah SWT karena hanya dengan rahmad-Nya, penyusunan tesis dengan judul "**Pengurangan *Flicker* Pada Video Animasi Kuno Menggunakan *Wavelet Tresholding***" ini dapat berjalan dengan lancar. Tesis ini disusun guna memenuhi persyaratan untuk memperoleh gelar Magister Teknik pada bidang konsentrasi Teknologi Permainan, bidang studi Jaringan Cerdas Multimedia, jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis memahami bahwa penyusunan tesis ini tidak lepas dari bantuan dari banyak pihak, oleh karena itu penulis mengucapkan banyak terimakasih kepada seluruh pihak yang telah membantu proses penyusunan tesis ini, diantaranya:

1. Bapak Prof. Dr.Ir. Mauridhi Hery Purnomo, M.Eng selaku pembimbing I yang telah banyak meluangkan waktu untuk memberi motivasi, masukan dan bimbingan yang sangat berguna kepada penluis.
2. Bapak Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc selaku pembimbing II dan dosen wali penulis selama menempuh pendidikan magister di Teknik Elektro ITS yang telah banyak berjasa dalam memberikan motivasi, bimbingan dan semangat kepada penulis.
3. Bapak Dr. I Ketut Eddy Purnama, ST., MT. selaku penguji I yang telah memberi masukan yang berguna dalam penyusunan tesis ini.
4. Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. selaku penguji II yang telah memberi masukan-masukan yang berguna dalam penyusunan tesis ini.
5. Dosen - dosen bidang jaringan cerdas multimedia, Dr. Supeno Mardi Susiki Nugroho, Bapak Dr. Surya Sumpeno, ST., M.Sc., ST., MuhtadinST, MT, Bapak Christyowidiasmoro, ST., MT. yang telah banyak memberikan ilmu kepada penulis selama menempuh pendidikan di teknik Elektro ITS.
6. Ayahanda dan Ibunda tercinta yang telah selalu mendoakan, memberikan kasih sayang, memberikan motivasi dan dukungan yang

tiada henti kepada penulis dan juga teruntuk adik tersayang Novita Ria Yasita TB yang slalu memberi dukungan dan semangat.

7. Terima kasih kepada Irvannur atas perhatian, dukungan dan motivasi selama penulis menyelesaikan tesis ini.
8. Terima kasih kepada teman - teman seperjuangan JCM angkatan 2011 dan JCM 2012 Mba Rossy, Mba Ratna, Mba Herti, Pak Beni, Mas wisnu, Mas Yudho, Bang isan (gametech), Mas Hery, Pak Afdhal, Bu mira, Pak Hadi yang telah bersedia berbagi ilmu dan pengalamannya.
9. Teman - teman seperjuangan dalam menempuh pendidikan, Muhammad Rizka, Fachri, Munawir, Mas Bagus, Mbak Nanik(Bunik), Andreas Sumendap, Ria Annisa(Buri).
10. dan juga pihak - pihak lain yang telah banyak membantu penulis yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan tesis ini. Saran dan kritik yang membangun dari para pembaca sangat penulis hargai agar dapat dilakukan perbaikan di waktu yang akan datang. Selain itu, penulis mengharapkan akan ada mahasiswa lain yang akan melanjutkan penelitian ini sehingga didapatkan hasil yang lebih bagus lagi.

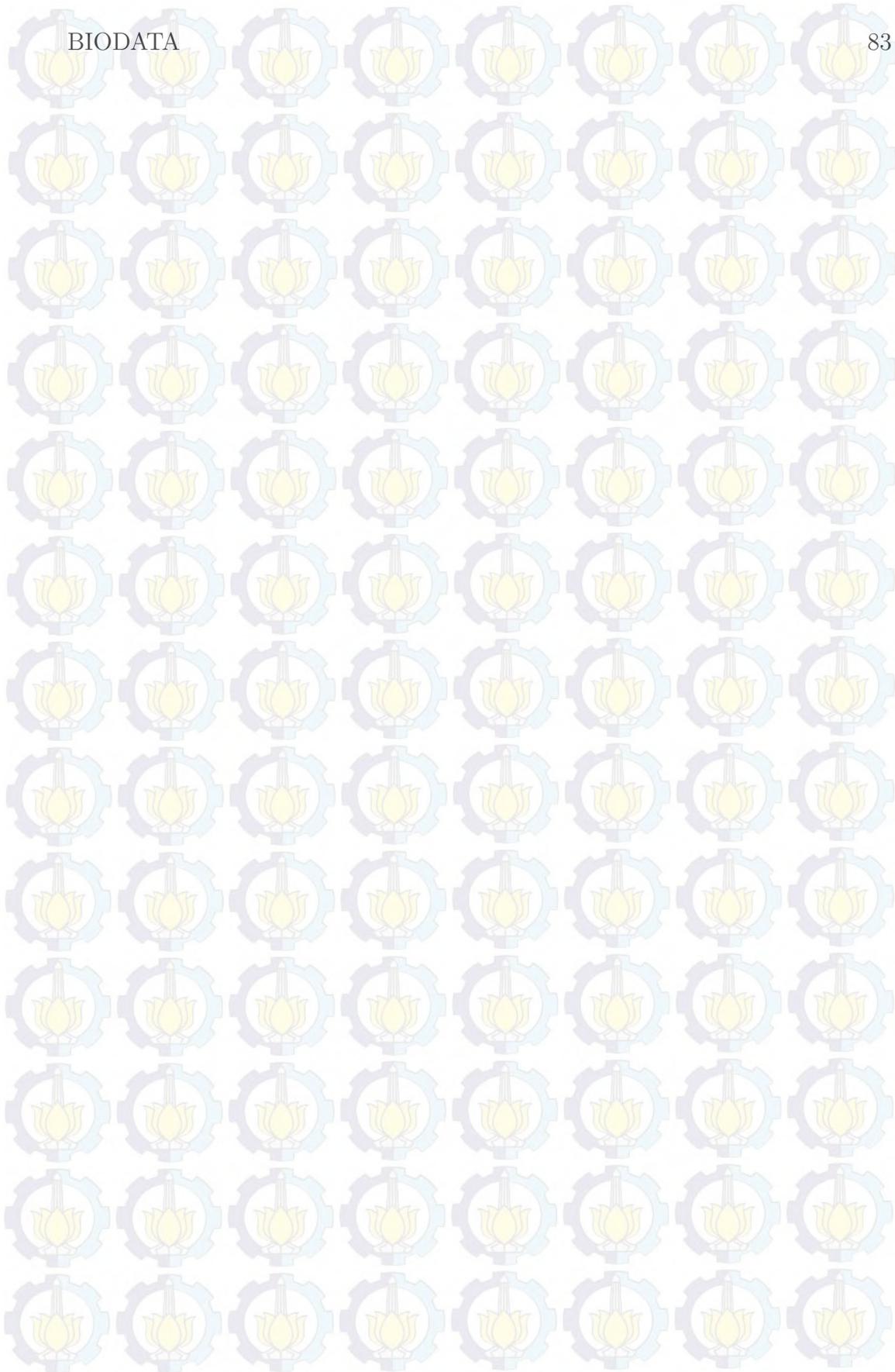
Surabaya 01 Juli 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
SURAT PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Background	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat dan Kontribusi Penelitian	3
BAB II KAJIAN PUSTAKA	5
2.1 Animasi	5
2.2 <i>Gaussian</i> noise	5
2.3 Histogram Citra	7
2.4 Transformasi Wavelet	8
2.5 Wavelet Haar	12
2.6 <i>Discrete Wavelet Transform</i> (DWT)	13
2.7 Teknik Tresholding	16
2.7.1 VisuShrink	18
2.7.2 SureShrink	18
2.7.3 BayesShrink	18
2.8 Teknik Penilaian Kualitas Citra	19
2.8.1 <i>Mean Square Error</i> (MSE)	20
2.8.2 <i>Peak Signal to noise Ratio</i> (PSNR)	20

BAB III	METODA PENELITIAN	21
3.1	Perancangan Sistem	21
3.2	<i>Pre-processing</i>	22
3.2.1	<i>Gaussian noise</i>	24
3.2.2	<i>Discrete Wavelet Transform (DWT)</i>	25
3.2.3	<i>Wavelet Tresholding</i>	27
3.2.4	<i>Invers Wavelet Transform (IDWT)</i>	28
3.2.5	<i>Mean Square Error (MSE)</i>	28
3.2.6	<i>Park Signal to noise Ratio (PSNR)</i>	29
BAB IV	HASIL PENELITIAN DAN PEMBAHASAN	31
4.1	Skenario Uji Coba	32
4.2	Pre-Processing	32
4.3	Membangkitkan <i>Gaussian Noise</i>	37
4.4	Pengujian <i>Discrete Wavelet Transform</i> video input "popeye"	41
4.5	Pengujian Tresholding koefisien detail wavelet level 5 video input "popeye"	41
4.6	Pengujian <i>inverse discrete wavelet transform (IDWT)</i>	45
4.7	Pengujian <i>Discrete Wavelet Transform (DWT)</i> video input "Steamboat"	48
4.8	Pengujian Tresholding koefisien detail wavelet level 5 video input "Steamboat"	49
4.9	Pengujian <i>inverse discrete wavelet transform (IDWT)</i>	52
4.10	Pengujian <i>Discrete Wavelet Transform (DWT)</i> video input "Felix"	55
4.11	Pengujian Tresholding koefisien detail wavelet level 5 video input "Felix"	56
4.12	Pengujian <i>inverse discrete wavelet transform (IDWT)</i>	59
4.13	Hasil pengurangan Flicker Pada Video animasi Kuno	62
BAB V	KESIMPULAN DAN PENELITIAN SELANJUTNYA	65
5.1	Kesimpulan	65
5.2	Penelitian Selanjutnya	65
DAFTAR PUSTAKA		67
LAMPIRAN		69
LAMPIRAN		75
LAMPIRAN		79



BAB I

PENDAHULUAN

1.1 Latar Belakang

Sampai saat ini video animasi masih menjadi tontonan yang menarik untuk anak-anak atau orang dewasa, dan telah menjadi industri besar yang mampu memberikan dampak pada bidang ekonomi dan budaya. Banyak rumah produksi film yang telah membuat program video animasi baru untuk memenuhi kebutuhan penggemarnya (Vetterli S, Chang dan Yu, 2000). Pembuatan program baru tersebut tentu memerlukan biaya yang besar, sedangkan banyak arsip dari video animasi kuno yang masih digemari hanya saja kualitas visual dari film tersebut kurang sempurna seperti intensitas berkedip, *noise*, bercak dan pergerakan objek dalam video kurang halus. Agar video animasi kuno dapat dimanfaatkan kembali maka harus dilakukan reduksi (pengurangan kerusakan pada video animasi kuno).

Pada proses pembuatan video animasi kuno, seorang animator harus membuat sejumlah gambar atau kerangka untuk sebuah gerakan. Video animasi membutuhkan 18 sampai 24 fps (*frame per second*) (Vetterli S dkk., 2000). Animator harus dapat menggambar sekitar 43.200 gambar untuk suatu video animasi kuno yang durasi 30 menit dengan 24 (*frame per second*), hal ini sangat tidak efisien. Seringkali dalam proses pembuatan gambar-gambar tersebut terjadi ketidaksesuaian antara gambar pertama dengan gambar kedua karena gambar dikerjakan secara manual atau digambar dengan menggunakan tangan. Dibutuhkan suatu ketelitian dan keuletan dalam membuat gambar-gambar tersebut agar suatu video animasi menampilkan kualitas gambar yang baik.

Penelitian sebelumnya telah dilakukan sebelumnya (Vetterli S dkk., 2000) melakukan penelitian untuk mengurangi flicker pada video animasi kuno dengan menggunakan dua teknik perbaikan yaitu dengan menggunakan transformasi *affine* parameter transformasi yang ditemukan pada dasar model waktu variasi mean dan varians) dan transformasi intensitas nonlinier untuk mencapai bahwa intensitas gambar berubah secara statistik mirip dengan frame disekitarnya. Pada penelitian ini hasil visual yang diperoleh tidak begitu baik dan setelah dilakukan pencocokan histogram terlihat bahwa

hasil histogram menunjukkan banyak di antara frame awal dengan frame selanjutnya tidak memiliki kecocokan dan masih terlihat kecacatan. Khriji melakukan penelitian serupa untuk memperbaiki gambar pada video kuno dengan menggunakan sebuah interpolasi non-linier, atau menggunakan filter fungsi rasional yang diterapkan pada pemulihan frame urutan gambar dari film-film lama digital (Khriji dan Gabbouj, 2000). Sampel yang akan diinterpolasi adalah karena cacat diam dan acak. Khriji melakukan pengujian pada beberapa frame dari sebuah film lama telah digunakan untuk menguji kinerja operator non-linier untuk rekonstruksi kecacatan. Hasil yang diperoleh menggunakan pelebaran morfologi spasial. Unsur penataan tergantung pada jenis cacat: biasanya itu adalah persegi, namun dalam beberapa kasus tertentu bisa berbeda. Metode ini sangat cepat, tapi tekstur yang dihasilkan halus dan kualitas gambar diproses secara signifikan lebih baik daripada yang asli. Selanjutnya, teknik yang diusulkan melakukan sedikit lebih baik dari yang berdasarkan interpolasi morfologi. Waktu proses cepat karena operator hanya diterapkan di bagian yang hilang. Selain itu, interpolator rasional memerlukan operasi hanya sedikit. Salah satu kelemahannya yaitu mereka tidak dapat merekonstruksi kecacatan yang terletak dekat dengan perbatasan citra dengan sempurna dan masih tetap bergerigi.

Penelitian lainya dilakukan Andra Gunara melakukan penelitian dengan menganalisa perbandingan reduksi *noise* pada citra antara *discrete wavelet transform* (DWT) dengan *dual-tree complex wavelet transform* (DTCWT) (Andra, Iwut dan Jangkung, 2007). Hasil dari penelitian di dapatkan bahwa level dekomposisi maksimum dan penggunaan filter yang berbeda tidak menjanjikan perbaikan optimal pada citra ternoise. Pada SNR 18, terjadi perubahan hasil denoising citra dimana hasil yang dihasilkan oleh DTCWT memberikan hasil yang lebih baik daripada DWT. Semakin besar ukuran citra maka akan semakin baik hasil *denoising*. Peneliti sebelumnya Yair Kuszpet melakukan penelitian mengenai mengurangi flicker menggunakan *post-processing* di H.264/AVC (Kuszpet dan dkk, n.d.). Pada penelitian ini menggunakan sebuah skema *post-processing* untuk berhubungan dengan artefak *flicker*. Hasil dari penelitian ini menunjukkan bahwa metode yang diusulkan secara signifikan mengurangi diskontinuitas dengan penurunan hampir diabaikan dalam PSNR.

Pada penelitian ini untuk memperbaiki ketidaksempurnaan dalam video animasi kuno yang disebabkan karena adanya *flicker* maka dilakukan penguurangan *noise* yaitu *flicker* yang juga disebut proses denoising. Proses

denoising citra dilakukan dengan proses filtering, analisa wavelet dan analisa penilaian kualitas citra menggunakan teknik *peak signal to noise ratio* (PSNR). Dalam penelitian ini citra yang ditransformasikan yaitu citra dalam domain wavelet, citra akan dibagi berdasarkan frekuensi rendah dan tinggi, sehingga menghasilkan arah yang baik. Pada tahap thresholding koefisien wavelet digunakan untuk membedakan informasi dari *noise* dalam domain wavelet yang menyebar keseluruh koefisien dan sebagian besar informasi citra terkonsentrasi dalam koefisien terbesar. Setelah di denoising dilakukan proses rekonstruksi berupa citra yang telah di tresholding, digabungkan semua subband dan hasil setelah dilakukannya rekonstruksi berupa citra yang telah terdenoising. Tahap akhir dalam penelitian ini adalah melakukan analisa dari perhitungan kualitas citra setelah di denoising menggunakan PSNR.

1.2 Rumusan Masalah

Menjadikan kualitas video animasi kuno yang bebas dari *flicker* dan memiliki kualitas gambar yang lebih halus, maka harus dilakukan perbaikan kualitas pada video dengan mengurangi *flicker* yang disebabkan pada proses pengambilan gambar per framenya yang dilakukan secara manual dan gambar yang dibuat tidak mencukupi standar untuk sebuah gerakan animasi yaitu 18-24 *frame per second* dengan pengenalan terhadap setiap frame yang memiliki nilai intensitas *flicker* dari setiap 2 frame berturut-turut dan mengurangi *flicker* yang telah dikenali menggunakan wavelet tresholding.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk memperbaiki kualitas video animasi kuno dengan melakukan pengurangan *flicker* agar dapat membantu dalam penelitian selanjutnya seperti *motion* estimasi dan *motion vector smoothing* untuk data teliti video animasi kuno.

1.4 Manfaat dan Kontribusi Penelitian

Adapun manfaat dari penelitian ini adalah dengan adanya perbaikan video animasi kuno dari kerusakan yang diakibatkan karena *flicker* maka akan memperoleh kualitas hasil gambar video yang lebih halus dan bagus dan dapat membantu dalam penelitian selanjutnya seperti *motion* estimasi dan *motion vector smoothing* untuk data teliti video animasi kuno.

BAB II

KAJIAN PUSTAKA

2.1 Animasi

Animasi adalah film yang berasal dari pengolahan gambar tangan sehingga menjadi gambar yang bergerak. Bentuk animasi tertua diperkirakan wayang kulit. Karena wayang memenuhi semua elemen animasi seperti layar, gambar bergerak, dialog dan ilustrasi musik. Animasi mulai berkembang sekitar abad ke-18 di Amerika (Vetterli S dkk., 2000). Pada saat itu teknik *stop motion animation* banyak disenangi. Teknik ini menggunakan serangkaian gambar diam atau frame yang dirangkai menjadi satu dan menimbulkan kesan seolah-olah gambar tersebut bergerak. Teknik ini sangat sulit, membutuhkan waktu, juga biaya yang banyak. Karena untuk menciptakan animasi selama satu detik, kita membutuhkan sebanyak 12-24 frame gambar diam. Bayangkan jika film animasi itu berdurasi satu jam bahkan lebih.

Pada proses pembuatan video animasi kuno, seorang animator harus membuat sejumlah gambar atau kerangka untuk sebuah gerakan. Video animasi membutuhkan 18 sampai 24 fps *frame per second* (Vetterli S dkk., 2000). Animator harus dapat menggambar sekitar 43.200 gambar untuk suatu video animasi kuno yang durasi 30 menit dengan 24 gambar per detik, hal ini sangat tidak efisien. Seringkali dalam proses pembuatan gambar-gambar tersebut terjadi ketidaksesuaian antara gambar pertama dengan gambar kedua karena gambar dikerjakan secara manual atau digambar dengan menggunakan tangan. Dibutuhkan suatu ketelitian dan keuletan dalam membuat gambar-gambar tersebut agar suatu video animasi menampilkan kualitas gambar yang baik. *Flicker* disebut sebagai “kedipan” yaitu permasalahan yang disebabkan adanya keadaan terang dan redupnya Pendar Posfor pada teknologi monitor lama sehingga total waktu untuk vertical retrace dan *scanning* cukup besar yang menyebabkan terjadinya efek *flicker*.

2.2 Gaussian noise

Berdasarkan bentuk dan karakteristiknya, *noise* pada citra dibedakan menjadi beberapa macam seperti Gaussian *noise* yang merupakan model *noise* yg mengikuti distribusi normal standar dengan rata-rata nol dan

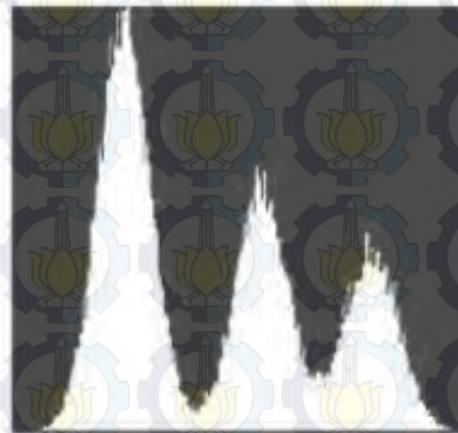
standar deviasi 1. Efek dari *noise* ini adalah munculnya titik-titik berwarna yg jumlahnya sama dengan presentase *noise*. Model dari Gaussian *noise* (Vetterli S dkk., 2000) dapat dilihat dalam persamaan 2.1 dan hasil dari pembangkitan *Gaussian noise* dapat dilihat pada Gambar 2.1 Gambar 2.2.

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}} \quad (2.1)$$

dimana z merepresentasikan intensitas, \bar{z} adalah nilai rata-rata z dan s adalah standar deviasi.



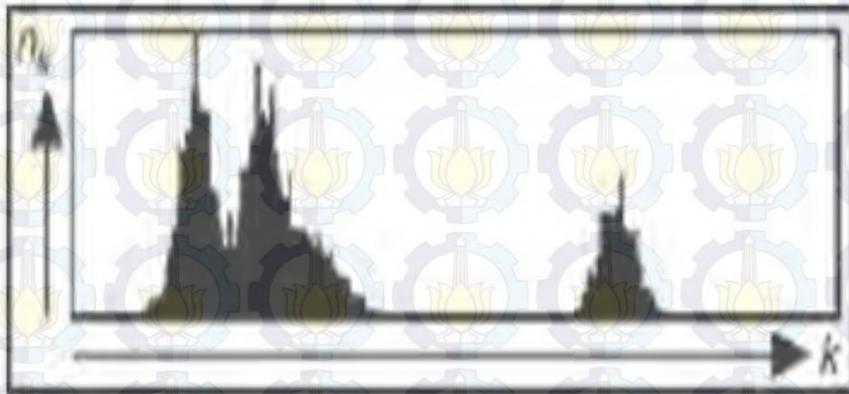
Gambar 2.1: Contoh Representasi *Gaussian noise*



Gambar 2.2: Contoh Histogram Representasi *Gaussian noise*

2.3 Histogram Citra

Informasi penting mengenai isi citra digital dapat diketahui dengan membuat histogram citra. Histogram citra adalah grafik yang menggambarkan penyebaran kuantitatif nilai derajat keabuan (gray level) pixel didalam (atau bagian tertentu) citra. Misalkan citra digital memiliki L derajat keabuan, yaitu dari nilai 0 sampai $L - 1$ (misalnya pada citra dengan kuantisasi derajat keabuan 8-bit, nilai derajat keabuan dari 0 sampai 255). Gambar 2.3 memperlihatkan contoh sebuah histogram citra, yang dalam hal ini k menyatakan derajat keabuan dan n_k menyatakan jumlah pixel yang memiliki nilai keabuan k (Munir, 2006).



Gambar 2.3: Contoh Histogram Citra (Sumber:(Munir, 2006))

Seringkali pada beberapa operasi pengolahan citra jumlah pixel yang memiliki derajat keabuan k dinormalkan terhadap jumlah seluruh pixel di dalam citra, Seringkali pada beberapa operasi pengolahan citra jumlah pixel yang memiliki derajat keabuan k dinormalkan terhadap jumlah seluruh *pixel* di dalam citra yang dijelaskan pada persamaan 2.2

$$h_k = \frac{n_k}{n}, k = 0, 1, \dots, L - 1 \quad (2.2)$$

Sehingga $0 \leq h_i \leq 1$. Persamaan 2.2 menyatakan frekuensi kemunculan nisbi (*relative*) dari derajat keabuan pada citra tersebut. Khusus untuk citra berwarna, histogramnya dibuat untuk setiap kanal RGB (merah, hijau, dan biru). Histogram citra menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar. Puncak histogram menunjukkan intensitas pixel yang menonjol. Lebar dari puncak menun-

juukkan rentang kontras dari gambar. Citra yang mempunyai kontras terlalu terang (*overexposed*) atau terlalu gelap (*underexposed*) memiliki histogram yang sempit.

