

**DENOISING PADA CITRA BER-ISO TINGGI DENGAN
MENGGUNAKAN KONSEP NEIGHBOURING WINDOW**

SKRIPSI

oleh :
ASRI WIJAYANTI
0610960010-96



PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012

UNIVERSITAS BRAWIJAYA



**DENOISING PADA CITRA BER-ISO TINGGI DENGAN
MENGGUNAKAN KONSEP NEIGHBOURING WINDOW**

SKRIPSI

**Sebagai salah satu syarat untuk memperoleh gelar sarjana
dalam bidang ilmu komputer**

oleh :

ASRI WIJAYANTI

0610960010-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN

DENOISING PADA CITRA BER-ISO TINGGI DENGAN MENGGUNAKAN KONSEP NEIGHBOURING WINDOW

oleh :
ASRI WIJAYANTI
0610960010-96

Setelah dipertahankan di depan Majelis Pengaji
Pada tanggal 6 Februari 2012

Dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana dalam bidang ilmu komputer

Pembimbing I

Pembimbing II

Drs. Muh. Arif Rahman, M. Kom
NIP. 196604231991111001

Drs. Marji, MT
NIP. 196708011992031001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya,

Dr. Abdul Rouf Algofari, M.Sc.
NIP. 196709071992203001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Asri Wijayanti
NIM : 0610960010
Jurusan : Matematika / Ilmu Komputer
Penulis Skripsi Berjudul : *Denoising Pada Citra Ber-Iso Tinggi Dengan Menggunakan Konsep Neighbouring Window*

Dengan ini menyatakan bahwa:

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila di kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 6 Februari 2012
Yang menyatakan,

(Asri Wijayanti)
NIM. 0610960010

UNIVERSITAS BRAWIJAYA



DENOISING PADA CITRA BER-ISO TINGGI DENGAN MENGGUNAKAN KONSEP NEIGHBOURING WINDOW

ABSTRAK

ISO merupakan ukuran tingkat kepekaan sensor cahaya. ISO tinggi berguna untuk mangambil gambar pada pencahayaan rendah seperti pada malam hari, kondisi mendung atau didalam ruangan. Namun penggunaan ISO yang terlalu tinggi dapat menyebabkan timbulnya *noise*. *Neighshrink* merupakan salah satu metode yang dapat digunakan dalam proses *denoising* dengan menggunakan konsep neighbouring window dengan ukuran $L \times L$, dimana L merupakan bilangan ganjil lebih dari 1. Pada penelitian ini citra yang digunakan adalah citra dengan ISO 1600 dengan ukuran *neighbouring window* 3x3 dan 5x5. Sedangkan untuk proses thresholding menggunakan nilai *universal threshold*, nilai *SURE threshold* serta nilai penjumlahan kuadrat tetangga. Dari hasil pengujian menunjukkan bahwa nilai *PSNR* paling tinggi diperoleh pada penggunaan *universal threshold* pada ukuran *neighbouring window* 5x5.

Kata kunci : ISO, *Neighshrink*, *neighbouring window*, *Universal threshold*, *SURE threshold*.

UNIVERSITAS BRAWIJAYA



DENOISING ON HIGH ISO IMAGE USING NEIGHBOURING WINDOW CONCEPT

ABSTRACT

ISO determines how responsive image sensor to light is. High ISO setting is useful for taking a picture in a darker environment such as in the evening, cloudy weather, or in a room. However, it may cause noise if the ISO setting is too high. Neighshrink is a method used in the denoising process. Neighshrink uses neighbouring window concept which can be represented as $L \times L$, where L is a positive odd number greater than 1. Images with ISO 1600 and neighbouring window size 3x3 and 5x5 used in this paper. The value of neighbouring window summation, universal threshold and SURE threshold will be used for the thresholding process. The test is conducted by comparing PSNR value from the use of universal threshold and SURE threshold at neighbouring window size 3x3 and 5x5. The test result shows that the highest PSNR value is obtained by the use of universal threshold at neighbouring window size 5x5.

Key words : ISO, Neighshrink, neighbouring window, Universal threshold, SURE threshold.

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Alhamdulillahi rabbil ‘alamin. Puji syukur Penulis haturkan kehadiran Allah SWT yang telah memberikan rahmatNya sehingga Penulis dapat menyelesaikan penulisan skripsi yang berjudul “Denoising Pada Citra Ber-Iso Tinggi Dengan Menggunakan Konsep *Neighbouring Window*”. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.

Skripsi ini bertujuan untuk menerapkan teori *Neighbouring Window* dan pemilihan threshold dalam proses *denoising* yang diterapkan pada citra ber-ISO tinggi.