Terdapat empat proses histogram, yaitu:

1. Apabila gambar gelap maka histogram cenderung ke sebelah kiri.
2. Apabila gambar terang maka histogram cenderung ke sebelah kanan.
3. Apabila gambar *low contrast* maka histogram mengumpul di suatu tempat.
4. Apabila gambar *high contrast* maka histogram merata di semua tempat.

Histogramnya terlihat hanya menggunakan setengah dari daerah derajat keabuan. Citra yang baik memiliki histogram yang mengisi daerah derajat keabuan secara penuh dengan distribusi yang merata pada setiap derajat keabuan *pixel*.

2.4 Transformasi Wavelet

Teori wavelet adalah suatu konsep yang relatif baru dikembangkan. Kata wavelet sendiri diberikan oleh Jean Morlet dan Alex Grossmann diawal tahun 1980-an, dan berasal dari bahasa Prancis, *ondelette* yang berarti gelombang kecil. Wavelet adalah fungsi matematika yang memotong-motong data menjadi kumpulan-kumpulan frekuensi yang berbeda, sehingga masing masing komponen tersebut dapat dipelajari dengan menggunakan skala resolusi yang berbeda. Wavelet merupakan sebuah fungsi variabel real t , diberi notasi ψ_t dalam ruang fungsi $L^2(R)$. Fungsi ini dihasilkan oleh parameter dilatasi dan translasi yang dinyatakan dalam persamaan 2.3 dan persamaan 2.4 (Saraswati, 2010)

$$\psi_{a,b}^t = a^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) \quad (2.3)$$

$$\psi_{j,k}^t = a^{\frac{j}{2}} \psi(2^j t - k); j, k \in Z \quad (2.4)$$

Dimana:

1. a = parameter dilatasi
2. b = parameter translasi

3. $\mathcal{R} =$ mengkondisikan nilai a dan b bernilai real

4. $2^j =$ parameter dilatasi

5. $k =$ parameter waktu atau lokasi ruang

6. $Z =$ mengkondisikan nilai j dan k bernilai integer

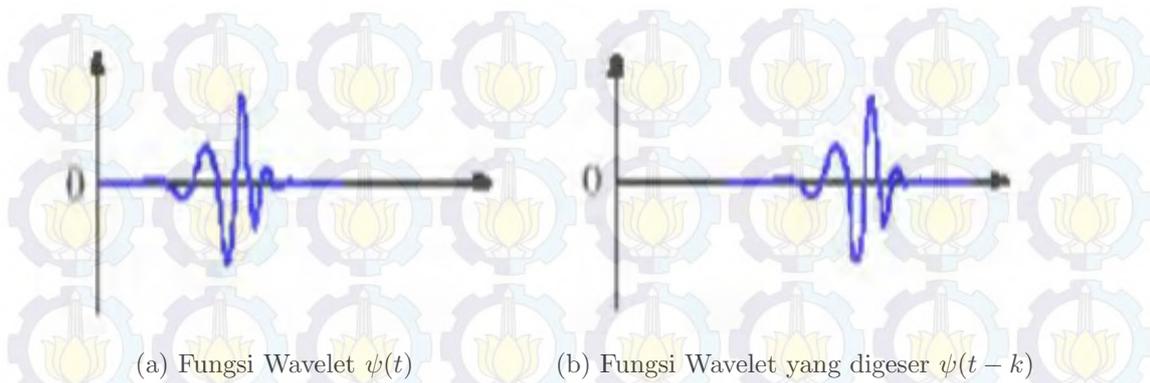
Transformasi wavelet merupakan uraian dari suatu sinyal atau citra menggunakan fungsi wavelet. Salah satu alasan mengapa transformasi wavelet menjadi begitu penting dalam berbagai bidang adalah karena sifat-sifat berikut:

1. Waktu kompleksitasnya bersifat linear. Transformasi wavelet dapat dilakukan dengan sempurna dengan waktu yang bersifat linear. Koefisien-koefisien wavelet yang terpilih bersifat jarang. Secara praktis, koefisien-koefisien wavelet kebanyakan bernilai kecil atau nol. Kondisi ini sangat memberikan keuntungan terutama dalam bidang kompresi atau pemampatan data.

2. Wavelet dapat beradaptasi pada berbagai jenis fungsi, seperti fungsi yang tidak kontinu, dan fungsi yang didefinisikan pada domain yang dibatasi (Saraswati, 2010).

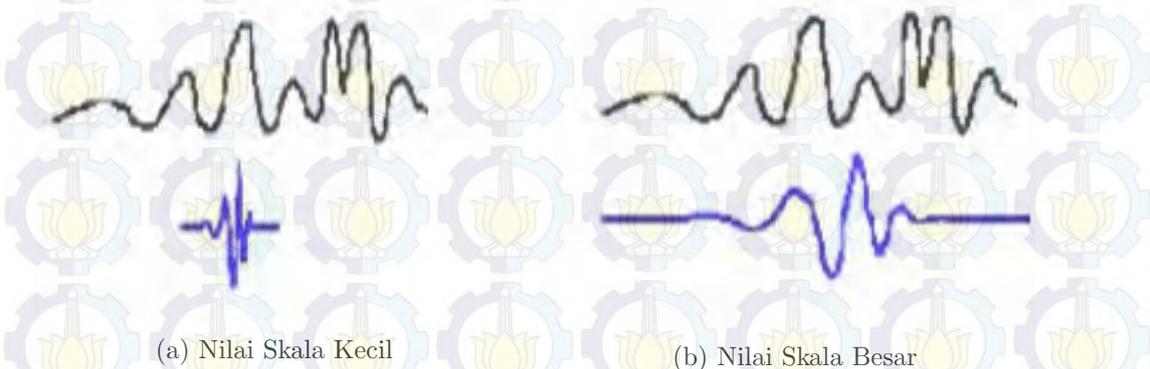
3. Sebuah wavelet yang energinya terkonsentrasi pada suatu selang waktu untuk memberikan kemampuan analisis transien, tidak stasioner, atau fenomena berubah terhadap waktu (*time varying*). Karakteristik dari pada wavelet antara lain adalah beresolusi singkat, translasi (pergeseran) dan dilatasi (skala). Secara sederhana, translasi pada wavelet bermaksud untuk menggeser permulaan dari sebuah wavelet. Secara matematis, pergeseran sebuah fungsi $f(t)$ dengan k yang direpresentasikan menjadi $f(t - k)$ (Mahjud, 2010) ditunjukkan pada Gambar 2.4 dan skala dalam sebuah wavelet berarti pelebaran atau penyempitan wavelet. Seperti yang terlihat pada Gambar 2.5.

Sebuah faktor skala dapat dinyatakan sebagai α . Apabila α diperkecil maka wavelet akan menyempit dan terlihat gambaran mendetail namun tidak menyeluruh, kebalikannya apabila α diperbesar maka wavelet akan melebar dan terlihat gambaran kasar, global namun menyeluruh. Dengan menggunakan wavelet pada skala resolusi yang berbeda, akan diperoleh gambar keduanya,



Gambar 2.4: Hasil dari translasi dan dilatasi wavelet: (a) Fungsi Wavelet $\psi(t)$
 (b) Fungsi Wavelet yang digeser $\psi(t - k)$

yaitu gambaran mendetail dan menyeluruh. Selain itu, terdapat korespondensi antara skala pada wavelet dengan frekuensi yang dianalisa oleh wavelet. Nilai skala yang kecil berkorespondensi dengan frekuensi tinggi (*high frequency*) sedangkan nilai skala yang besar berkorespondensi dengan frekuensi rendah.



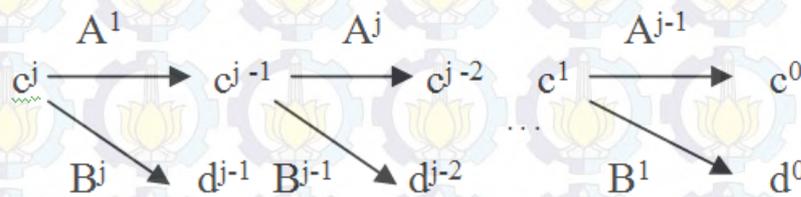
Gambar 2.5: Bentuk translasi dari fungsi wavelet: (a) Nilai Skala Kecil (b) Nilai Skala Besar

Tahap pertama analisis wavelet adalah menentukan tipe wavelet, yang disebut dengan *mother wavelet*, yang akan digunakan. Karena *mother wavelet* menghasilkan semua fungsi wavelet yang digunakan dalam transformasi melalui translasi dan penskalaan, maka *mother wavelet* juga akan menentukan karakteristik dari transformasi wavelet yang dihasilkan. Oleh karena itu, perlu pencatatan secara teliti terhadap penerapan wavelet dan pemilihan yang tepat terhadap *mother wavelet* harus dilakukan agar dapat menggunakan transformasi wavelet secara efisien. Fungsi-fungsi yang termasuk di dalam keluarga wavelet dipaparkan pada Gambar 2.6.

Keluarga Wavelet	Wavelet
Daubechies	'db1' or 'haar', 'db2', ..., 'db10', 'db45'
Coiflet1	'coif1', ..., 'coif5'
Symlet	'sym2', ..., 'sym8', ..., 'sym45'
Discrete Meyer	'dmey'
Biorthogonal	'bior1.1', 'bior1.3', 'bior1.5', 'bior2.2', 'bior2.4', 'bior2.6', 'bior2.8', 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7', 'bior3.9', 'bior4.4', 'bior5.5', 'bior6.8',

Gambar 2.6: Beberapa fungsi wavelet yang umum digunakan

Transformasi wavelet memiliki dua seri dalam pengembangannya yaitu *Continous Wavelet Transform (CWT)* dan *Discrete Wavelet Transform (DWT)*. *Mother wavelet* merupakan fungsi dasar yang digunakan dalam transformasi. Semua fungsi yang digunakan dalam transformasi CWT dan DWT diturunkan dari *mother wavelet* melalui translasi atau pergeseran dan penskalaan atau kompresi. Proses transformasi wavelet dilakukan dengan mengkonvolusi sinyal dengan data tapis atau dengan proses perata-rataan dan pengurangan secara berulang, yang sering disebut dengan metode filter bank Gambar 2.7 (Sediyono, 2009).



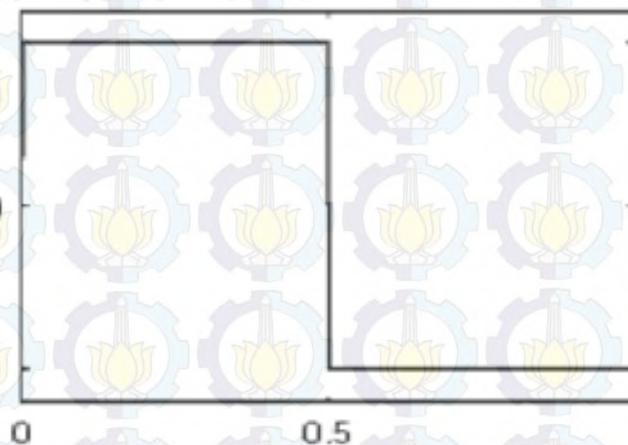
Gambar 2.7: Transformasi Wavelet dengan Metode Filter Bank

Transformasi wavelet memiliki prinsip dasar membagi data menjadi komponen-komponen frekuensi yang berbeda. Proses ini dinamakan dekom-

posisi. Proses transformasi sinyal menjadi koefisien-koefisien wavelet diperoleh dengan filtering menggunakan highpass filter dan *low pass* filter yang kemudian di downsampling. Operasi *downsampling* adalah operasi untuk mengurangi sampel sinyal menjadi setengahnya dan tetap mempertahankan periodanya. *Downsampling* dapat dilakukan dengan cara menghilangkan titik-titik yang bernomor ganjil dan hanya akan mengambil sampel-sampel yang berindeks genap. Sedangkan untuk menggabungkan koefisien-koefisien wavelet dinamakan rekonstruksi. Proses ini merupakan kebalikan dari proses dekomposisi (Pramiswari, 2012).

2.5 Wavelet Haar

Fungsi Haar telah dipakai dari tahun 1910 yang diperkenalkan oleh Alfred Haar dari Hungaria. *Haar wavelet* adalah metode *haar* yang sederhana dari seluruh mother wavelets yang ada. Wavelet Haar masuk dalam kategori *ortogonal* dan terdukung secara kompak, karena *wavelet Haar* sama dengan *wavelet db1* (*Daubechies orde 1*). Metode ini telah dipakai, dan sangat simple, komputasi cepat dan attractive ((A dan Ray, 2014)). *Haar wavelet* didefinisikan dalam interval $t \in [0, 1]$, tetapi pada kasus umumnya wavelet haar dinotasikan dengan $t \in [A, B]$ kemudian dibagi kedalam "m" *equal subintervals* untuk notasi $\Delta t = (B - A)/m$. Dalam kasus ini, fungsi *orthogonal Haar* didefinisikan kedalam interval $[A, B]$. Panjang tapis *wavelet Haar* adalah 2 (Prihartono, 2011). Fungsi penskala pada *Wavelet Haar* ditunjukkan pada ??.



Gambar 2.8: Fungsi Penskala wavelet Haar (Sumber:(Dwi Prihartono, 2011))

$$h_0(t) = \begin{cases} 1, t \in [A, B] \\ 0, elsewhere \end{cases} \quad (2.5)$$

$$h_i(t) = \begin{cases} 1, \zeta_1(i) \leq t < \zeta_2(i) \\ -1, \zeta_2(i) \leq t < \zeta_3(i) \\ 0, otherwise \end{cases} \quad (2.6)$$

dimana :

$$\zeta_1(i) = A + \left(\frac{k-1}{2^j}\right)(B-A) = A + \left(\frac{k-1}{2^j}\right)m\Delta t \quad (2.7)$$

$$\zeta_1(i) = A + \left(\frac{k-\frac{1}{2}}{2^j}\right)(B-A) = A + \left(\frac{k-\frac{1}{2}}{2^j}\right)m\Delta t, \quad (2.8)$$

$$\zeta_1(i) = A + \left(\frac{k}{2^j}\right)(B-A) = A + \left(\frac{k}{2^j}\right)m\Delta t, \quad (2.9)$$

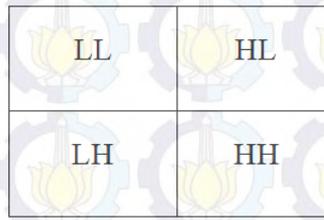
dengan :

$$I = 1, 2, 3, m, m, \dots = 2^j$$

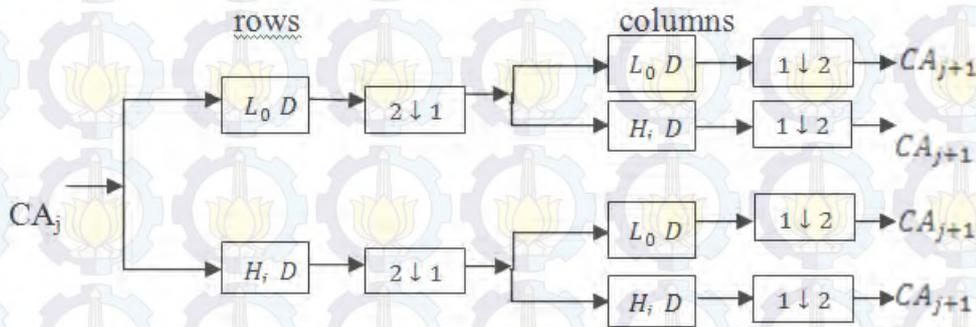
J = integer positif yang merupakan maximum level
j and k = integer dekomposisi.

2.6 Discrete Wavelet Transform (DWT)

Dasar dari DWT dimulai pada tahun 1976 dimana teknik untuk dekomposisi sinyal waktu diskrit ditemukan. Di dalam CWT, sinyal dianalisis menggunakan seperangkat fungsi dasar yang saling berhubungan dengan penskalaan dan transisi sederhana. Sedangkan di dalam DWT, penggambaran sebuah skala waktu sinyal digital didapatkan dengan menggunakan teknik filterisasi digital. Secara garis besar proses dalam teknik ini adalah dengan melewati sinyal yang akan dianalisis pada filter dengan frekuensi dan skala yang berbeda. Filterisasi sendiri merupakan sebuah fungsi yang digunakan dalam pemrosesan sinyal. Wavelet dapat direalisasikan menggunakan iterasi filter dengan penskalaan. Resolusi dari sinyal, yang merupakan rata-rata dari jumlah detail informasi dalam sinyal, ditentukan melalui filterisasi ini dan skalanya didapatkan dengan *up sampling* dan *down sampling* (*sub sampling*).



Gambar 2.9: Dekomposisi wavelet 1 level



Gambar 2.10: Dekomposisi wavelet 2 level

Dimana:

$2 \downarrow 1$: merupakan *downsample* kolom

$1 \downarrow 2$: merupakan *downsample* baris

Seperti yang terlihat pada Gambar diatas, jika suatu citra dilakukan proses transformasi wavelet diskrit dua dimensi dengan level dekomposisi satu, maka akan menghasilkan empat buah subband, yaitu:

1. Koefisien Aproksimasi (CA_{j+1}) atau disebut juga subband LL
2. Koefisien Detil Horisontal ($CD(h)_{j+1}$) atau disebut juga subband HL
3. Koefisien Detil Vertical ($CD(v)_{j+1}$) atau disebut juga subband LH
4. Koefisien Detil Diagonal ($CD(d)_{j+1}$) atau disebut juga subband HH

Dengan Level Dekomposisi 1 Subband hasil dari dekomposisi dapat didekomposisi lagi karena level dekomposisi wavelet bernilai dari 1 sampai n atau disebut juga transformasi wavelet multilevel. Jika dilakukan dekomposisi lagi, maka subband LL yang akan didekomposisi karena subband LL berisi sebagian besar dari informasi citra. Jika dilakukan dekomposisi dengan level

dekomposisi dua maka *subband LL* akan menghasilkan empat buah subband baru, yaitu subband *LL2* (Koefisien Aproksimasi2), *HL2* (Koefisien Detil Horizontal 2), *LH2* (Koefisien Detil Vertikal 2), dan *HH2* (Koefisien Detil Diagonal 2). Dan begitu juga seterusnya jika dilakukan dekomposisi lagi.

LL2	HL2	HL1
LH2	HH2	
LH1		HH1

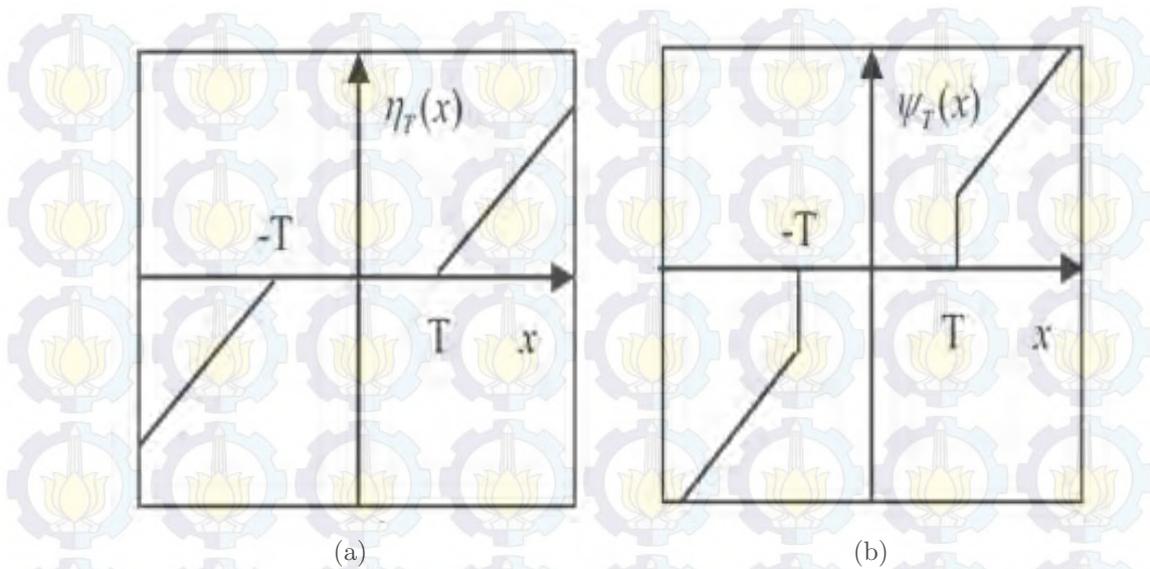
Gambar 2.11: Dekomposisi wavelet 2 level

Dengan menggunakan koefisien DWT ini maka dapat dilakukan proses *Inverse Discrete Wavelet Transform* (IDWT) untuk merekonstruksi menjadi sinyal asal. DWT menganalisis sinyal pada frekuensi berbeda dengan resolusi yang berbeda melalui dekomposisi sinyal sehingga menjadi detil informasi dan taksiran kasar. DWT bekerja pada dua kumpulan fungsi yang disebut fungsi penskalaan dan fungsi wavelet yang masing-masing berhubungan dengan *lowpass* filter dan *highpass* filter. Seperti yang telah dijelaskan sebelumnya dekomposisi ini didasarkan pada aturan *Nyquist* yang salah satunya mengatakan bahwa frekuensi komponen *sample* harus kurang atau sama dengan setengah dari frekuensi *sampling*. Jadi diambil frekuensi *sample* $\frac{\pi}{2}$ dari frekuensi *sampling* π dalam *subsample* oleh 2 pada dekomposisi wavelet.

Pada dasarnya, rekonstruksi adalah proses kebalikan dari dekomposisi. Pendekatan dan koefisien di setiap tingkatan melewati *lowpass* filter dan *highpass* sintesis. Proses ini terus berlanjut sampai jumlah yang sama tingkat seperti dalam proses dekomposisi untuk mendapatkan sinyal asli.

2.7 Teknik Tresholding

Pada semua filter wavelet menggunakan wavelet operasi tresholding untuk denoising yang dilakukan pada komponen citra dengan frekuensi tinggi dan muncul dalam koefisien wavelet. Proses tresholding merupakan proses dimana pengurangan *noise* itu terjadi, pada proses ini koedisien detail dari citra di peroleh melalui proses dekomposisi akan dilewatkan pada fungsi “penyaring” yang disebut dengan fungsi thresholding.



Gambar 2.12: Fungsi Thresholding : (a) *Soft Thresholding*, (b) *Hard Thresholding*

Pada proses thresholding ini koefisien detail citra akan dibandingkan dengan suatu nilai *threshold* t , dan akan mendapat perlakuan sesuai dengan fungsi *threshold* yang digunakan. Ada dua fungsi pada *thresholding* yaitu *hard thresholding* dan *soft thresholding* dapat dilihat pada Gambar 2.12. Fungsi *hard thresholding* diperlihatkan pada akan merubah nilai detail menjadi nol apabila mutlak dari nilai detail lebih kecil dari nilai *threshold*, dan nilai detail tidak berubah jika nilainya lebih besar atau sama dengan nilai *threshold*. Sedangkan fungsi *soft thresholding* akan mengurangi nilai detail dengan nilai *threshold* jika nilai detail lebih besar atau sama dengan nilai *threshold*, atau merubahnya menjadi nol jika nilai mutlak dari nilai detail lebih kecil dari nilai *threshold*, dan menambahkan dengan nilai *threshold* jika nilai detail lebih kecil dari minus nilai *threshold* diperlihatkan pada Gambar 2.12.