Dalam penyusunan dan penyelesaian skripsi ini, Penulis mengucapkan terima kasih kepada :

1. Drs. Muh. Arif Rahman, M.Kom selaku pembimbing utama dalam penulisan skripsi ini
2. Drs. Mardji, MT selaku pembimbing pendamping, dosen pembimbing akademis dan Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
3. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer
4. Segenap staf dan karyawan Jurusan Matematika Universitas Brawijaya yang telah membantu penyusunan skripsi ini.
5. Kepada Orang Tua dan juga keluarga besarku yang tak pernah berhenti memberikan doa dan dukungannya kepada Penulis.
6. Sahabatku Nita, Riris, Nafis, Tyas, Dinda dan Septi yang telah meluangkan waktunya untuk membantu proses belajar dan penyusunan skripsi.
7. Rekan-rekan Ilkomers 2006 yang telah memberikan dukungan dan semangatnya.
8. Rekan-rekan Ilkomers 2007, Marissa, Betha, Ivan, Pipit, yang telah memberikan dukungannya.
9. Keluarga “Sawah” yang telah memberikan doa dan dukungan kepada Penulis.
10. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis sadari bahwa masih banyak kekurangan dalam penulisan laporan ini yang disebabkan oleh keterbatasan kemampuan dan pengalaman. Semoga penulisan skripsi ini bermanfaat bagi pembaca.

Malang, Februari 2012

Penulis



DAFTAR ISI

LEMBAR PENGESAHAN	v
LEMBAR PERNYATAAN	vii
ABSTRAK.....	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR LAMPIRAN	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Metode Penelitian	4
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Pengolahan Citra Digital	7
2.2 Pengertian dan Macam Noise	8
2.3 Pengertian ISO	9
2.4 Metode Denoising	10
2.5 Transformasi Wavelet (Wavelet Transform)	11
2.6 Dekomposisi Wavelet	12
2.7 Skema Denoising Dengan Wavelet	13
2.8 Pendekatan Shrinkage Untuk Denoising	14
2.9 Inverse Discrete Wavelet Transform (IDWT)	17
2.10 Kualitas Citra.....	17
BAB III METODOLOGI DAN PERANCANGAN	19
3.1 Perencanaan Sistem.....	20
3.1.1 Deskripsi Umum Sistem	20
3.1.2 Batasan Sistem.....	21
3.2 Perancangan Proses	21
3.2.1 Dekomposisi	21
3.2.2 Perhitungan Shrinkage.....	23
3.2.3 Invers Haar Wavelet Transform	29

3.2.4	Perancangan Pengujian	31
3.3	Perhitungan Manual	32
3.3.1	Proses Dekomposisi Citra	32
3.3.2	Perhitungan Shrinkage	35
3.3.3	Rekonstruksi	42
3.3.4	Perhitungan Peak Signal to Noise Ratio (PSNR)	44
3.4	Pengujian	45
3.4.1	Pengujian Kesalahan.....	45
3.4.2	Pengujian Rasio	45
3.5	Rancangan Antarmuka	46
BAB IV	IMPLEMENTASI DAN PEMBAHASAN	49
4.1	Lingkungan Implementasi	49
4.1.1	Lingkungan Perangkat Keras	49
4.1.2	Lingkungan Perangkat Lunak	49
4.2	Implementasi Program	50
4.2.1	Implementasi Antarmuka.....	50
4.2.2	Implementasi Program	51
4.2.3	Implementasi Pengujian	60
4.3	Skenario Pengujian.....	61
4.3.1	Data Pengujian.....	61
4.3.2	Hasil Pengujian.....	61
4.4	Analisa Hasil Pengujian	70
BAB V	KESIMPULAN DAN SARAN	73
5.1	Kesimpulan	73
5.2	Saran	74
DAFTAR PUSTAKA.....		75
LAMPIRAN		77

DAFTAR GAMBAR

Gambar 2.1 Proses Pengolahan Citra.....	8
Gambar 2.2 a. Salt and pepper Noise b. Gaussian Noise	9
Gambar 2.3 Transformasi Wavelet 2D 1 level (Putra, 2010).....	11
Gambar 2.4 Dekomposisi Lena 1 tingkat.....	12
Gambar 2.5 Contoh neighbouring window ukuran 3x3 (Dengwen,2008).....	15
Gambar 3.1 Tahapan Penelitian.....	19
Gambar 3.2 Deskripsi Umum Sistem.....	20
Gambar 3.3 Diagram Alir Dekomposisi dengan Wavelet Haar	22
Gambar 3.4 Diagram Alir Penentuan Threshold Universal dengan Estimator	24
Gambar 3.5 Diagram Alir Penentuan <i>neighbour</i> ($S_i, j2$)	25
Gambar 3.6 Diagram Alir Perhitungan Threshold Sure	27
Gambar 3.7 Diagram Alir Perbandingan Threshold SURE.....	28
Gambar 3.8 Diagram Alir Neighshrink	29
Gambar 3.9 Diagram Alir Proses Rekonstruksi Wavelet Haar.....	30
Gambar 3.10 Diagram Alir Proses PNSR	31
Gambar 3.11 Rancangan Antarmuka	46
Gambar 4.1 Implementasi Antarmuka.....	50
Gambar 4.2 Grafik nilai MSE citra pada penggunaan Universal Threshold	67
Gambar 4.3 Grafik Nilai PSNR Citra Pada Penggunaan Universal Threshold	67
Gambar 4.4 Grafik nilai MSE citra pada penggunaan SURE Threshold	68
Gambar 4.5 Grafik nilai PSNR citra pada penggunaan SURE Threshold	68
Gambar 4.6 Grafik Perbandingan Nilai Mse Citra Universal Threshold Dan Sure Threshold.....	69
Gambar 4.7 Grafik Perbandingan Nilai Psnr Citra Universal Threshold Dan Sure Threshold.....	69