Terdapat dua jenis *threshold* yaitu *Global Threshold* dan *Level Adaptive Threshold*. Pada *global threshold*, nilai *threshold* dihitung sekali dan berlaku secara global untuk seluruh level dekomposisi, sebaliknya pada *level adaptive threshold*, nilai *threshold* akan dihitung dari tiap nilai koefisien detail pada tiap level dekomposisi, sehingga pada tiap level dekomposisi, sehingga pada level akan diperoleh tiga nilai *threshold* untuk koefisien detail vertical, horizontal dan diagonal (H.Sianapar, 2013). Tiga teknik *soft thresholding* yang paling umum adalah *VisuShrink*, *SureShrink*, dan *BayesShrink*.

$$C(\sigma_x, \beta) = \frac{\beta \cdot \alpha(\sigma_x)}{2\Gamma(\frac{1}{\beta})} \quad (2.17)$$

Perkiraan dari *threshold* $T_B = \frac{\sigma^2}{\sigma_x}$ tidak hanya hampir optimal tetapi juga memiliki perbandingan yang intuitif. Normalisasi *threshold* $\frac{T_B}{\sigma}$ berbanding terbalik dengan σ , standar deviasi x dan berbanding terbalik dengan σ_x untuk standar deviasi *noise*. Bila $\frac{\sigma}{\sigma_x} \ll 1$, sinyal lebih kuat dari *noise*, $\frac{T_B}{\sigma}$ dipilih untuk menjadi lebih kecil untuk mempertahankan sebagian besar sinyal dan menghapus beberapa *noise*, bila $\frac{\sigma}{\sigma_x} \gg 1$ *noise* mendominasi dan normalisasi *threshold* dipilih untuk menghapus sinyal yang banyak yang telah diliputi *noise*. dengan demikian, pilihan untuk *threshold* adalah menyesuaikan dengan sinyal dan karakteristik *noise* sebagaimana parameter σ dan σ_x .

$$\tau(T) = E(\hat{x} - x)^2 = E_x E_{y|x}(\hat{x} - x)^2 \quad (2.18)$$

Persamaan 2.18 dimana $\hat{x} = \eta\tau(y), y|x \sim N(x, \sigma^2)$ dan $x \sim GG_{x,\beta}$ untuk mengoptimalkan T^* pada persamaan 2.19.

$$T^*(\sigma_x, \beta) = \operatorname{argmin}_T \tau(T) \quad (2.19)$$

Fungsi dari parameter σ_x dan β . Belum ada solusi untuk karena itu perhitungan numerik digunakan untuk mencari nilainya. hal ini diamati pada *threshold* yang ditetapkan oleh persamaan $x.x$ mendekati untuk T^* . Fungsi dari parameter T^* dan β . Belum ada solusi untuk T^* karena itu perhitungan numerik digunakan untuk mencari nilainya. hal ini diamati pada *threshold* yang ditetapkan oleh persamaan $x.x$ mendekati untuk T^* .

2.8 Teknik Penilaian Kualitas Citra

Penilaian kualitas citra telah dievaluasi oleh manusia secara subjektif, dimana sejumlah pengamat dipilih dengan kemampuan visual mereka untuk ditunjukkan serangkaian tes citra pada layar dan diminta menilai kualitas citra tersebut. Evaluasi subjektif ini dapat diandalkan namun membutuhkan waktu yang lama dan mahal, sehingga terdapat model komputasi secara otomatis dapat memprediksi kualitas citra (E. Bire, 2012). Adapun beberapa teknik yang digunakan untuk menilai kualitas citra antara lain yaitu *Mean Square Error (MSE)* dan *Peak Signal to noise Ratio (PSNR)*

2.8.1 Mean Square Error (MSE)

Pengukuran kualitas citra yang paling sederhana, dimana mempresen-
tasikan kekuatan *noise* atau perbedaan antara citra asli dan citra ber*noise*.
Nilai yang besar untuk MSE berarti citra tersebut berkualitas buruk. MSE
didefinisikan sebagai berikut (E. Bire, 2012) pada persamaan 2.20

$$MSE = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^M (x(m,n) - \hat{x}(m,n))^2 \quad (2.20)$$

$x(m,n)$ merupakan citra asli sedangkan $\hat{x}(m,n)$ merupakan citra terdis-
torsi. M dan N merupakan jumlah *pixel* baris dan kolom masing-masing citra.

2.8.2 Peak Signal to *noise* Ratio (PSNR)

PSNR adalah rasio antara kekuatan maksimum sinyal dan kekuatan
noise yang biasanya dinyatakan dalam bentuk decibel logaritmik (E. Bire,
2012). PSNR didefinisikan dengan persamaan 2.21

$$PSNR = 10 \log \frac{255^2}{MSE} \quad (2.21)$$

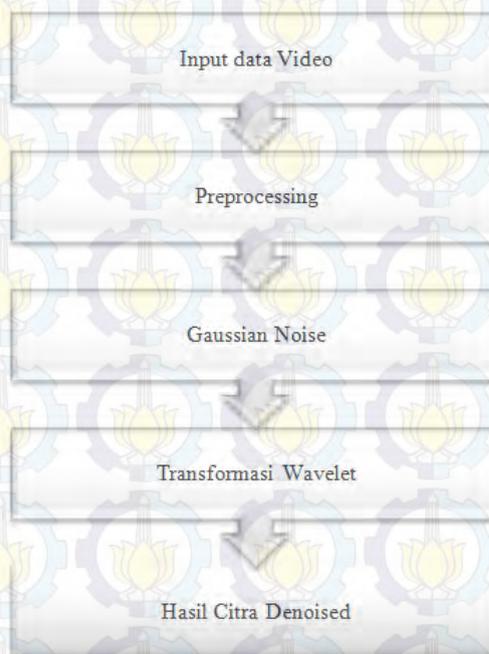
BAB III

METODA PENELITIAN

Pada penelitian ini akan dilakukan penelitian mengenai pengurangan *flicker* yang diasumsikan sebagai masalah *Gaussian noise* dengan menggunakan wavelet tresholding. Dalam penelitian ini, menggunakan video animasi kuno yaitu popaye, felix dan steamboat. *Input* data berupa data video yang telah di ekstrak menjadi frame-frame berdasarkan dari *framerate* masing-masing video. Secara umum metoda penelitian yang dilakukan dalam penelitian terdiri dari beberapa tahapan yang dijelaskan pada alur diagram Gambar 3.1.

3.1 Perancangan Sistem

Secara umum metode penelitian yang akan dilakukan dalam penelitian berikut secara garis besar terdiri dari beberapa tahapan yang terinci pada diagram Gambar 3.1.



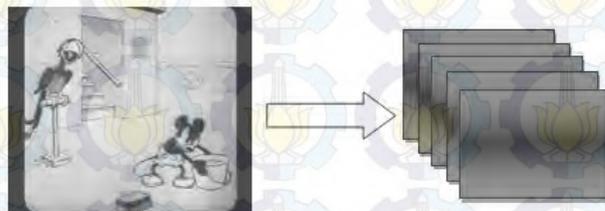
Gambar 3.1: Blok Diagram Sistem Pengurangan *Flicker* Pada Animasi Kuno

Proses awal dari penelitian ini adalah mengumpulkan data input berupa video animasi kuno. Video input berupa animasi kuno “popaye.wmv” dengan *framerate* 30 *frame/second* dan *Length* 00.00.03, ”Steamboat.wmv” dengan *framerate* 30 *frame/second* dan *Length* 00.00.03, dan “Felix.wmv” dengan *framerate* 30 *frame/second* dan *Length* 00.00.02. Ukuran dari masing-masing video adalah 320 *height* x 240 *width*. Perlu diketahui bahwa setelah melakukan ekstraksi pada video menjadi frame-frame maka hal selanjutnya dilakukan adalah menampilkan histogram dari antar frame dan dapat dilihat bahwa antar frame memiliki kemiripan (hampir sama) intensitas derajat keabuan.

Video selanjutnya dilakukan proses preprocessing image dimana video animasi kuno di *convert* menjadi frame-frame berdasarkan jumlah *framerate*, dari hasil pre-processing akan dipakai selanjutnya sebagai proses dekomposisi menggunakan DWT (*discrete wavelet transform*) dengan tipe wavelet Haar dengan level 5 yang melalui dua kali proses dekomposisi, hasil dari proses dekomposisi digunakan untuk proses *tresholding* dengan menggunakan teknik *soft tresholding* yaitu bayesshrink, hasil dari proses *tresholding* kemudian di rekontruksi ulang dengan IDWT (*invers descret wavelet transform*) menjadi frame hasil *denoised*.

3.2 Pre-processing

Tahap kedua yang dilakukan setelah video di *convert* menjadi 30 frame adalah melakukan konvolusi filter spasial linear pada domain spasial dijelaskan pada Gambar 3.2. Setelah konvolusi dilakukan pada frame yang mengandung *flicker* yang dianggap sebagai masalah *Gaussian noise*, dimana pada setiap frame *noise* dapat dibangkitkan dengan cara membangkitkan bilangan acak [0] dengan melakukan distribusi *Gaussian*. Kemudian untuk titik-titik yang terkena *noise*, nilai fungsi citra ditambahkan dengan nilai *noise* yang ada. Proses dekomposisi adalah tahap selanjutnya dilakukan setelah frame mengandung *Gaussian noise*.

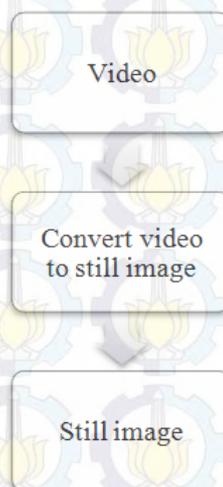


Gambar 3.2: Bagan Alir Proses Konversi Video menjadi *Still Image*

Dalam alur proses konversi video menjadi *still image* sesuai dengan panjang *frame per second* dari input video. Rata-rata durasi waktu yang dimiliki data input video berdurasi 7 detik. Tahap selanjutnya adalah proses konversi frame asli ke dalam bentuk *grayscale* dengan menggunakan teknik konvolusi dapat dilihat pada Gambar 3.3.



Gambar 3.3: Diagram alur konversi frame asli menjadi frame *grayscale*

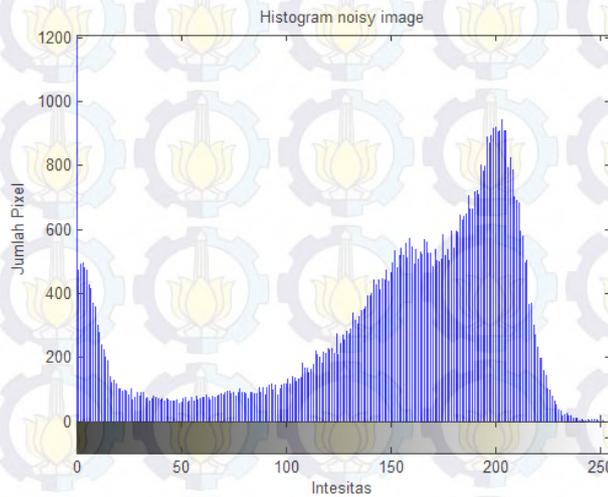


Gambar 3.4: Contoh Representasi *Gaussian noise*

nilai intensitas 201, piksel 1677 memiliki nilai intensitas 192.



(a) Frame ke-1 “Steamboat” *Gaussian noise*



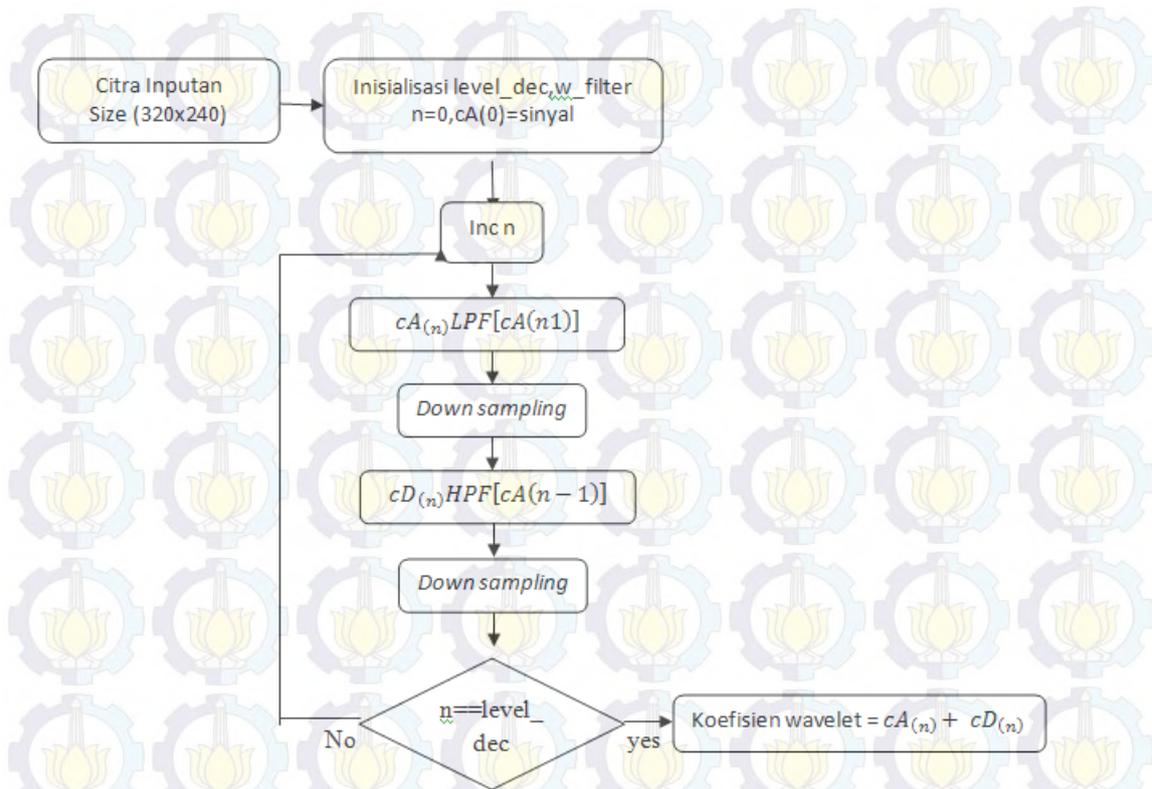
(b) Histogram Frame ke-1 “Steamboat” *Gaussian noise*

Gambar 3.6: (a) Frame ke-1 “Steamboat” *Gaussian noise*=0.01, (b) Histogram frame ke-5 “Steamboat” *Gaussian noise*=0.01.

3.2.2 Discrete Wavelet Transform (DWT)

Pada tahap awal langkah yang harus dilakukan adalah memasukkan data input berupa frame yang telah dijadikan *Gaussian noise* dan memilih tipe wavelet yang digunakan yaitu wavelet Haar dengan dekomposisi 5 level. Diagram alir dari proses transformasi wavelet diskrit dapat dilihat pada Gambar 3.7, data frame yang diinputkan adalah frame yang telah di konvolusi dan dijadikan *Gaussian noise*.

Data input selanjutnya di filter dan *down sampling*. Kedua proses tersebut dilakukan secara berturut-turut pada baris dan kolom citra. Didekom-

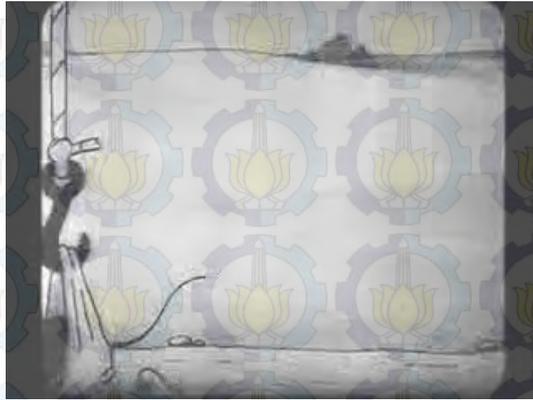


Gambar 3.7: Diagram alir transformasi wavelet diskrit

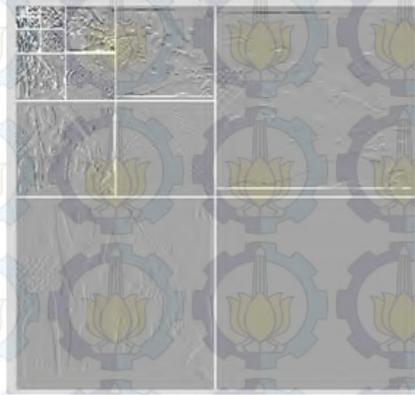
posisi dilakukan pada level 1 yang menghasilkan koefisien detail dan koefisien aproksimasi. Koefisien aproksimasi adalah koefisien yang selanjutnya digunakan untuk proses dekomposisi level selanjutnya sampai mencapai dekomposisi pada 5 level. Gambar 3.8 akan menunjukkan bentuk dekomposisi citra 5 level.

Pada Gambar 3.8 menunjukkan hasil dekomposisi 5 level pada frame yang telah mengandung *Gaussian noise* dimana data dua dimensi diganti dengan empat blok yang disesuaikan dengan subband yang mewakili *low pass* filtering dan *high pass* filtering di setiap arah.

Transformasi dilakukan pada semua baris yang menghasilkan matrik, dimana sisi kiri berisi koefisien *low pass down sample* dari setiap baris dan sisi kanan berisi koefisien *high pass* selanjutnya dekomposisi dilakukan untuk semua kolom. Dari proses DWT didapatkan nilai koefisien-koefisien wavelet berupa LL, LH, HL dan HH. Untuk proses perbaikan citra dilakukan pada subband LH, HL, dan HH sedangkan subband LL disebut koefisien aproksimasi dan diproses sampai 5 level selanjutnya. Cara yang dilakukan untuk menghilangkan *noise* akibat *Gaussian noise* dilakukan dengan proses tresholding



(a) Frame ke-1 “Steamboat”



(b) Hasil Dekomposisi Frame ke-1 “Steamboat”

Gambar 3.8: Hasil Dekomposisi dari Frame ke-1 “Steamboat” menjadi 5 level menggunakan teknik *soft* tresholding yaitu *Bayes* Tresholding.

3.2.3 Wavelet Tresholding

Proses denoised merupakan proses tresholding, sehingga proses perbaikan citra dilakukan pada koefisien detail saja. Pada tahap ini berfokus pada perkiraan parameter σ_x dan β dari GGD (*generalized Gaussian distribution*), dan menghasilkan perkiraan data-driven dari yang adaptif d subband yang berbeda untuk koefisien wavelet koefisien wavelet disetiap subband, kemudian dicari threshold T yang dapat meminimalkan resiko Bayesian.. Varians *noise* σ^2 perlu diperkirakan terlebih dahulu.

$$\sigma = \frac{\text{Median}|Y_{ij}|}{0.6745}, Y_{ij} \in \text{subband}d_1^{\text{diagonal}} \quad (3.1)$$

Dalam beberapa situasi, dimungkinkan untuk mengukur σ^2 berdasarkan informasi dari frame yang berbeda. Jika seperti ini tidak terjadi, diperkirakan dari subband $[HH_1]$ oleh perkiraan median robust, juga digunakan dalam (Donoho, 1995) persamaan 3.1.

Parameter β tidak secara eksplisit masuk ke dalam bentuk ekspresi dari $[T_B(\sigma_x)]$, hanya sinyal dari standar deviasi σ_x . Oleh karena itu sudah cukup untuk memperkirakan secara langsung σ_x atau σ_x^2 .

Ingat model pengamatan $Y = X + V$, dengan X dan V independen satu sama lain, maka dinamakan adalah varian dilihat pada persamaan 3.2.

$$\sigma_x^2 = \sigma_y^2 + \sigma^2 \quad (3.2)$$

Dimana $\hat{\sigma}_x^2$ adalah varians Y . Dimana Y dimodelkan sebagai Zero-Mean, dapat dilihat secara empiris dengan persamaan 3.3.

$$\hat{\sigma}_x^2 = \frac{1}{n^2} \sum_{i,j=1}^n Y_{ij}^2 \quad (3.3)$$

Dimana $n \times n$ adalah ukuran dari subband yang telah dipertimbangkan, karena persamaan 3.5:

$$\hat{\sigma}_x = \sqrt{\max(\sigma_y^2 - \hat{\sigma}_x^2, 0)} \quad (3.4)$$

$$T_B(\hat{\sigma}_x) = \frac{\hat{\sigma}_x^2}{\hat{\sigma}_x} \quad (3.5)$$

Dimana pada penelitian ini $\hat{\sigma}_x^2 \geq \sigma_y^2$, $\hat{\sigma}_x$ dianggap 0. Artinya $T_B(\hat{\sigma}_x)$ adalah ∞ atau dalam persamaannya $T_B(\hat{\sigma}_x) = \max|Y_{ij}|$ dan semua koefisien di atur menjadi 0. Hal ini terjadi disaat σ bernilai besar (contohnya , $\sigma > 20$ untuk frame grayscale.

3.2.4 InversWavelet Transform (IDWT)

Tahap rekontruksi koefisien detail melalui proses IDWT (*inverse discrete wavelet transform*), rekontruksi dilakukan untuk mneggabungkan semua subband dan hasil rekontruksi berupa citra yang telah di denoised (pengerangan *flicker* pada setiap frame). Tahap selanjutnya yang dilakukan untuk menunjukkan penilaian dari kualitas citra yang telah di denoised maka dilakukan pengukuran kualitas citra menggunakan MSE dan PSNR.

3.2.5 Mean Square Error (MSE)

Pengukuran kualitas citra yang paling sederhana, dimana mempresen-tasikan kekuatan *noise* atau perbedaan antara citra asli dan citra bernoise. Nilai yang besar untuk MSE berarti citra tersebut berkualitas buruk. MSE didefenisikan dengan persamaan 3.6.

$$MSE = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^M (x(m,n) - \hat{x}(m,n))^2 \quad (3.6)$$

$X(m,n)$ merupakan citra asli sedangkan $x(m,n)$ merupakan citra terdistorsi. M dan N merupakan jumlah piksel baris dan kolom masing-masing citra.

3.2.6 Park Signal to *noise* Ratio (PSNR)

Pada persamaan 3.7 merupakan decibel logaritmatik antara rasio kekuatan maksimum sinyal dan kekuatan *noise*.

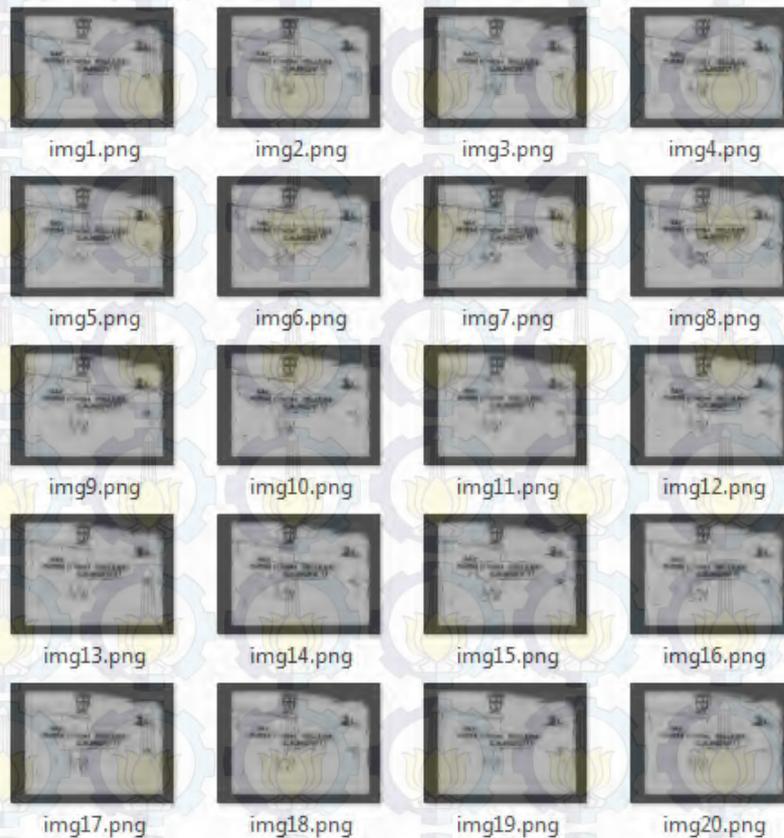
$$PSNR = 10 \log \frac{255^2}{MSE} \quad (3.7)$$

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini akan dibahas hasil evaluasi efektifitas metode yang digunakan dalam mengatasi permasalahan pengurangan flicker pada video animasi kuno yang dianggap sebagai masalah gaussian noise. Pengujian dilakukan dengan mengikuti berbagai urutan algoritma sesuai dengan blok diagram pada Gambar 3.1 .

Percobaan menggunakan 30 data citra grayscale 8 bit ($2^8 = 256$) hasil dari pembagian video animasi kuno dengan ukuran 320×240 piksel yang selanjutnya dilakukan proses reduksi (pengurangan) noise pada wavelet 2-D dengan jenis wavelet haar dan metode tresholding yang digunakan *bayesshrink* pada Gambar 4.1.



Gambar 4.1: Frame hasil ekstrak video “popeye.wmv”

4.1 Skenario Uji Coba

Skenario uji coba merupakan urutan-urutan proses yang dilakukan pada saat pengujian, diantaranya:

1. Ekstraksi Video animasi kuno “popeye”, “steamboat” dan “felix” menjadi 30 frame berdasarkan *framerate* dari setiap video input.
2. Mengkonvolusi frame hasil ekstraksi dengan melakukan filter spasial linear pada domain spasial menjadi frame *grayscale*.
3. Frame yang mengandung *flicker* dianggap sebagai masalah *Gaussian noise*, dimana pada setiap frame noise dibangkitkan dengan bilangan acak [0] yang berarti diberikan nilai nol pada setiap pixel dengan melakukan distribusi *Gaussian*.
4. Transformasi wavelet diskrit dilakukan pada frame yang telah menjadi *Gaussian noise* dengan menggunakan tipe wavelet haar level 5.
5. Koefisien detail hasil dari dekomposisi pada setiap subband di tresholding (proses denoising) dengan teknik bayesshrink.
6. Subband koefisien detail hasil tresholding dan koefisien aproksimasi direkonstruksi ulang dan digabungkan menjadi frame hasil denoised.
7. Pengukuran kualitas hasil denoised menggunakan MSE dan PSNR.

Pengujian ini dilakukan dengan menjadikan flicker menjadi permasalahan *Gaussian noise*. Noise yang ditambahkan adalah *Gaussian noise*. Penghilangan noise tersebut dengan menggunakan *Bayes Threshold*. Selanjutnya dilakukan proses denoising menggunakan wavelet thresholding dengan teknik bayesshrink pada koefisien detail dari setiap subband hasil dari dekomposisi *discrete wavelet transform* (DWT).

4.2 Pre-Processing

Pada proses pre-processing hal yang pertama dilakukan adalah mengekstrak video input menjadi frame-frame yang selanjutnya dikonvolusi menjadi citra *grayscale* untuk melihat bentuk histogram yang memiliki frekuensi kemunculan derajat keabuan dari frame yang mengandung noise (*flicker*) antar frame 1 sampai frame ke-30. Pada Gambar 4.2, Gambar 4.3 dan Gambar 4.4 dapat dilihat histogram yang memiliki intensitas pixel yang menonjol

dari antar frame yang mengandung noise pada antar frame dari setiap video input "popoeye", "Steamboat" dan "Felix".

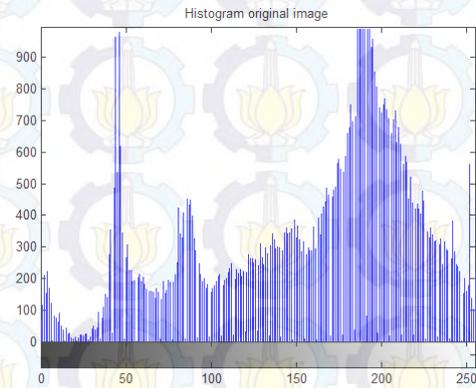
Dari Gambar 4.2 dapat dilihat bahwa antar frame dari video input "popoeye" tidak memiliki perbedaan intensitas derajat keabuan yang tidak jauh berbeda ditunjukkan dengan histogram yang pada antar frame mengandung *noise*. Pada antar framenya terlihat perbedaan intensitas derajat keabuan pada range 50 sampai dengan 255. Setelah semua frame di *convert* menjadi *grayscale* maka selanjutnya akan dilakukan pembangkitan *gaussian noise*.

Dari Gambar 4.3 dapat dilihat bahwa antar frame dari video input "steamboat" memiliki perbedaan intensitas derajat keabuan yang hampir sama ditunjukkan dengan histogram yang pada antar frame mengandung *noise*. Pada antar framenya terlihat perbedaan intensitas derajat keabuan pada range 130 sampai dengan 210.

Dari Gambar 4.4 dapat dilihat bahwa antar frame dari video input "Felix" memiliki perbedaan intensitas derajat keabuan yang hampir sama ditunjukkan dengan histogram yang pada antar frame mengandung *noise*. Pada antar framenya terlihat perbedaan intensitas derajat keabuan pada range 120 sampai dengan 150.



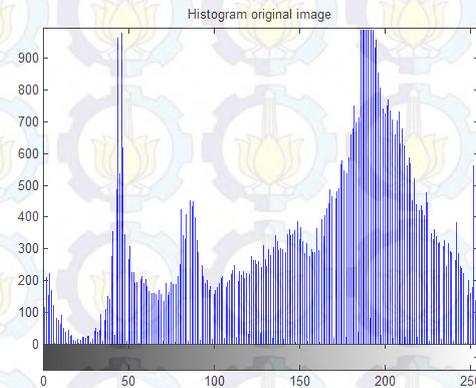
(a) Frame ke-1 dari video input "popeye"



(b) Histogram frame ke-1 dari video input "popeye"



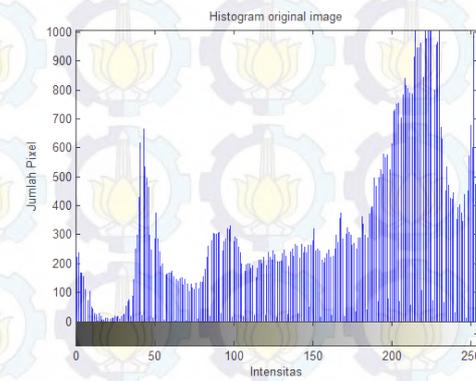
(c) Frame ke-2 dari video input "popeye"



(d) Histogram frame ke-2 dari video input "popeye"



(e) Frame ke-3 dari video input "popeye"

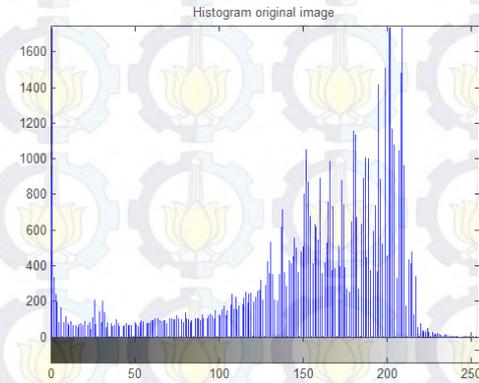


(f) Histogram frame ke-3 dari video input "popeye"

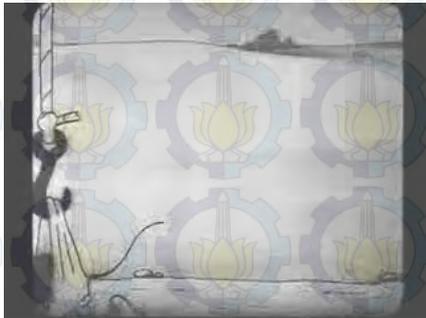
Gambar 4.2: Frame ke-1, frame ke-2 dan frame ke-3 serta histogram dari video input "popeye"



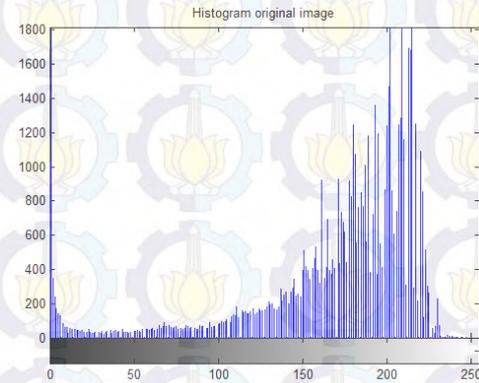
(a) Frame ke-1 dari video input "Steamboat"



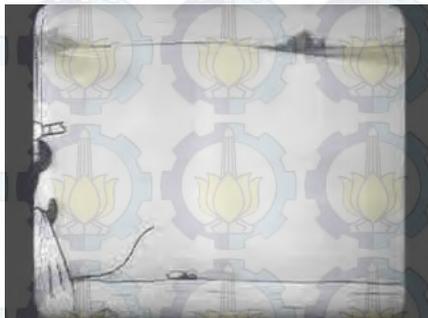
(b) Histogram frame ke-1 dari video input "Steamboat"



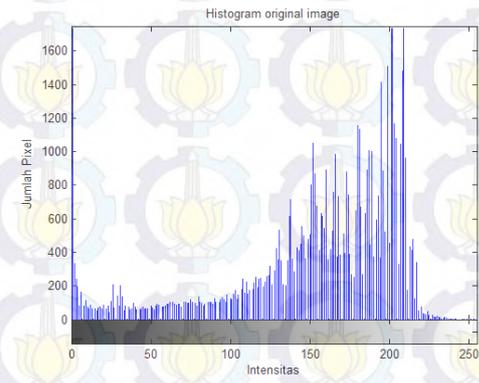
(c) Frame ke-2 dari video input "Steamboat"



(d) Histogram frame ke-2 dari video input "Steamboat"



(e) Frame ke-3 dari video input "Steamboat"

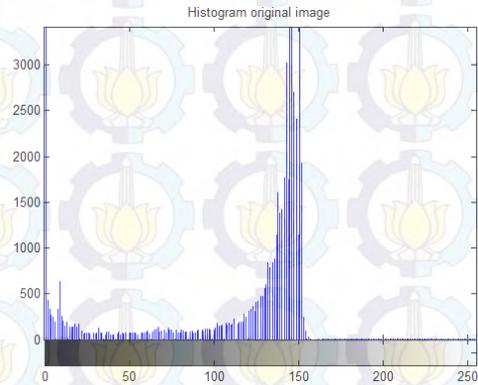


(f) Histogram frame ke-3 dari video input "Steamboat"

Gambar 4.3: Frame ke-1, frame ke-2 dan frame ke-3 serta histogram dari video input "Steamboat"



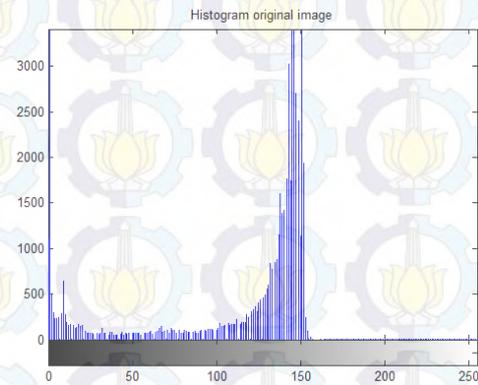
(a) Frame ke-1 dari video input "Felix"



(b) Histogram frame ke-1 dari video input "Felix"



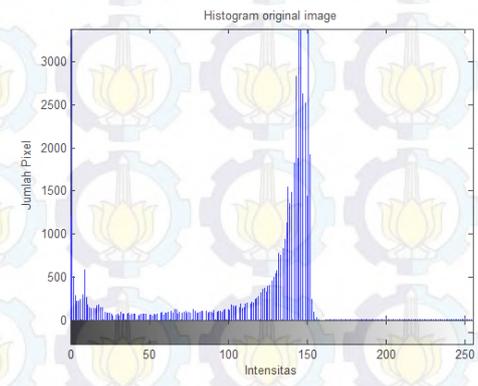
(c) Frame ke-2 dari video input "Felix"



(d) Histogram frame ke-2 dari video input "Felix"



(e) Frame ke-3 dari video input "Felix"



(f) Histogram frame ke-3 dari video input "Felix"

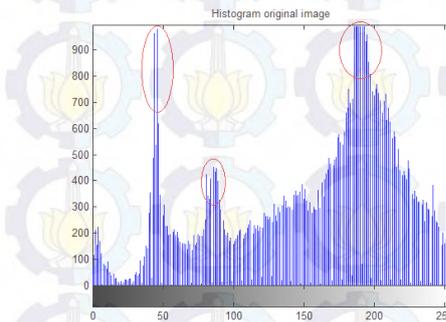
Gambar 4.4: Frame ke-1, frame ke-2 dan frame ke-3 serta histogram dari video input "Felix"

4.3 Membangkitkan *Gaussian Noise*

Pada tahap *gaussian noise* melakukan proses membangkitkan bilangan acak berdistribusi *Gaussian* menggunakan fungsi *rnd* dengan memberikan nilai *default* dari *gaussian* 0 dengan variansi 0.01 untuk setiap pixel antar frame yang histogramnya dapat dilihat pada Gambar 4.6(c), Gambar 4.7(c) dan Gambar 4.9 gaussian noise pada antar frame video input "Popaye", "Steamboat" dan "Felix".



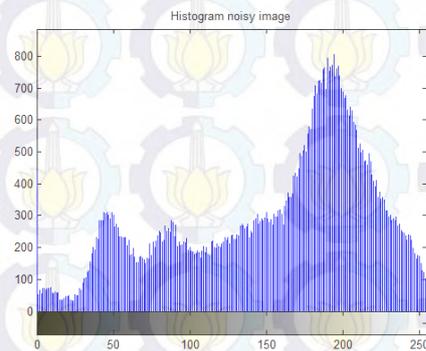
(a) Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Popaye"



(b) Histogram Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Popaye"



(c) Data Input Frame pertama sesudah di *Gaussian Noise* dari video input "Popaye"



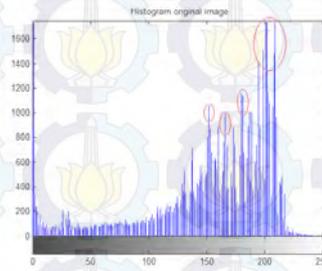
(d) Histogram Data Input Frame pertama sesudah di *Gaussian Noise* dari video input "Popaye"

Gambar 4.5: Data Input Frame pertama sebelum dan sesudah di *Gaussian Noise* serta histogram dari video input "Popaye"

Pada histogram dengan video input "Popaye" tanda pada Gambar 4.5(b) dan menunjukkan nilai intensitas derajat keabuan yang terlalu tinggi pada frame pertama dan setelah dibangkitkan dengan *Gaussian Noise* maka terjadi perbedaan pada nilai intensitas derajat keabuan dari frame pertama terlihat pada Gambar 4.5(d). Begitu pula pada Gambar 4.6(b) menunjukkan perbedaan nilai intensitas derajat keabuan yang terlalu tinggi sehingga setelah



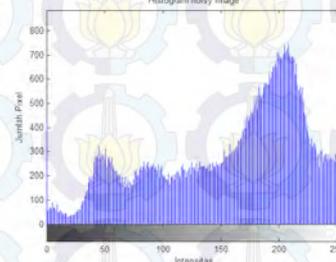
(a) Data Input Frame kedua sebelum di *Gaussian Noise* dari video input "popaye"



(b) Histogram Data Input Frame kedua sebelum di *Gaussian Noise* dari video input "popaye"



(c) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "popaye"

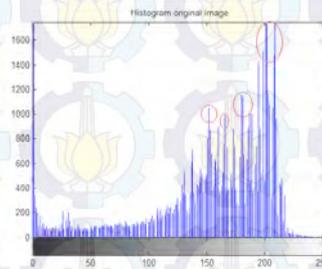


(d) Histogram Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "popaye"

Gambar 4.6: Data Input Frame kedua sebelum dan sesudah di *Gaussian Noise* serta histogram dari video input "popaye"



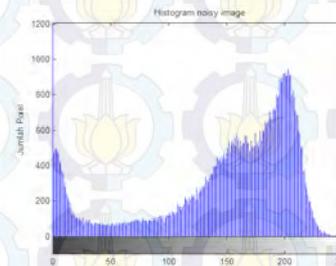
(a) Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Steamboat"



(b) Histogram Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Steamboat"



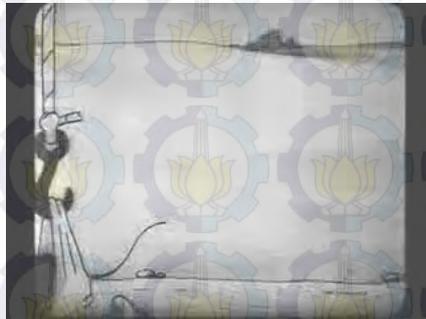
(c) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "Steamboat"



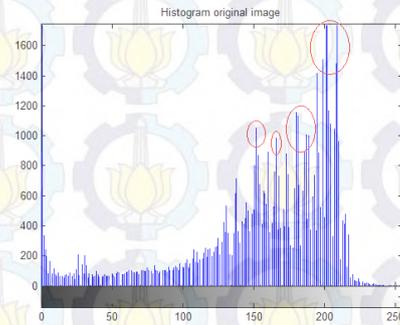
(d) Histogram Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "Steamboat"

Gambar 4.7: Data Input Frame pertama sebelum dan sesudah di *Gaussian Noise* serta histogram dari video input "Steamboat"

di Gaussian Noise menunjukkan intensitas derajat keabuan yang berbeda pada Gambar 4.6(d).



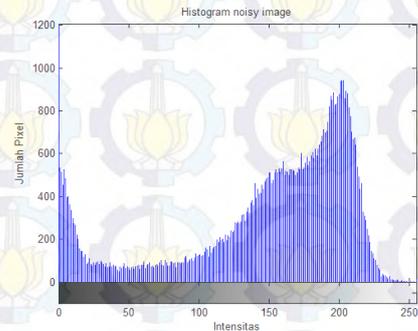
(a) Data Input Frame kedua sebelum di *Gaussian Noise* dari video input ”Steamboat”



(b) Histogram Data Input Frame pertama sebelum di *Gaussian Noise* dari video input ”Steamboat”



(c) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input ”Steamboat”

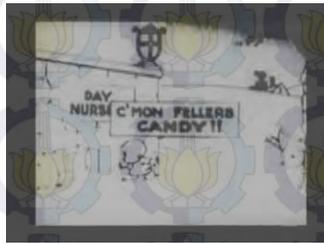


(d) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input ”Steamboat”

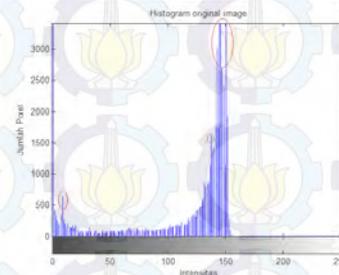
Gambar 4.8: Data Input Frame kedua sebelum dan sesudah di *Gaussian Noise* serta histogram dari video input ”Steamboat”

Pada histogram dengan video input ”Steamboat” tanda pada Gambar 4.7(b) dan menunjukkan nilai intensitas derajat keabuan yang terlalu tinggi pada frame pertama dan setelah dibangkitkan dengan *Gaussian Noise* maka terjadi perbedaan pada nilai intensitas derajat keabuan dari frame pertama terlihat pada Gambar 4.7(d). Begitu pula pada Gambar 4.8(b) menunjukkan perbedaan nilai intensitas derajat keabuan yang terlalu tinggi sehingga setelah di Gaussian Noise menunjukkan intensitas derajat keabuan yang berbeda pada Gambar 4.8(d).

Pada histogram video input ”Felix” tanda pada Gambar 4.9(b) dan menunjukkan nilai intensitas derajat keabuan yang terlalu tinggi pada frame pertama dan setelah dibangkitkan dengan *Gaussian Noise* terjadi perbedaan



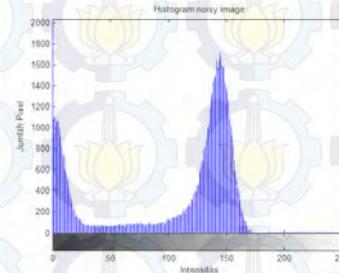
(a) Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Felix"



(b) Histogram Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Felix"



(c) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "Felix"

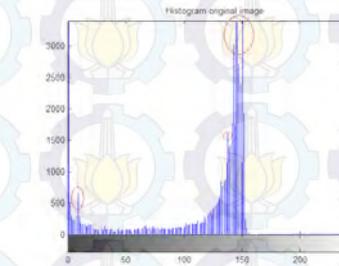


(d) Histogram Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "Felix"

Gambar 4.9: Data Input Frame pertama sebelum dan sesudah di *Gaussian Noise* serta histogram dari video input "Felix"



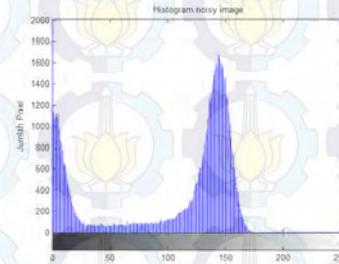
(a) Data Input Frame kedua sebelum di *Gaussian Noise* dari video input "Felix"



(b) Histogram Data Input Frame pertama sebelum di *Gaussian Noise* dari video input "Felix"



(c) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "Felix"



(d) Data Input Frame kedua sesudah di *Gaussian Noise* dari video input "Felix"

Gambar 4.10: Data Input Frame kedua sebelum dan sesudah di *Gaussian Noise* serta histogram dari video input "Felix"

pada nilai intensitas derajat keabuan dari frame pertama terlihat pada Gambar 4.9(d). Pada Gambar 4.10(b) menunjukkan perbedaan nilai intensitas derajat keabuan yang terlalu tinggi sehingga setelah di Gaussian Noise menunjukkan intensitas derajat keabuan yang berbeda pada Gambar 4.10(d).

4.4 Pengujian *Discrete Wavelet Transform* video input ”‘popeye’”

Proses dekomposisi 5 level dilakukan setelah frame asli menjadi frame *Gaussian noise*, melalui proses DWT dimana data dua dimensi diganti dengan empat blok yang bersesuaian dengan subband yang mewakili *low pass* filtering dan *high pass* filtering di setiap arah. Pertama, dilakukan transformasi pada semua baris yang menghasilkan matrik, dimana sisi kiri berisi koefisien *low pass down sample* dari setiap baris, dan sisi kanan berisi koefisien *high pass*. Kemudian dekomposisi diterapkan untuk semua kolom.

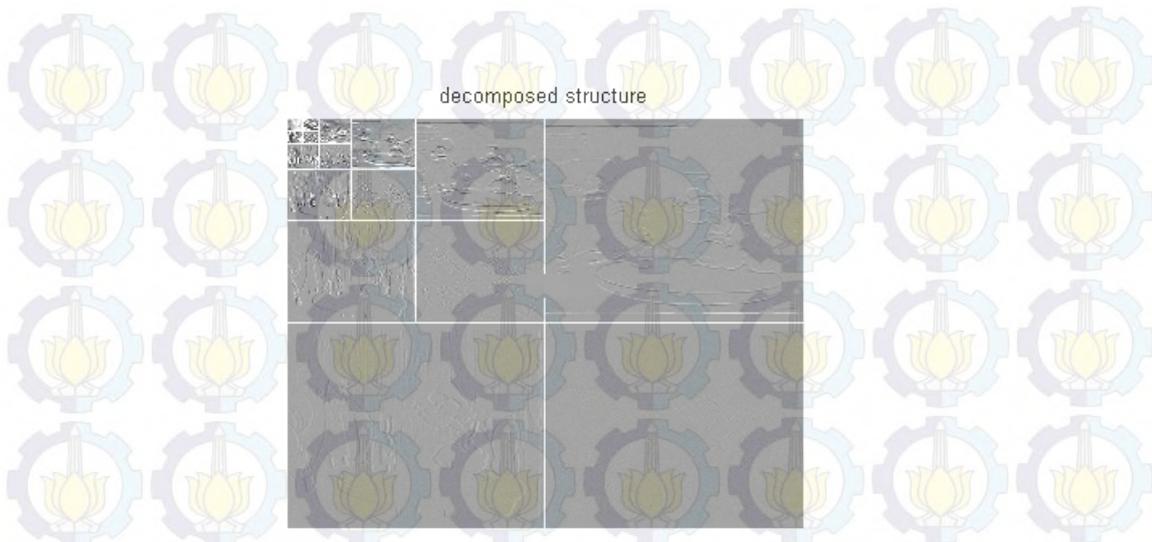


Gambar 4.11: Frame ke-1 dari video input ”‘popeye’”

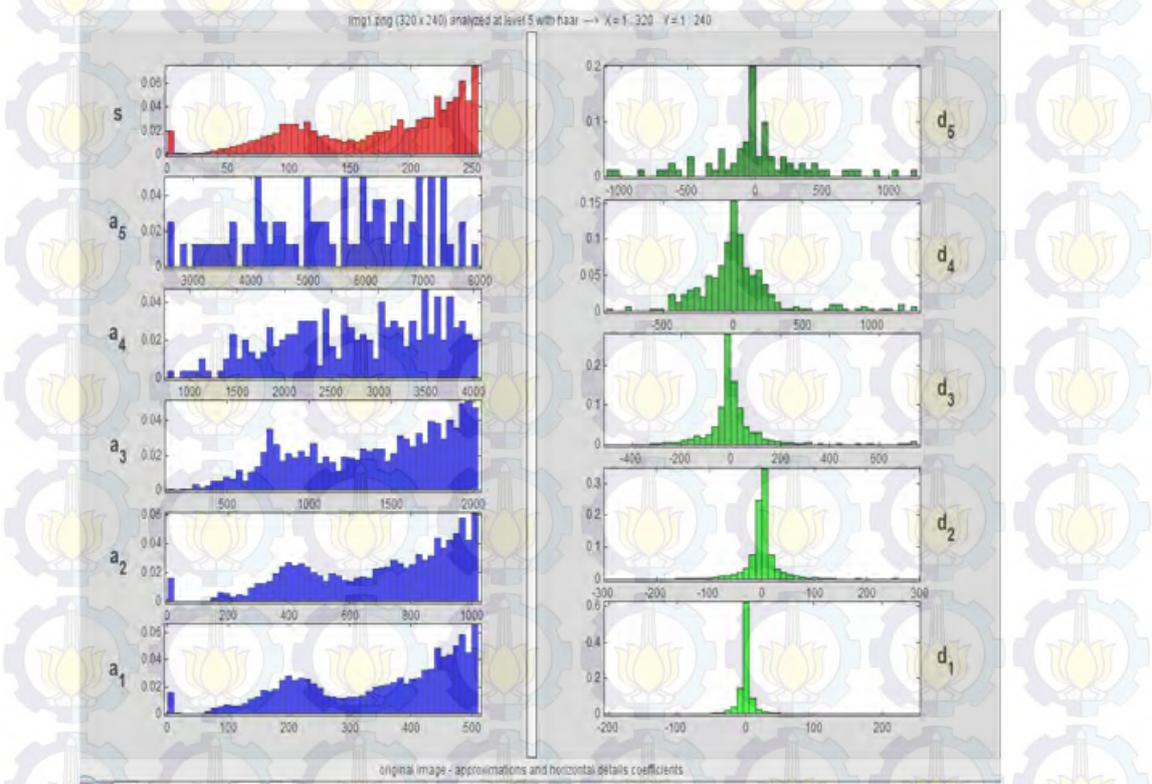
Pada proses DWT, didapatkan koefisien-koefisien wavelet berupa subband LL1, subband LH1, subband HL1 dan subband HH1. Kedua proses tersebut dilakukan sebanyak lima kali, terhadap baris dan terhadap kolom sehingga diperoleh 13 subband keluaran. Pada Gambar 4.12 diatas adalah hasil dari proses dekomposisi level 5 yang dilakukan pada subband LL, LH, HL, dan HH dimana hasilnya berupa 1 subband HL, LH dan HH yang disebut sebagai berupa fitur-fitur dari koefisien detail dan subband LL adalah koefisien aproksimasi yang kemudian akan digunakan untuk proses level selanjutnya.

4.5 Pengujian Tresholding koefisien detail wavelet level 5 video input ”‘popeye’”

Parameter dari σ_x adalah standar deviasi dan β adalah bentuk parameter yang telah diamati bahwa dengan parameter β mulai dari 0 sampai 0.01, dapat



Gambar 4.12: Frame ke-1 dari video input "popeye" didekomposisi sampai level 5



Gambar 4.13: Frame pertama dan Histogram koefisien aproksimasi dan koefisien detail dari hasil dekomposisi 5 level wavelet Haar dari video input "popeye"

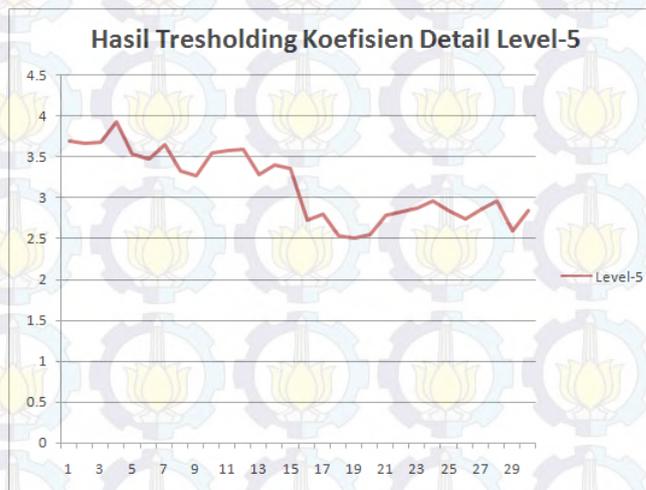
menggambarkan distribusi pada koefisien dalam subband diatur untuk frame yang besar.

Tabel 4.1: Pengujian Tresholding koefisien detail wavelet level 5 video input ”‘pop-eye’”

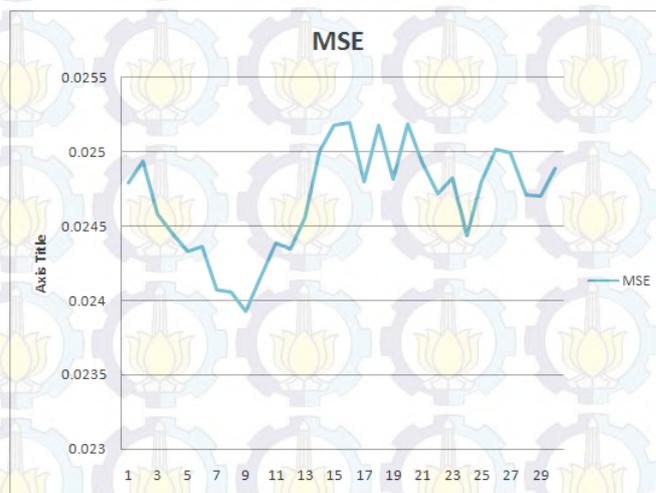
Frame ke-	Level-1	Level-2	Level-3	Level-4	Level-5	MSE
Frame ke 1	0.0302	0.32237	0.6781	1.5309	3.6882	0.0248
Frame ke 2	0.0302	0.32084	0.6778	1.53	3.6677	0.0249
Frame ke 3	0.0298	0.29242	0.6924	1.5417	3.6806	0.0246
Frame ke 4	0.0352	0.35863	0.8748	2.0392	3.9272	0.0244
Frame ke 5	0.0338	0.34973	0.8785	1.9102	3.5346	0.0243
Frame ke 6	0.0292	0.30011	0.7234	1.4987	3.4747	0.0244
Frame ke 7	0.0283	0.30516	0.6847	1.582	3.6431	0.0241
Frame ke 8	0.0293	0.31307	0.6828	1.5814	3.3331	0.0241
Frame ke 9	0.0245	0.26655	0.5877	1.4019	3.2627	0.0239
Frame ke 10	0.0291	0.3051	0.6892	1.5054	3.5513	0.0242
Frame ke 11	0.0294	0.30653	0.6828	1.5077	3.5817	0.0244
Frame ke 12	0.0293	0.30438	0.6839	1.4349	3.5949	0.0243
Frame ke 13	0.0228	0.19004	0.4301	1.1144	3.2754	0.0246
Frame ke 14	0.0244	0.20013	0.4461	1.1189	3.3966	0.025
Frame ke 15	0.0245	0.20066	0.4462	1.0908	3.3581	0.0252
Frame ke 16	0.0253	0.19947	0.4581	1.1502	2.7199	0.0252
Frame ke 17	0.0249	0.20056	0.4463	1.1422	2.7986	0.0248
Frame ke 18	0.0249	0.1971	0.4396	1.1338	2.5408	0.0252
Frame ke 19	0.025	0.19877	0.4482	1.1736	2.5057	0.0248
Frame ke 20	0.0252	0.19642	0.4487	1.1046	2.5495	0.0252
Frame ke 21	0.0302	0.23662	0.5329	1.3311	2.7874	0.0249
Frame ke 22	0.0299	0.23565	0.5156	1.3247	2.8246	0.0247
Frame ke 23	0.0314	0.2403	0.5298	1.363	2.8673	0.0248
Frame ke 24	0.0316	0.24024	0.5455	1.4307	2.9561	0.0244
Frame ke 25	0.032	0.23709	0.5475	1.3923	2.846	0.0248
Frame ke 26	0.0313	0.23508	0.53	1.3389	2.7472	0.025
Frame ke 27	0.031	0.24487	0.5217	1.2861	2.8571	0.025
Frame ke 28	0.0313	0.2418	0.5431	1.3131	2.9628	0.0247
Frame ke 29	0.0264	0.20602	0.4686	1.227	2.5893	0.0247
Frame ke 30	0.0316	0.24038	0.5434	1.3495	2.841	0.0249

Dengan mengasumsikan distribusi tersebut untuk koefisien wavelet, telah diperkirakan secara empiris β dan σ_x untuk setiap subband koefisien detail yang telah diperoleh dijadikan masukan pada proses tresholding dimana nilai tengah dari subband koefisien detail (d_5) di bagi dengan nilai 0.674 untuk mendapatkan nilai *threshold* dan menemukan nilai T treshold untuk data input ”‘pop-eye’” adalah 8.8955 yang meminimalkan resiko Bayesian dengan

menganalisa dari hasil proses mean square error.

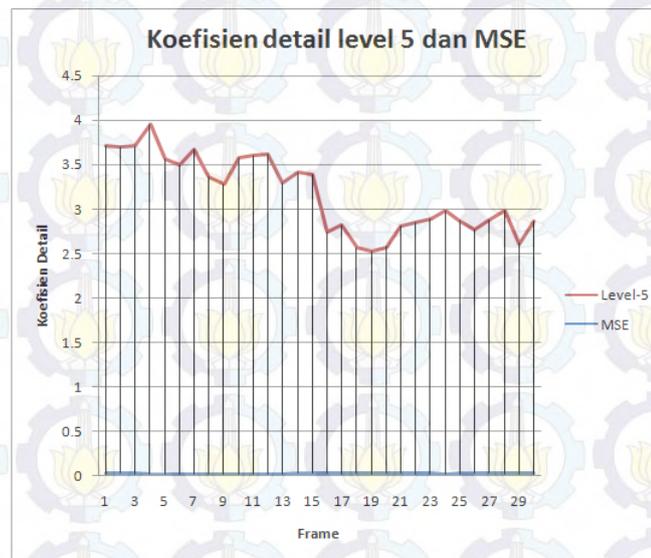


Gambar 4.14: Grafik Hasil pengujian threshold level 5 dari video input “pop-eye”



Gambar 4.15: Grafik Hasil pengujian MSE dari koefisien detail level 5 dari video input “pop-eye”

Tabel 4.1 menunjukkan nilai *threshold* dan *mean square error* dilihat dari grafik pada Gambar 4.14 dan Gambar 4.15 dapat dianalisa bahwa nilai *threshold* dari setiap subband pada koefisien detail antar frame memiliki tingkat kecenderungan perubahan intensitas derajat keabuan yang tidak jauh berbeda antar framennya. Perbedaan intensitas derajat keabuan pada antar frame hasil tresholding menggunakan bayesshrink mengalami pengurangan



Gambar 4.16: Grafik Hasil pengujian koefisien detail level 5 dan MSE dari video input “popeye”

flicker (noise) yang baik antar framenya yang ditunjukkan oleh dari Gambar 4.16.

4.6 Pengujian *inverse discrete wavelet transform (IDWT)*

Proses IDWT (*inverse discrete wavelet transform*) dilakukan setelah proses tresholding pada koefisien detail yang kemudian di rekontruksi atau menggabungkan semua subband dari hasil dekomposisi yaitu dengan pproses *up-sampling* dan pemfilteran dengan koefisien-koefisien filter balik. Proses ini dilakukan dengan menyisipkan sebuah kolom yang berharga nol diantara setiap kolom dan melakukan konvolusi pada setiap baris dengan filter satu dimensi. Penyisipan juga dilakukan pada sebuah baris yang berharga nol diantara setiap kolom dan melakukan konvolusi pada setiap kolom dengan filter yang lain.

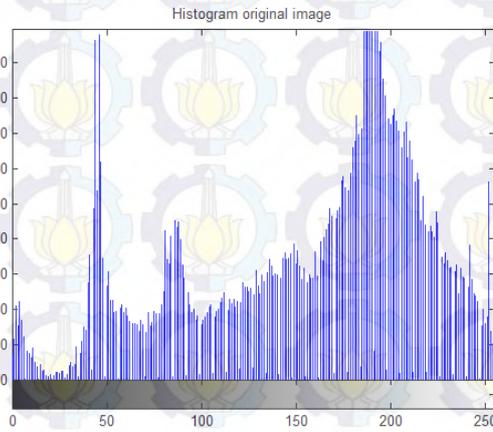
Pada Gambar 4.17 dan Gambar 4.18 menunjukkan frame pertama dan kedua sebelum dan sesudah terjadi proses *invers wavelet transform* atau disebut sebagai proses *denoised* dan histogram hasil dekomposisi citra yang sebelum diberikan noise (*Gaussian noise*) dapat terlihat dengan adanya perbedaan intensitas derajat keabuan pada citra asli dan citra yang telah di *denoised*. Perbandingan citra asli dan citra yang telah di *denoised* terlihat pada intensitas derajat keabuan antar frame yang dapat dianalisa bahwa pengurangan noise (*flicker*) pada antar frame menggunakan wavelet *tresholding* dengan teknik bayesshrink memberikan hasil yang baik.



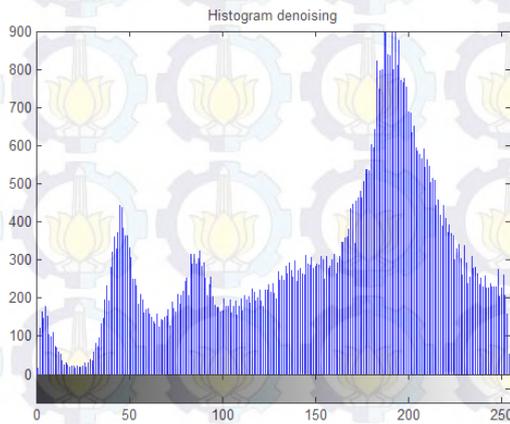
(a) Frame ke-1 dari video input "popeye"



(b) Frame ke-1 hasil *denoised* dari video input "popeye"



(c) Histogram frame ke-1 dari video input "popeye"



(d) Histogram frame ke-1 hasil *denoised* dari video input "popeye"

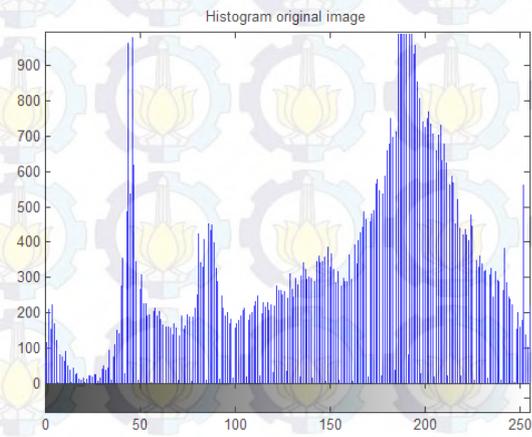
Gambar 4.17: Frame pertama sebelum di *denoised* dan frame hasil *denoised* dan histogram dari video input "popeye"



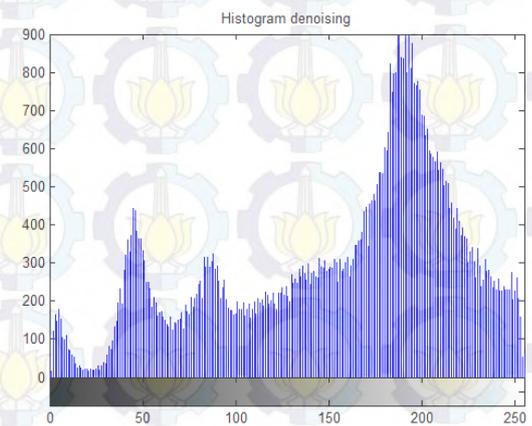
(a) Frame ke-2 dari video input "popeye"



(b) Frame ke-2 hasil *denoised* dari video input "popeye"



(c) Histogram frame ke-2 dari video input "popeye"

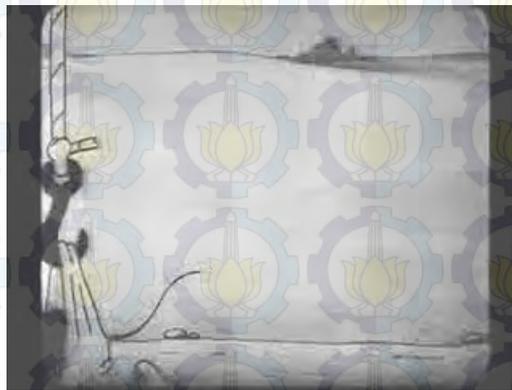


(d) Histogram frame ke-2 hasil *denoised* dari video input "popeye"

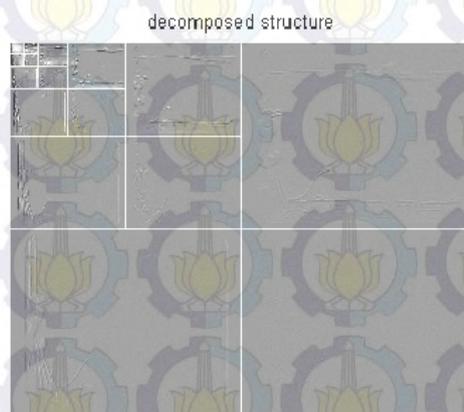
Gambar 4.18: Frame kedua sebelum di *denoised* dan frame histogram dari video input "popeye"

4.7 Pengujian *Discrete Wavelet Transform* (DWT) video input ”‘Steamboat’”

Proses dekomposisi 5 level dilakukan setelah frame asli menjadi frame *Gaussian noise*, melalui proses DWT dimana data dua dimensi diganti dengan empat blok yang bersesuaian dengan subband yang mewakili *low pass* filtering dan *high pass* filtering di setiap arah. Pertama, dilakukan transformasi pada semua baris yang menghasilkan matrik, dimana sisi kiri berisi koefisien *low pass down sample* dari setiap baris, dan sisi kanan berisi koefisien *high pass*. Kemudian dekomposisi diterapkan untuk semua kolom.

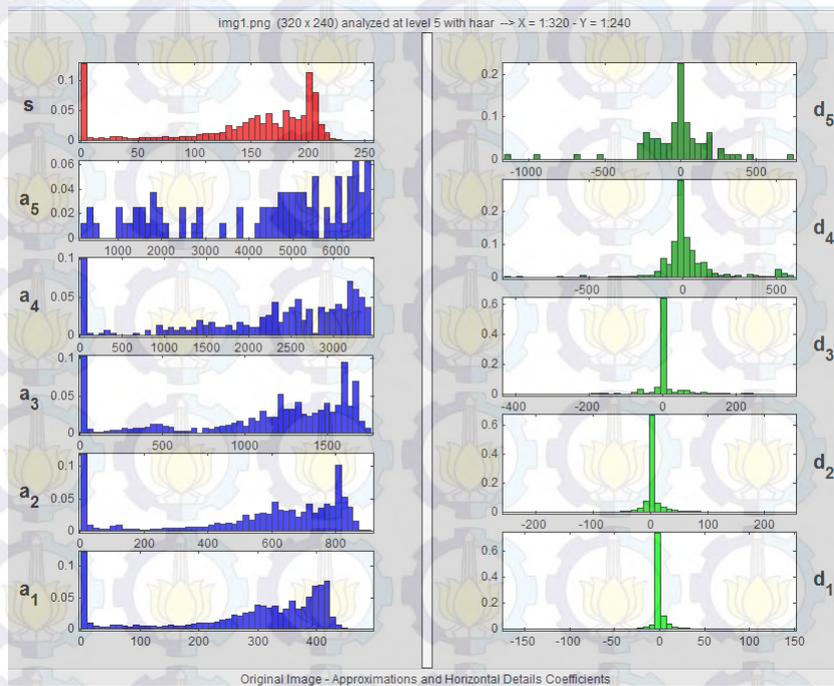


Gambar 4.19: Frame ke-1 dari video input ”‘Steamboat’”



Gambar 4.20: Frame ke-1 dari video input ”‘Steamboat’” didekomposisi sampai level 5

Dari proses DWT hasil yang didapatkan adalah koefisien-koefisien wavelet berupa subband aproksimasi (LL1), subband diagonal(LH1), subband vertical (HL1) dan subband horizontal (HH1) dari video input ”‘steamboat’”.



Gambar 4.21: Frame pertama dan Histogram koefisien aproksimasi dan koefisien detail dari hasil dekomposisi 5 level wavelet Haar dari video input ”‘Steamboat’”

Kedua proses tersebut dilakukan sebanyak lima kali, terhadap baris dan terhadap kolom sehingga diperoleh 13 subband keluaran. Pada Gambar 4.20 diatas adalah hasil dari proses dekomposisi level 5 yang dilakukan pada subband aproksimasi hasil dekomposisi dari proses sebelumnya dimana hasilnya berupa 1 subband diagonal, vertical dan horizontal yang disebut sebagai berupa fitur-fitur dari koefisien detail dan subband LL adalah koefisien aproksimasi.

4.8 Pengujian Tresholding koefisien detail wavelet level 5 video input ”‘Steamboat’”

Parameter dari σ_x adalah standar deviasi dan β adalah bentuk parameter yang telah diamati bahwa dengan parameter β mulai dari 0 sampai 0.01, dapat menggambarkan distribusi pada koefisien dalam subband diatur untuk frame yang besar.

Dengan mengasumsikan distribusi tersebut untuk koefisien wavelet, telah diperkirakan secara empiris β dan σ_x untuk setiap subband koefisien detail yang telah diperoleh dijadikan masukan pada proses tresholding dimana nilai tengah dari subband koefisien detail (d_5) di bagi dengan nilai 0.674 untuk

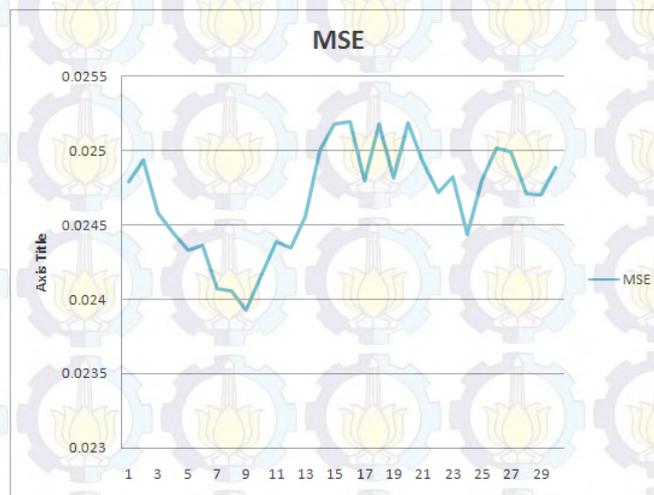
Tabel 4.2: Pengujian Tresholding koefisien detail wavelet level 5 video input
”‘Steamboat”’

Frame ke-	Level-1	Level-2	Level-3	Level-4	Level-5	MSE
Frame ke 1	0.0427	0.461	0.9839	2.1992	5.2979	0.0197
Frame ke 2	0.0353	0.3838	0.811	1.8058	5.2831	0.0205
Frame ke 3	0.0376	0.4072	1.0161	2.4404	5.8069	0.0185
Frame ke 4	0.0398	0.3905	0.9771	2.4093	5.3164	0.0185
Frame ke 5	0.0426	0.4246	1.1353	2.9402	5.4681	0.0175
Frame ke 6	0.039	0.4268	1.1424	3.0199	5.4871	0.0174
Frame ke 7	0.039	0.425	1.1257	3.0081	5.5926	0.0174
Frame ke 8	0.0316	0.3372	0.9561	2.676	5.5728	0.0174
Frame ke 9	0.0363	0.418	1.1137	3.1822	5.4057	0.0166
Frame ke 10	0.0369	0.3282	0.8955	2.1059	5.3166	0.0208
Frame ke 11	0.0426	0.4062	1.0937	2.5689	5.3963	0.0202
Frame ke 12	0.0415	0.394	1.0381	2.5422	5.3523	0.0203
Frame ke 13	0.0415	0.3914	1.0246	2.5384	5.3439	0.0202
Frame ke 14	0.0451	0.3689	1.0088	2.4766	5.2651	0.0208
Frame ke 15	0.044	0.3811	1.0199	2.3801	5.3003	0.0209
Frame ke 16	0.0422	0.3791	0.9952	2.3069	5.1415	0.0214
Frame ke 17	0.042	0.3802	0.9838	2.2695	5.0333	0.0215
Frame ke 18	0.0394	0.3425	0.9297	2.1925	4.8818	0.0217
Frame ke 19	0.0393	0.3433	0.9318	2.1881	4.9239	0.0217
Frame ke 20	0.0411	0.3445	0.9176	2.1695	4.9603	0.0221
Frame ke 21	0.043	0.3252	0.9668	2.1117	4.6144	0.022
Frame ke 22	0.0405	0.3203	0.9545	2.181	4.7388	0.0218
Frame ke 23	0.0412	0.327	0.8451	2.1509	4.5101	0.0226
Frame ke 24	0.043	0.3353	0.8504	2.1028	4.7751	0.0223
Frame ke 25	0.043	0.3344	0.8474	2.1024	4.7741	0.0224
Frame ke 26	0.0437	0.3247	0.8796	2.1107	4.6069	0.0227
Frame ke 27	0.0416	0.3212	0.858	2.1241	4.5855	0.0225
Frame ke 28	0.0416	0.3212	0.858	2.1241	4.5855	0.0232
Frame ke 29	0.0393	0.3252	0.7996	2.0412	4.26	0.0232
Frame ke 30	0.0409	0.3251	0.8044	2.0249	4.2427	0.0234

mendapatkan nilai *threshold* dan menemukan nilai *T* threshold untuk data input "pop-eye" adalah 8.1542 yang meminimalkan resiko Bayesian dengan menganalisa dari hasil proses *mean square error*.

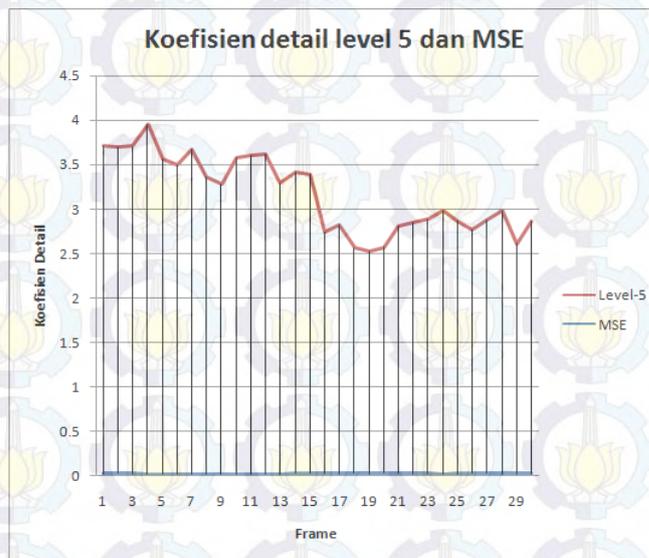


Gambar 4.22: Grafik Hasil pengujian threshold level 5 dari video input "Steamboat"



Gambar 4.23: Grafik Hasil pengujian MSE dari koefisien detail level 5 dari video input "Steamboat"

Tabel 4.2 menunjukkan nilai *threshold* dan *mean square error* dilihat dari grafik pada Gambar 4.22 dan Gambar 4.23 dapat dianalisa bahwa nilai *threshold* dari setiap subband pada koefisien detail antar frame memiliki tingkat kecenderungan perubahan intensitas derajat keabuan yang tidak jauh berbeda antar framenya. Perbedaan intensitas derajat keabuan pada antar



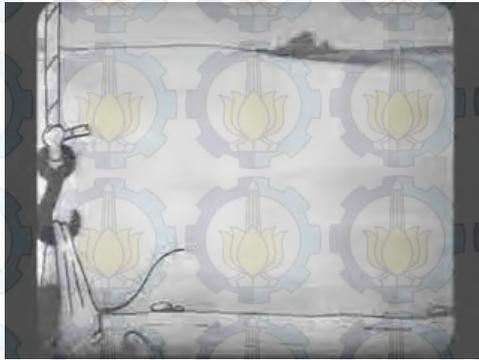
Gambar 4.24: Grafik Hasil pengujian koefisien detail level 5 dan MSE dari video input “Steamboat”

frame hasil tresholding menggunakan bayesshrink mengalami pengurangan *flicker (noise)* yang baik antar framanya yang ditunjukkan oleh dari Gambar 4.24.

4.9 Pengujian *inverse discrete wavelet transform (IDWT)*

Proses IDWT (*inverse discrete wavelet transform*) dilakukan setelah proses tresholding pada koefisien detail yang kemudian di rekontruksi atau menggabungkan semua subband dari hasil dekomposisi yaitu dengan pproses up-sampling dan pemfilteran dengan koefisien-koefisien filter balik. Proses ini dilakukan dengan menyisipkan sebuah kolom yang berharga nol diantara setiap kolom dan melakukan konvolusi pada setiap baris dengan filter satu dimensi. Penyisipan juga dilakukan pada sebuah baris yang berharga nol diantara setiap kolom dan melakukan konvolusi pada setiap kolom dengan filter yang lain.

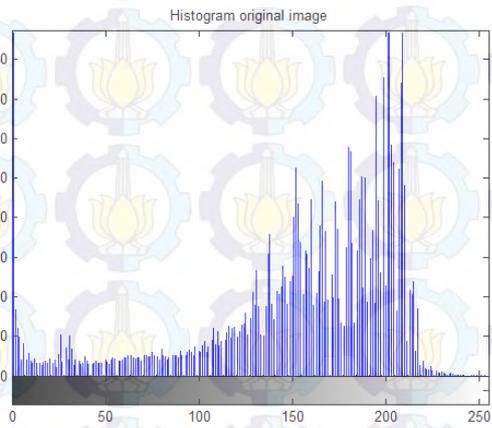
Pada Gambar 4.25 dab Gambar 4.26 menunjukkan frame pertama dan kedua sebelum dan sesudah terjadi proses *invers wavelet transform* atau disebut sebagai proses *denoised* dan histogram hasil dekomposisi citra yang sebelum diberikan noise (*Gaussian noise*) dapat terlihat dengan adanya perbedaan intensitas derajat keabuan pada citra asli dan citra yang telah di *denoised*. Perbandingan citra asli dan citra yang telah di denoised terlihat pada intensitas derajat keabuan antar frame yang dapat dianalisa bahwa pengurangan noise (*flicker*) pada antar frame menggunakan wavelet *tresh-*



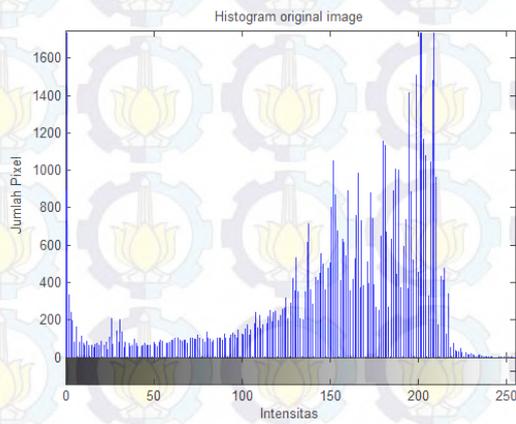
(a) Frame ke-1 dari video input "Steamboat"



(b) Frame ke-1 hasil *denoised* dari video input "Steamboat"

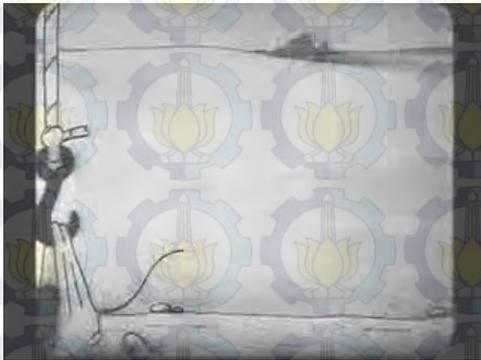


(c) Histogram frame ke-1 dari video input "Steamboat"



(d) Histogram frame ke-1 hasil *denoised* dari video input "Steamboat"

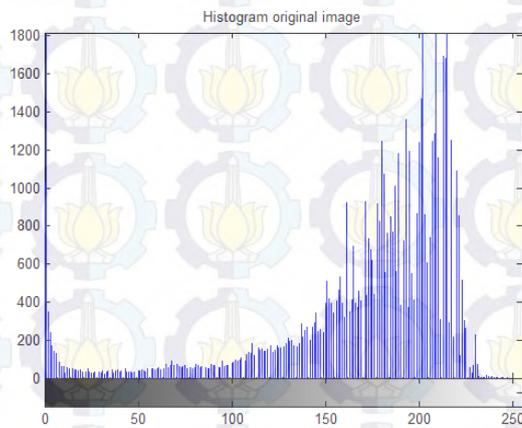
Gambar 4.25: Frame pertama sebelum di *denoised* dan frame hasil *denoised* dan histogram dari video input "Steamboat"



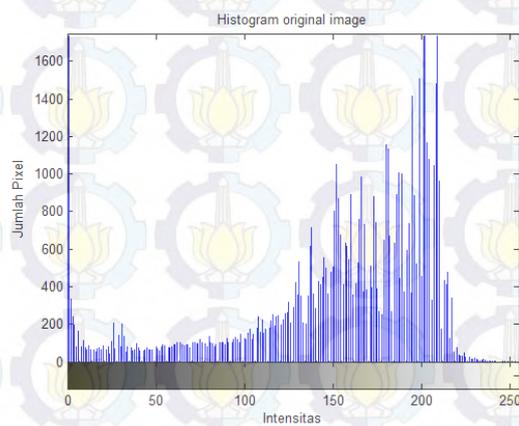
(a) Frame ke-2 dari video input "Steamboat"



(b) Frame ke-2 hasil *denoised* dari video input "Steamboat"



(c) Histogram frame ke-2 dari video input "Steamboat"



(d) Histogram frame ke-2 hasil *denoised* dari video input "Steamboat"

Gambar 4.26: Frame kedua sebelum di *denoised* dan frame histogram dari video input "Setamboot"

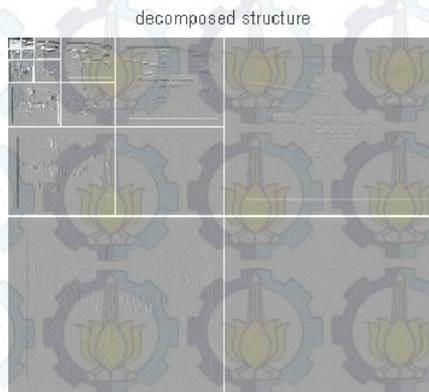
olding dengan teknik bayesshrink memberikan hasil yang baik.

4.10 Pengujian *Discrete Wavelet Transform* (DWT) video input ”‘Felix’”

Proses *discrete wavelet transform* (DWT) dengan melakukan dekomposisi 5 level pada frame asli menjadi frame *Gaussian noise* untuk data input ”‘Felix’” Gambar 4.27 dimana data dua dimensi diganti dengan empat blok yang bersesuaian dengan subband yang mewakili *low pass* filtering dan *high pass* filtering di setiap arah.



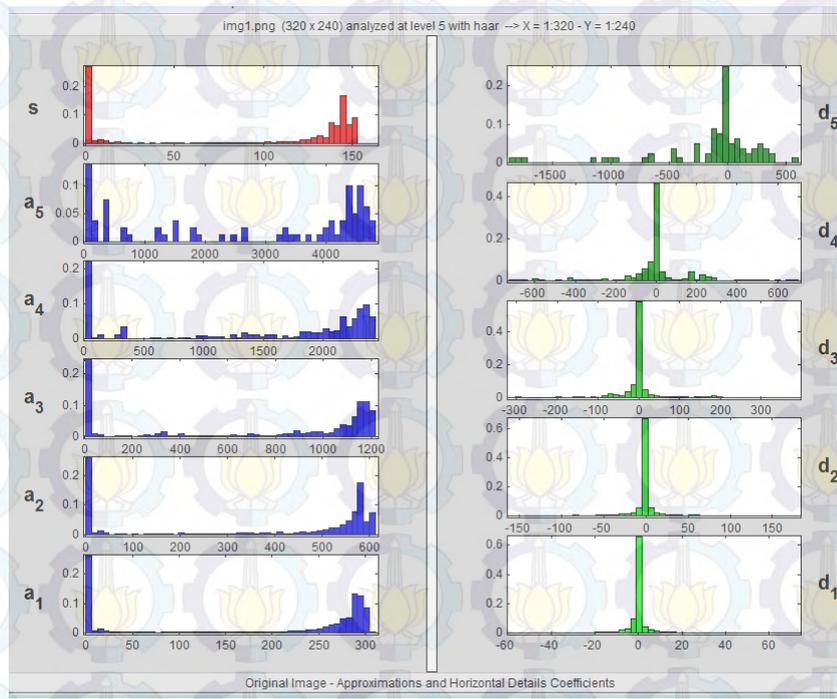
Gambar 4.27: Frame ke-1 dari video input ”‘Felix’”



Gambar 4.28: Frame ke-1 dari video input ”‘Felix’” didekomposisi sampai level 5

Pertama, dilakukan transformasi pada semua baris yang menghasilkan matrik, dimana sisi kiri berisi koefisien *low pass down sample* dari setiap baris, dan sisi kanan berisi koefisien *high pass*. Dekomposisi diterapkan untuk semua kolom seperti pada Gambar 4.28. Dari proses DWT hasil yang didapatkan adalah koefisien-koefisien wavelet berupa subband aproksimasi (LL1), subband

diagonal(LH1), subband vertical (HL1) dan subband horizontal (HH1) dari video input "Felix". Kedua proses tersebut dilakukan sebanyak lima kali, terhadap baris dan terhadap kolom sehingga diperoleh 13 subband keluaran. Pada Gambar 4.28 diatas adalah hasil dari proses dekomposisi level 5 yang dilakukan pada subband aproksimasi hasil dekomposisi dari proses sebelumnya dimana hasilnya berupa 1 subband diagonal, vertical dan horizontal yang disebut sebagai berupa fitur-fitur dari koefisien detail dan subband LL adalah koefisien aproksimasi Gambar 4.29.



Gambar 4.29: Frame pertama dan Histogram koefisien aproksimasi dan koefisien detail dari hasil dekomposisi 5 level wavelet Haar dari video input "Felix"

4.11 Pengujian Tresholding koefisien detail wavelet level 5 video input "Felix"

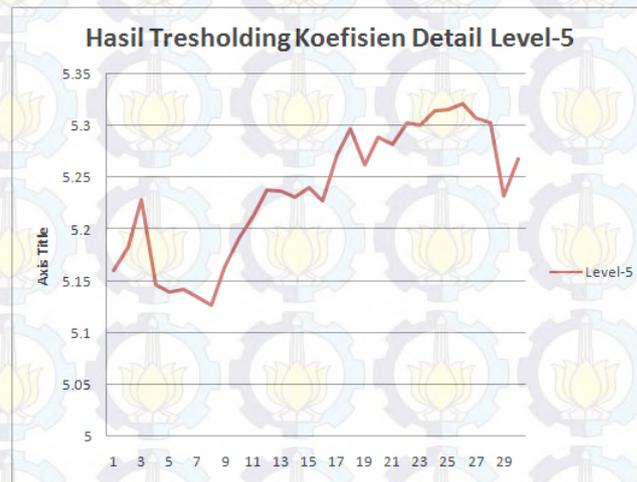
Parameter dari σ_x adalah standar deviasi dan β adalah bentuk parameter yang telah diamati bahwa dengan parameter β mulai dari 0 sampai 0.01, dapat menggambarkan distribusi pada koefisien dalam subband diatur untuk frame yang besar.

Dengan mengasumsikan distribusi tersebut untuk koefisien wavelet, telah diperkirakan secara empiris β dan σ_x untuk setiap subband koefisien detail yang telah diperoleh dijadikan masukan pada proses tresholding dimana nilai

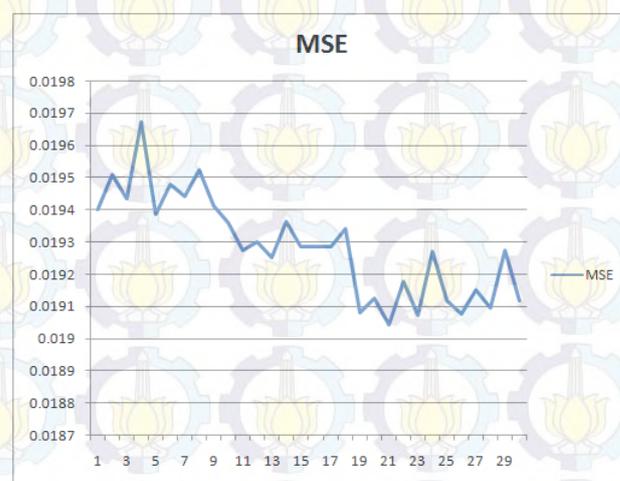
Tabel 4.3: Pengujian Tresholding koefisien detail wavelet level 5 video input ”Felix”

Frame ke-	Level-1	Level-2	Level-3	Level-4	Level-5	MSE
Frame ke 1	0.03921	0.34147	0.8055	1.6065	5.1607	0.0194
Frame ke 2	0.03918	0.34161	0.80029	1.6032	5.1819	0.01951
Frame ke 3	0.03902	0.34273	0.79231	1.6241	5.2288	0.01943
Frame ke 4	0.03865	0.34142	0.71128	1.5655	5.1464	0.01967
Frame ke 5	0.03941	0.3396	0.70467	1.5348	5.139	0.01939
Frame ke 6	0.03974	0.3345	0.73181	1.5678	5.1413	0.01948
Frame ke 7	0.04017	0.33022	0.73877	1.5564	5.1344	0.01944
Frame ke 8	0.03986	0.34386	0.74716	1.5973	5.1268	0.01952
Frame ke 9	0.03969	0.3433	0.76186	1.6216	5.1643	0.01941
Frame ke 10	0.03933	0.34018	0.75515	1.6088	5.1918	0.01936
Frame ke 11	0.03936	0.36474	0.74464	1.6466	5.2134	0.01927
Frame ke 12	0.03936	0.36755	0.74214	1.6524	5.2382	0.0193
Frame ke 13	0.03957	0.38873	0.72592	1.6405	5.2368	0.01925
Frame ke 14	0.03955	0.38445	0.71432	1.6437	5.2304	0.01936
Frame ke 15	0.03957	0.37571	0.68537	1.5964	5.2406	0.01929
Frame ke 16	0.03884	0.37017	0.72602	1.6364	5.2277	0.01928
Frame ke 17	0.03881	0.37726	0.73611	1.6492	5.2717	0.01929
Frame ke 18	0.03873	0.38166	0.74484	1.6729	5.297	0.01934
Frame ke 19	0.03859	0.38084	0.77267	1.6867	5.2616	0.01908
Frame ke 20	0.03822	0.38868	0.7699	1.7213	5.2887	0.01912
Frame ke 21	0.03767	0.40275	0.75702	1.6747	5.2819	0.01904
Frame ke 22	0.03824	0.38782	0.74427	1.6695	5.3023	0.01918
Frame ke 23	0.03886	0.3834	0.73155	1.637	5.3003	0.01907
Frame ke 24	0.03809	0.39498	0.70985	1.5896	5.3142	0.01927
Frame ke 25	0.03833	0.38589	0.7457	1.6425	5.3158	0.01912
Frame ke 26	0.03859	0.36912	0.75291	1.6706	5.3211	0.01908
Frame ke 27	0.03915	0.37626	0.7337	1.5966	5.3077	0.01915
Frame ke 28	0.03882	0.37963	0.72967	1.5828	5.3021	0.0191
Frame ke 29	0.03838	0.38905	0.73138	1.6062	5.2322	0.01927
Frame ke 30	0.03886	0.38745	0.73521	1.6072	5.2681	0.01911

tengah dari subband koefisien detail (d_5) di bagi dengan nilai 0.674 untuk mendapatkan nilai *threshold* dan menemukan nilai T treshold untuk data input "popeye" adalah 7.4239 yang meminimalkan resiko Bayesian dengan menganalisa dari hasil proses *mean square error*.

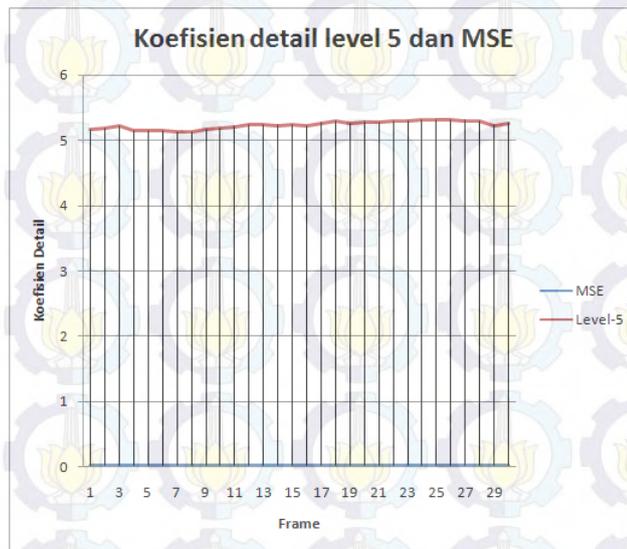


Gambar 4.30: Grafik Hasil pengujian threshold level 5 dari video input "Felix"



Gambar 4.31: Grafik Hasil pengujian MSE dari koefisien detail level 5 dari video input "Felix"

Tabel 4.1 menunjukkan nilai *threshold* dan *mean square error* dilihat dari grafik pada Gambar 4.30 dan Gambar 4.31 dapat dianalisa bahwa nilai *threshold* dari setiap subband pada koefisien detail antar frame memiliki tingkat kecenderungan perubahan intensitas derajat keabuan yang tidak jauh berbeda antar framanya. Perbedaan intensitas derajat keabuan pada antar



Gambar 4.32: Grafik Hasil pengujian koefisien detail level 5 dan MSE dari video input “Felix”

frame hasil tresholding menggunakan bayesshrink mengalami pengurangan flicker (noise) yang baik antar framenya yang ditunjukkan oleh dari Gambar 4.32.

4.12 Pengujian *inverse discrete wavelet transform (IDWT)*

Proses IDWT (*inverse discrete wavelet transform*) dilakukan setelah proses tresholding pada koefisien detail yang kemudian di rekontruksi atau menggabungkan semua subband dari hasil dekomposisi yaitu dengan pproses up-sampling dan pemfilteran dengan koefisien-koefisien filter balik. Proses ini dilakukan dengan menyisipkan sebuah kolom yang berharga nol diantara setiap kolom dan melakukan konvolusi pada setiap baris dengan filter satu dimensi. Penyisipan juga dilakukan pada sebuah baris yang berharga nol diantara setiap kolom dan melakukan konvolusi pada setiap kolom dengan filter yang lain.

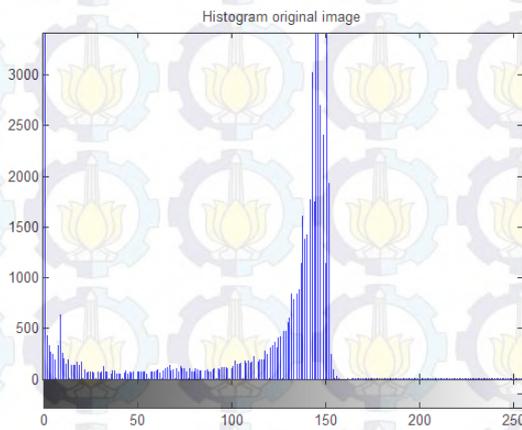
Pada Gambar 4.33 dab Gambar 4.34 menunjukkan frame pertama dan kedua sebelum dan sesudah terjadi proses *invers wavelet transform* atau disebut sebagai proses *denoised* dan histogram hasil dekomposisi citra yang sebelum diberikan noise (*Gaussian noise*) dapat terlihat dengan adanya perbedaan intesitas derajat keabuan pada citra asli dan citra yang telah di *denoised*. Perbandingan citra asli dan citra yang telah di denoised terlihat pada intensitas derajat keabuan antar frame yang dapat dianalisa bahwa pengurangan noise (*flicker*) pada antar frame menggunakan wavelet *tresholding* dengan teknik bayesshrink memberikan hasil yang baik.



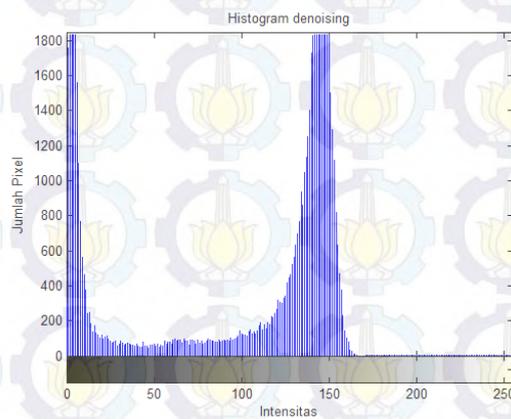
(a) Frame ke-1 dari video input "Felix"



(b) Frame ke-1 hasil *denoised* dari video input "Felix"



(c) Histogram frame ke-1 dari video input "Felix"

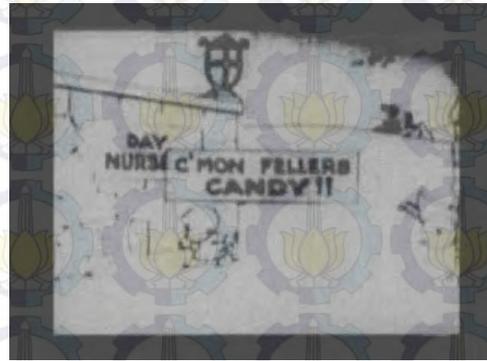


(d) Histogram frame ke-1 hasil denoised dari video input "Felix"

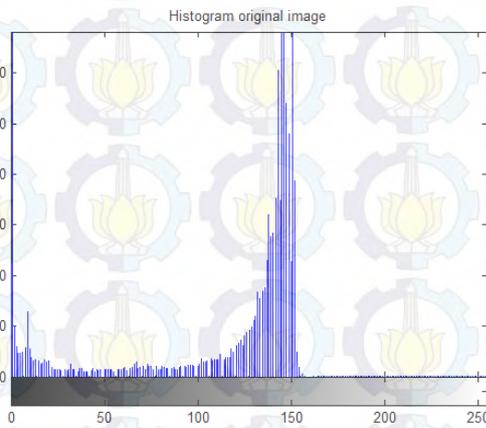
Gambar 4.33: Frame pertama sebelum di *denoised* dan frame hasil denoised dan histogram dari video input "Felix"



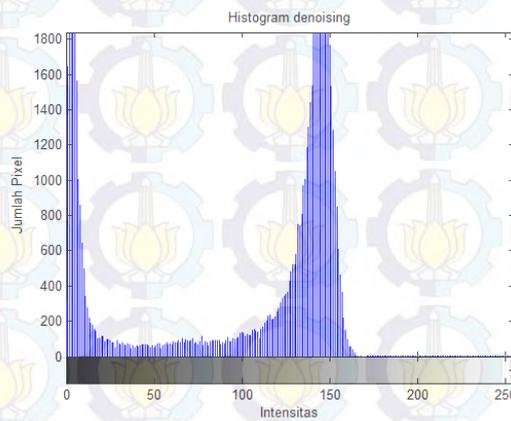
(a) Frame ke-2 dari video input "Felix"



(b) Frame ke-2 hasil *denoised* dari video input "Felix"



(c) Histogram frame ke-2 dari video input "Felix"



(d) Histogram frame ke-2 hasil *denoised* dari video input "Felix"

Gambar 4.34: Frame kedua sebelum di *denoised* dan frame histogram dari video input "Felix"

4.13 Hasil pengurangan Flicker Pada Video animasi Kuno

Pengukuran kualitas frame hasil denoised adalah proses akhir dari penelitian ini untuk dapat mengetahui berapa persentase berkurangnya noise(flicker) antar frame pada setiap data input ”‘popeye’’,”’Steamboat”’ dan ”‘Felix”’ menggunakan teknik *Peak Signal to Noise Ratio* (PSNR).

Tabel 4.4: Tabel Pengukuran Kualitas Frame hasil Denoised Dengan PSNR

Frame ke-	popeye		Steamboat		Felix	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
Frame ke 1	0.0248	45.474	0.0197	46.196	0.0194	45.351
Frame ke 2	0.0249	45.468	0.0205	45.997	0.0195	45.254
Frame ke 3	0.0246	45.496	0.0185	46.247	0.0194	45.329
Frame ke 4	0.0244	45.348	0.0185	46.239	0.0197	45.226
Frame ke 5	0.0243	45.348	0.0175	46.363	0.0194	45.447
Frame ke 6	0.0244	45.547	0.0174	46.381	0.0195	45.29
Frame ke 7	0.0241	45.56	0.0174	46.31	0.0194	45.421
Frame ke 8	0.0241	45.529	0.0174	46.289	0.0195	45.393
Frame ke 9	0.0239	45.618	0.0166	46.296	0.0194	45.296
Frame ke 10	0.0242	45.537	0.0208	45.844	0.0194	45.295
Frame ke 11	0.0244	45.452	0.0202	45.945	0.0193	45.288
Frame ke 12	0.0243	45.403	0.0203	45.783	0.0193	45.373
Frame ke 13	0.0246	45.493	0.0202	45.988	0.0193	45.342
Frame ke 14	0.025	45.433	0.0208	45.872	0.0194	45.227
Frame ke 15	0.0252	45.342	0.0209	45.869	0.0193	45.337
Frame ke 16	0.0252	45.434	0.0214	45.839	0.0193	45.263
Frame ke 17	0.0248	45.464	0.0215	45.782	0.0193	45.308
Frame ke 18	0.0252	45.414	0.0217	45.844	0.0193	45.333
Frame ke 19	0.0248	45.437	0.0217	45.763	0.0191	45.327
Frame ke 20	0.0252	45.42	0.0221	45.665	0.0191	45.341
Frame ke 21	0.0249	45.33	0.022	45.629	0.019	45.433
Frame ke 22	0.0247	45.431	0.0218	45.742	0.0192	45.279
Frame ke 23	0.0248	45.41	0.0226	45.717	0.0191	45.372
Frame ke 24	0.0244	45.468	0.0223	45.684	0.0193	45.339
Frame ke 25	0.0248	45.417	0.0224	45.683	0.0191	45.482
Frame ke 26	0.025	45.395	0.0227	45.481	0.0191	45.373
Frame ke 27	0.025	45.372	0.0225	45.565	0.0192	45.408
Frame ke 28	0.0247	45.445	0.0232	45.532	0.0191	45.359
Frame ke 29	0.0247	45.437	0.0232	45.608	0.0193	45.338
Frame ke 30	0.0249	45.498	0.0234	45.493	0.0191	45.333

Dari seluruh data ujicoba yang telah didenoised didapatkan hasil PSNR yang ditunjukkan pada Tabel 4.4. Dari Tabel 4.4 dapat dilihat pada video

input "popeye" nilai MSE tertinggi adalah 0.02518 dan nilai MSE terendah adalah 0.02392 untuk nilai PSNR tertinggi adalah 45.4977 dan nilai PSNR terendah 45.3398. Data input "Steamboat" menunjukkan nilai mSE tertinggi 0.02319 dan nilai MSE terendah adalah 0.0173, nilai PSNR tertinggi 45.9974 dan nilai PSNR terendah adalah 46.1964. Data Input "Felix" memiliki nilai MSE tertinggi adalah 0.01967 dan nilai MSE terendah adalah 0.01908 sedangkan nilai PSNR tertinggi adalah 45.4822 dan nilai PSNR terendah adalah 45.2625.

BAB V

KESIMPULAN DAN PENELITIAN SELANJUTNYA

5.1 Kesimpulan

Pada pengujian pengurangan (reduksi) *flicker* pada animasi kuno yang diasumsikan sebagai masalah *Gaussian Noise* menggunakan wavelet *thresholding* pada tiga video input "popaye", "steamboat" dan "felix" dengan teknik *bayesshrink* dan memilih *mother wavelet* Haar level 5 dapat diambil kesimpulan bahwa: Pengujian pada data uji coba hasil *convert* dari video dengan durasi 1 detik yang memiliki *framerate* 30 *frame/second* dengan membangkitkan *Gaussian Noise* pada antar frame yang selanjutnya pengurangan *flicker* antar frame dilakukan dengan teknik *thresholding bayesshrink* yang sebelumnya dilakukan proses transformasi pada koefisien wavelet untuk mendapatkan koefisien detail yang kemudian di *threshold* memberikan hasil yang baik. Dari hasil ujicoba pada data yang diujikan dengan metode wavelet *thresholding* didapatkan hasil pengurangan *flicker* yang baik dengan menghitung kualitas dari antar frame yang telah diuji menggunakan nilai MSE dan PSNR memberikan hasil pada video input "Popaye" nilai MSE adalah 0.02392 dan nilai PSNR tertinggi adalah 45.4977. Data input "Steamboat" menunjukkan nilai MSE adalah 0.0173 dan nilai PSNR tertinggi 45.9974. Data Input "Felix" memiliki nilai MSE adalah 0.01908 dan nilai PSNR tertinggi adalah 45.4822.

5.2 Penelitian Selanjutnya

Pada penelitian ini masih memiliki kekurangan untuk perbaikan kualitas citra, karena itu diperlukan penyempurnaan di masa mendatang, pada penelitian selanjutnya dapat menggunakan metode yang berbeda seperti *Logarithmic Image Processing* (LIP) dengan kontras biasa, *image brightness* dan perenggangan kontras atau spesifikasi yang berbeda.

DAFTAR PUSTAKA

A, P. dan Ray, S. S. (2014), 'Two-dimensional Haar wavelet Collocation Method for the solution of Stationary Neutron Transport Equation in a homogeneous isotropic medium', *Annals of Nuclear Energy* 70, 30-35 .

Andra, G., Iwut, Tritoasomoro, I. dan Jangkung, R. (2007), 'Analisa Perbandingan Reduksi Noise Pada Citra Antara Discrete Wavelet Transform (DWT) Dengan Dual-Tree Complex Wavelet Transform (DTCWT)', *Seminar Nasional Sistem dan Informatika, SNSI07-023*. .

Donoho, D. L. (1995), 'Denoising by soft-thresholding', *IEEE Transactions on Information Theory*, vol. 41, no.3, pp.613-627 .

Dwi Prihartono, T. d. d. (2011), 'Identifikasi Iris Mata Menggunakan Alihragam Wavelet Haar, TRANSMISI', 13(2), 71 - 75 .

E. Bire, Christa, d. (2012), 'Denoising Pada Citra Menggunakan Transformasi Wavelet', ISBN 979 - 26 - 0255 - 0 .

H.Sianapar, d. (2013), *Matlab Untuk Pemrosesan Citra Digital*, ISBN : 978-602-1514-11-5.

Khriji, L. dan Gabbouj, M. (2000), 'Multichannel Image Processing Using Fuzzy Vector Median-Rational Hybrid Filters', *X European Signal Processing Conf. Eusipco, Tampere, Finland, 4-11*, .

Kuszpet, Y. dan dkk (n.d.), 'Post Processing For Flicker Reduction In H.264/AVC', *Jurnal Generic, Vol.5 No.2*. .

Mahjud, I. (2010), 'Pengkodean Dan Transmisi Citra Digital Yang Terkompresi Dengan Wavelet.', *Volume 6, Nomor 1, ISSN : 1858-3709* .

Munir, R. (2006), 'Aplikasi Image Thresholding untuk Segmentasi Objek', *Seminar Nasional Aplikasi Teknologi Informasi, ISSN, 1907-5022* .

Pramiswari, Niken, A. d. P. Y. d. S. R. (2012), 'Implementasi Denoising Citra RGB Menggunakan Metode Wavelet Berbasis Logika Fuzzy', *Skripsi Teknik Informatika, Fakultas Teknologi Informasi*, .

Prihartono, T. D. (2011), 'identifikasi Iris Mata Menggunakan Alihragam Wavelet Haar', *TRANSMISI*, 13 (2), 71-75 .

Saraswati, N. W. S. (2010), 'Transformasi Wavelet Dan Thresholding Pada Citra Menggunakan Matlab,'.

Sedyono, Eko, d. (2009), 'Klasifikasi Sidik Jari Menggunakan Metode Wavelet Symlet', *Jurnal Informatika*, Vol. 5, No. 2, pp. 17-33. .

Vetterli S, M., Chang, G. dan Yu, B. (2000), 'Adaptive wavelet thresholding for image denoising and compression.', *IEEE Transactions on Image Processing*, 9 (9), 1532-1546 .

BAB VI LAMPIRAN

6.1 Lampiran I

6.1.1 Data Input Video "Popeye"



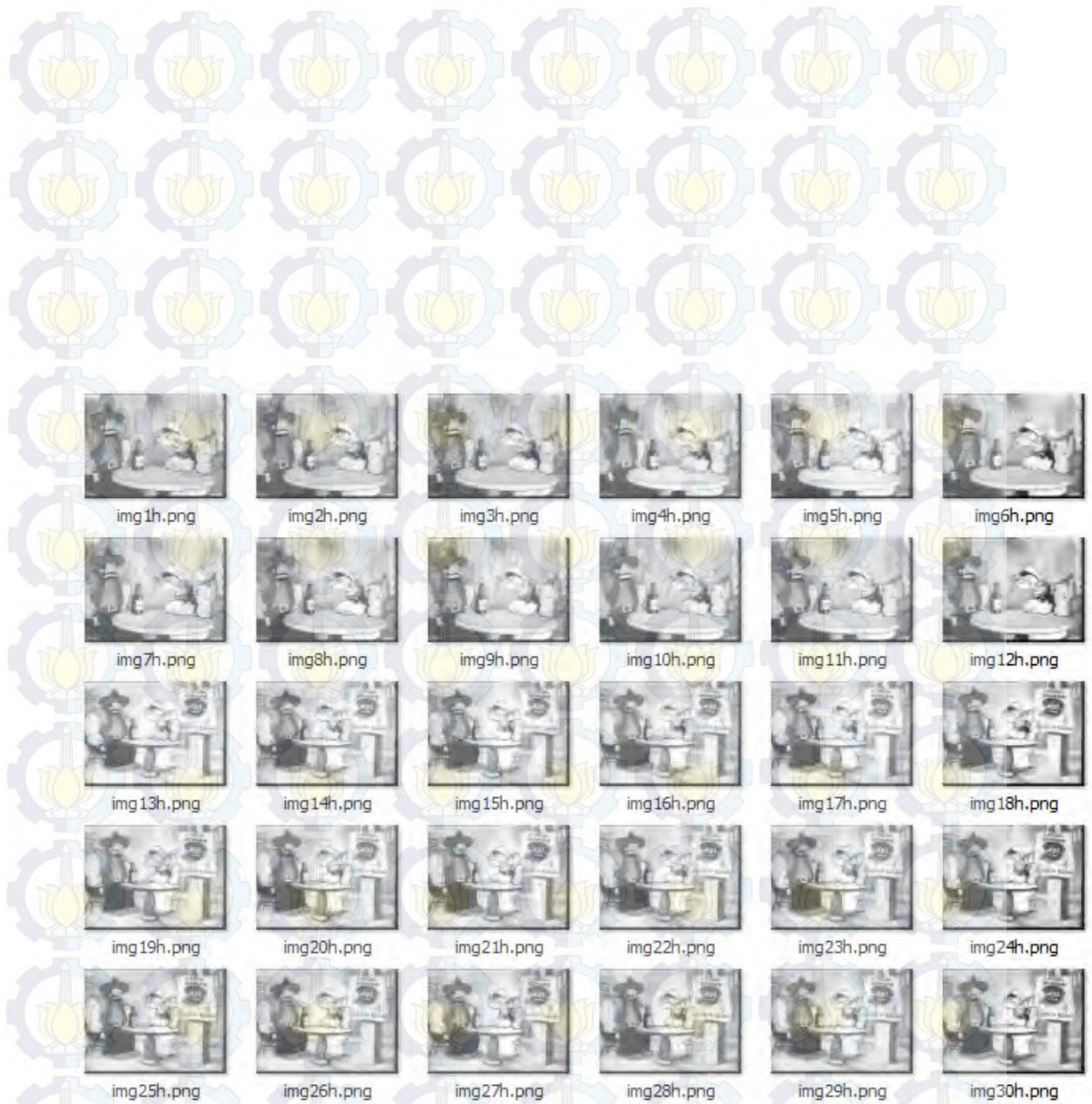
Gambar 6.1: Data input video "Popeye" yang di ekstrak menjadi 30 frame

6.1.2 Data Input Video "Steambot"

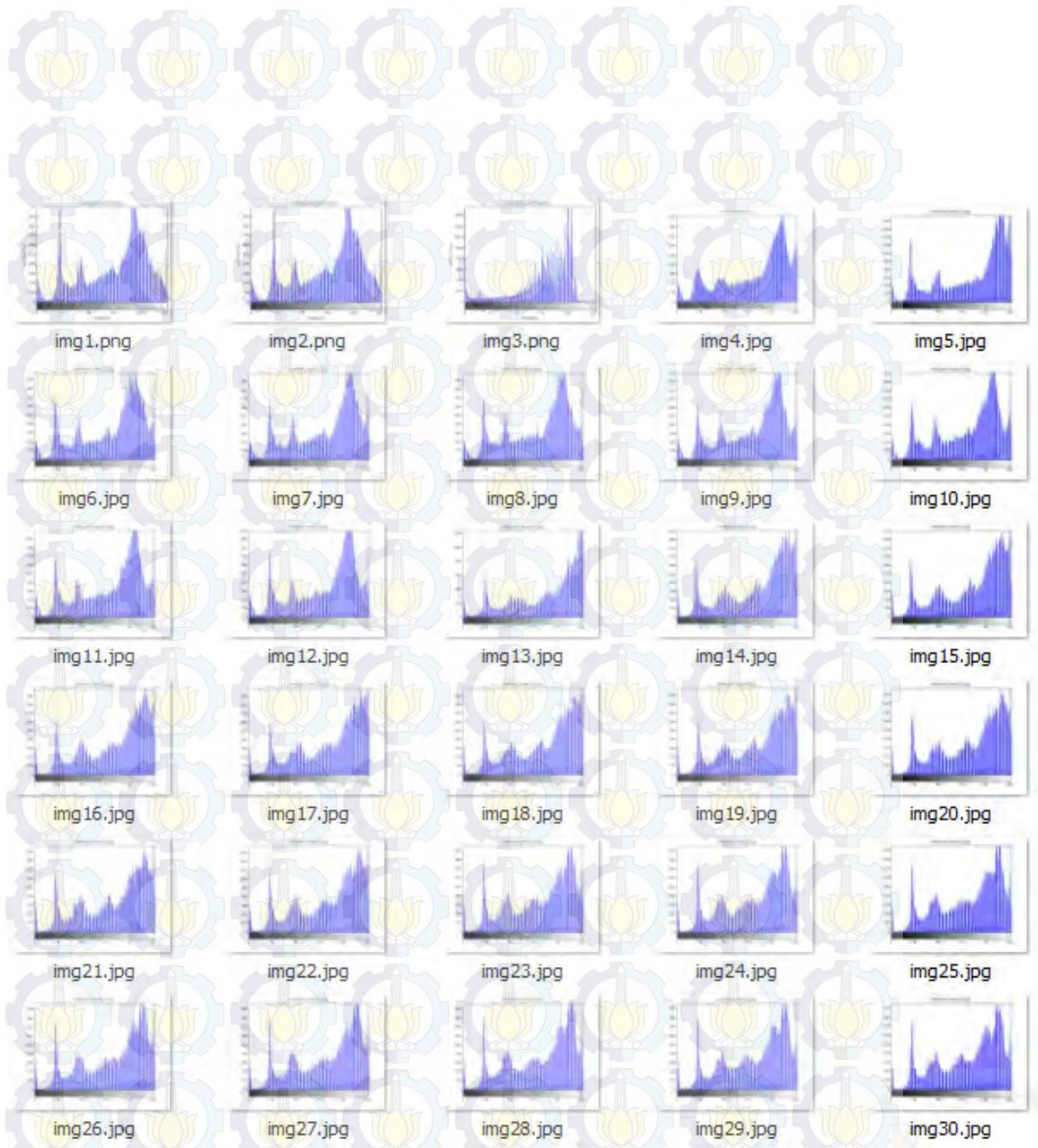
6.1.3 Data Input Video "Felix"



Gambar 6.2: Frame data input video "Popeye" hasil dari membangkitkan *Gaussian Noise*



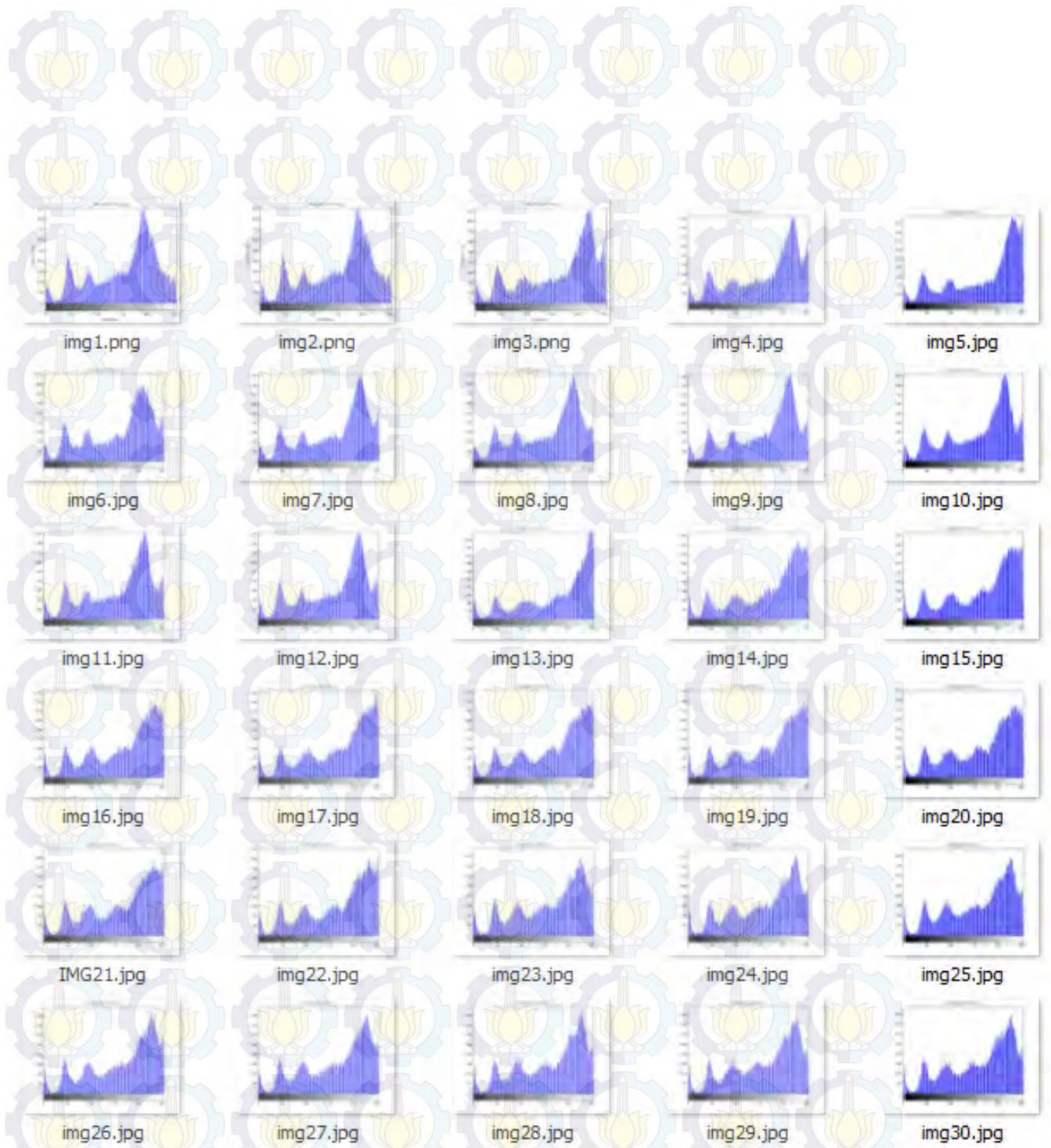
Gambar 6.3: Frame data input vide” Popeye ” hasil Denoised



Gambar 6.4: Histogram dari frame data input video ”‘Popeye”’



Gambar 6.5: Histogram frame data input vide” Popeye ” hasil dari membangkitkan *Gaussian Noise*



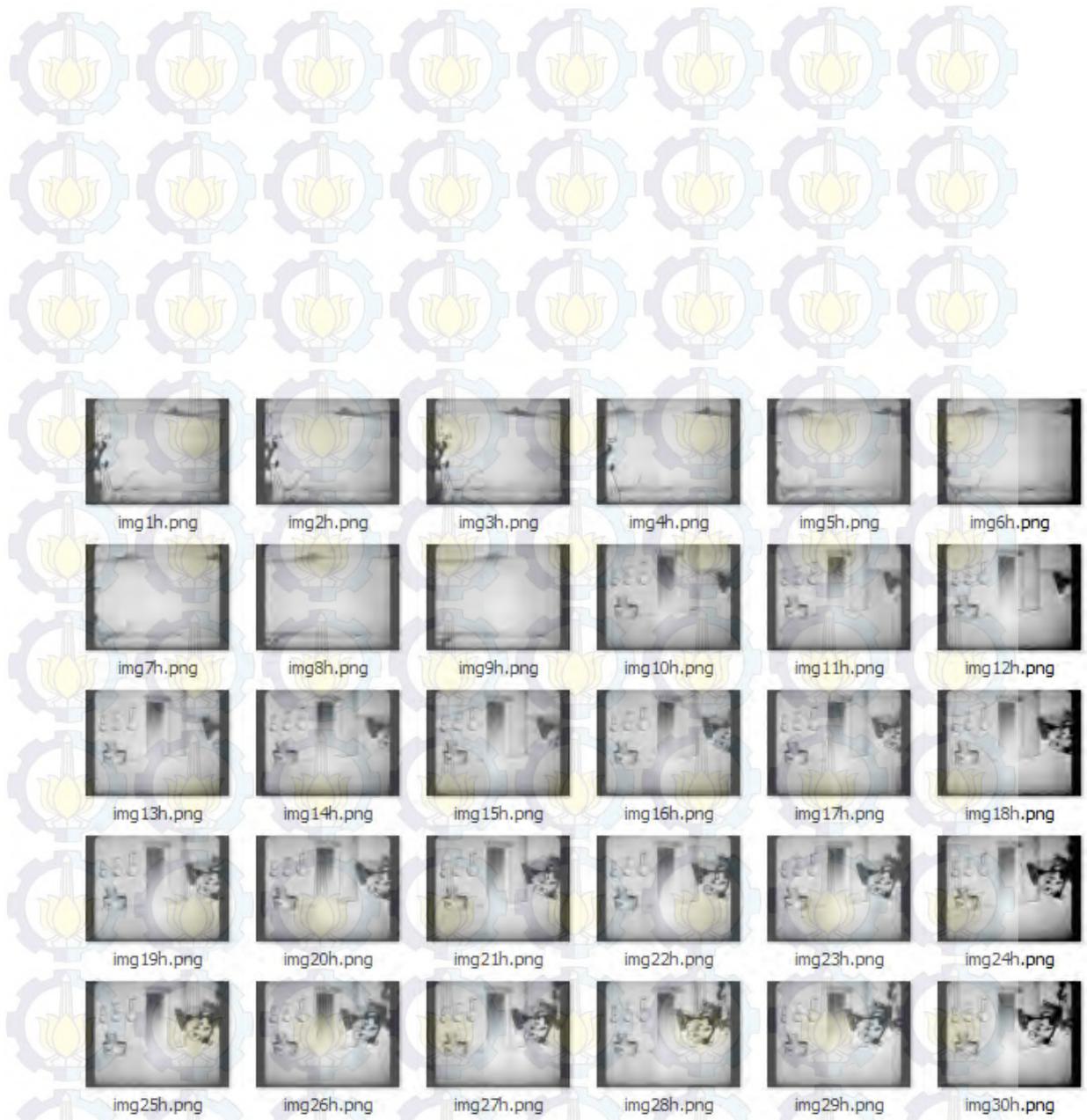
Gambar 6.6: Histogram frame data input vide” Popeye ” hasil Denoised



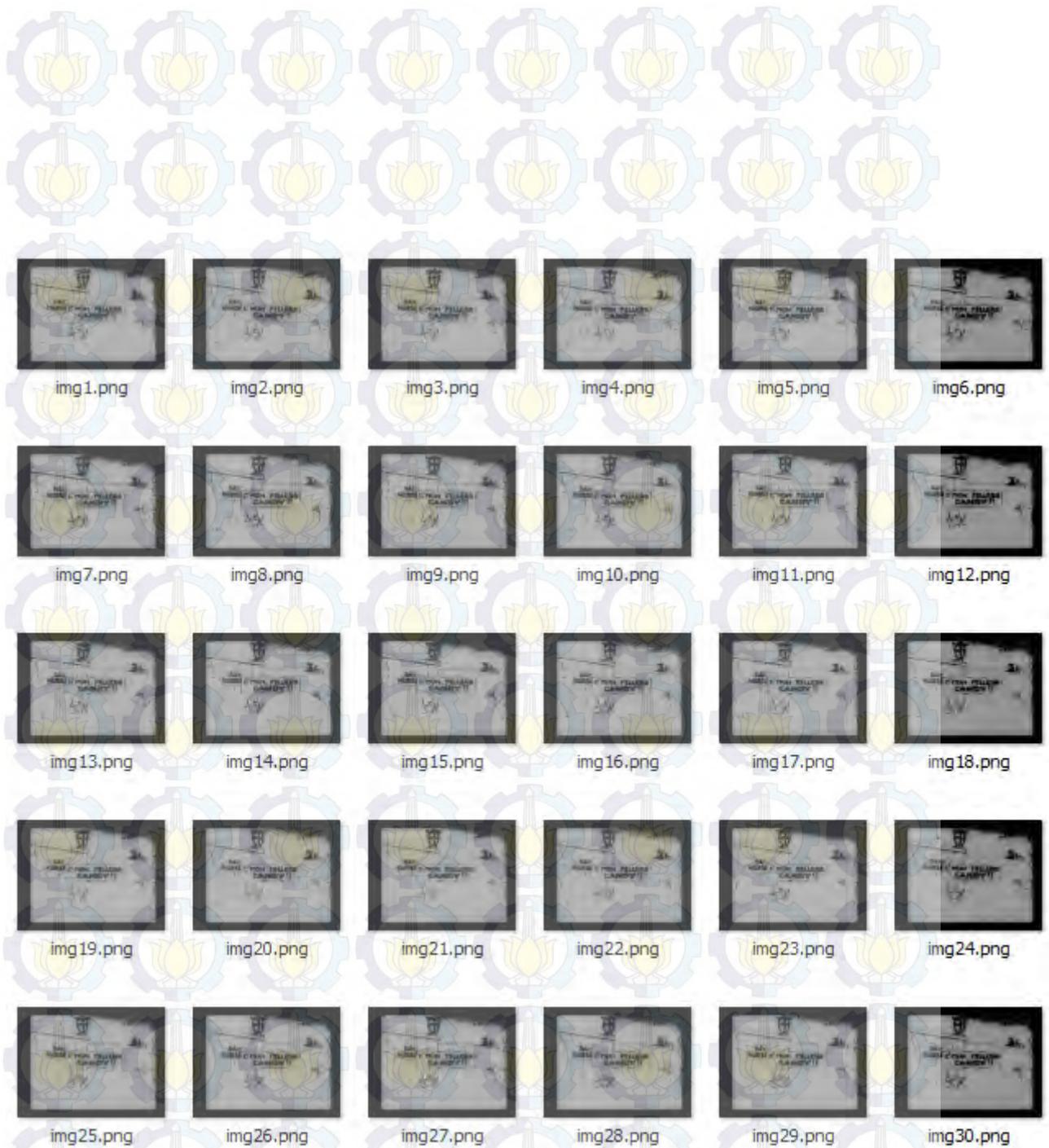
Gambar 6.7: Data input video ”‘Steamboat’” yang di ekstrak menjadi 30 frame



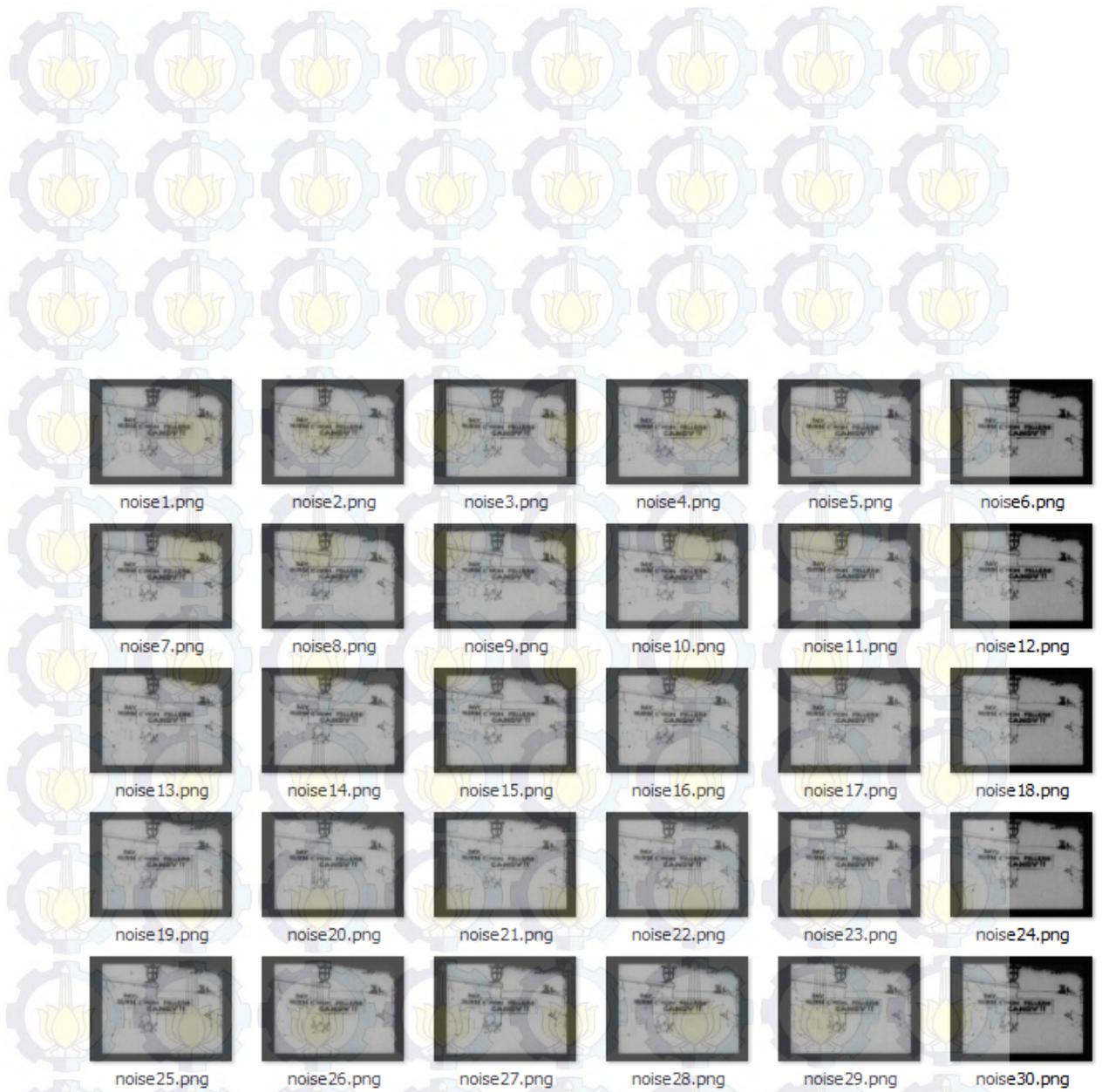
Gambar 6.8: Frame data input video "Steamboat" hasil dari membangkitkan *Gaussian Noise*



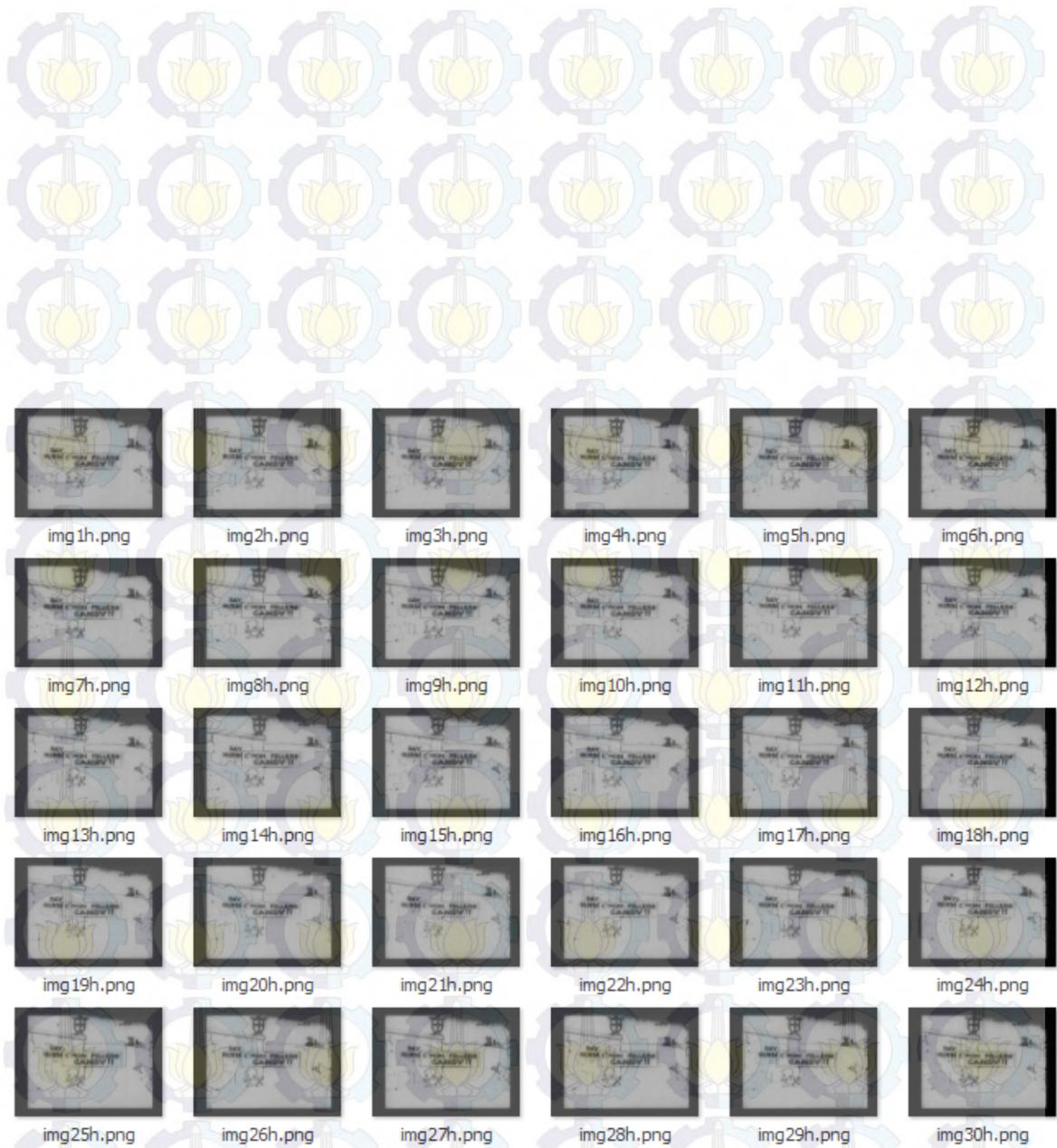
Gambar 6.9: Frame data input vide” Steamboat” hasil Denoised



Gambar 6.10: Data input video "Felix" yang di ekstrak menjadi 30 frame



Gambar 6.11: Frame data input video "Felix" hasil dari membangkitkan *Gaussian Noise*



Gambar 6.12: Frame data input vide "Felix" hasil Denoised

BIODATA



I. Data Pribadi

1. Nama : Desita Ria Yusian TB
2. Tempat/Tanggal Lahir : Lhokseumawe, 03 Desember 1989
3. Jenis Kelamin : Perempuan
4. Agama : Islam
5. Alamat : Perumahan Politeknik Blok 8 No.2
Kota Lhokseumawe, Propinsi Aceh
email : cytaria@gmail.com

II. Riwayat Pendidikan

Pendidikan Dasar dan Menengah

No.	Tingkat Pendidikan	Nama Sekolah	Lulus Tahun
1.	SD	SD Negeri Beunot Bayu, Lhokseumawe	2000
2.	SMP	SMP Negeri 5 Lhokseumawe	2003
3.	SMA	SMA Negeri 1 Lhokseumawe	2006

Pendidikan Tinggi

No.	Perguruan Tinggi	Gelar	Lulus	Jurusan	Bidang Spesialisasi
1.	Politeknik Negeri Lhokseumawe	SST	2010	Elektro	Teknik Informatika

III. Riwayat Pekerjaan

Dosen Jurusan Teknik Informatika Politeknik Negeri Lhokseumawe, 2010-2011

IV. Karya Ilmiah dan Pelatihan

1. TB, Desita, Ria, Yusian, and Purnomo, Mauridhi. Hery, and Suprpto, Yoyon, Kusnendar. (2014) "Reduction Flicker in Old Animation Video Using Wavelet Thresholding", Seminar on Intelligent Technology and Its Applications Vol 15

Surabaya Juli 2014

Desita Ria Yusian TB