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Tabel 3.1 Nilai Citra Asli.....	32
Tabel 3.2 Nilai Matrik Filter.....	32
Tabel 3.3 Hasil Perkalian Baris Matrik Filter dengan Kolom Citra	33
Tabel 3.4 Hasil Pengelompokan Aproksimasi dan Detail	33
Tabel 3.5 Hasil Pengelompokan bagian Aproksimasi dan Detail dari Perkalian Kolom Matrik Filter dengan Baris Citra	34
Tabel 3.6 Hasil Dekomposisi Tingkat 1	34
Tabel 3.7 Hasil Dekomposisi 3 Tingkat.....	35
Tabel 3.8 Jumlah kuadrat tetangga pada level 1	36
Tabel 3.9 Jumlah kuadrat tetangga level 2	36
Tabel 3.10 Jumlah kuadrat tetangga level 3	37
Tabel 3.11 Nilai $\frac{\partial g_n}{\partial w_n}$ dan $\ g_n(w_n)\ _2^2$ pada subband LH.....	37
Tabel 3.12 Nilai $\frac{\partial g_n}{\partial w_n}$ dan $\ g_n(w_n)\ _2^2$ pada subband HL.....	38
Tabel 3.13 Nilai $\frac{\partial g_n}{\partial w_n}$ dan $\ g_n(w_n)\ _2^2$ pada subband HH	39
Tabel 3.14 Soft Shrinkage dengan menggunakan Sure threshold ...	40
Tabel 3.15 Neighshrink dengan menggunakan optimal threshold Sure	41
Tabel 3.16 Soft Shrinkage dengan menggunakan universal threshold	41
Tabel 3.17 Neighshrink dengan menggunakan universal threshold	42
Tabel 3.18 Invers Matrik filter LH	42
Tabel 3.19 Citra Hasil rekonstruksi dengan Sure Threshold	43
Tabel 3.20 Citra Hasil rekonstruksi dengan Universal Threshold... 43	43
Tabel 3.21 Rancangan RMSE Denoising Citra	45
Tabel 3.22 Rancangan PSNR Denoising Citra.....	45
Tabel 4.1 Hasil pengujian MSE dan PSNR metode Universal Threhold.....	62
Tabel 4.2 Hasil pengujian MSE dan PSNR metode SURE Threshold	63

Tabel 4.3 Hasil pengujian MSE dan PSNR metode Neighshrink dengan menggunakan Universal Threshold dan SURE Threshold pada ukuran Neighbouring window 5x5	63
Tabel 4.4 Hasil pengujian secara visual	64

UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

Lampiran 1 Nilai MSE dan PSNR Hasil Pengujian Metode Neighshrink dengan menggunakan Universal Threshold pada ukuran Neighbouring window 3x3 dan 5x5	78
Lampiran 2 Nilai MSE dan PSNR Hasil Pengujian Metode Neighshrink dengan menggunakan SURE Threshold pada ukuran Neighbouring window 3x3 dan 5x5	80
Lampiran 3 Perbandingan Nilai MSE dan PSNR Hasil Pengujian Metode Neighshrink dengan menggunakan Universal Threshold dan SURE Threshold pada ukuran Neighbouring window 5x5	82
Lampiran 4 Contoh Hasil Denoising Citra Uji secara Visual pada metode Neighshrink dengan menggunakan Universal Threshold	84
Lampiran 5 Contoh Hasil Denoising Citra Uji secara Visual pada metode Neighshrink dengan menggunakan SURE Threshold	90

UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Citra (*image*) merupakan istilah lain untuk gambar. Sebagai salah satu komponen multimedia, citra memegang peranan sangat penting sebagai bentuk informasi visual. Seiring dengan perkembangan teknologi dalam bidang komputerisasi, teknologi pengolahan citra (*image processing*) telah banyak dipakai di berbagai bidang. Pada umumnya sistem pencitraan menyebabkan terjadinya gangguan atau derau (biasa disebut *noise*) yang menyebabkan citra tersebut tidak dapat dilihat dengan jelas karena adanya noise. Misalnya pada sistem pencitraan secara digital. Pengambilan citra secara digital memungkinkan penggunaan ISO untuk memperoleh kualitas secara optimal, dimana ISO adalah ukuran tingkat sensitifitas sensor kamera terhadap cahaya (King, 2010). Namun penggunaan ISO yang terlalu tinggi dapat menyebabkan timbulnya noise. Bila ISO dinaikkan, noise yang awalnya kecil akan ikut menjadi tinggi (King, 2010).

Pengurangan noise (*denoising*) adalah salah satu proses dalam peningkatan kualitas citra (*image enhancement*). Peningkatan kualitas citra adalah proses perbaikan citra agar lebih mudah diinterpretasikan oleh mata manusia. Proses-proses yang termasuk dalam peningkatan kualitas citra terdiri atas pengubahan kecerahan gambar (*image brightness*), perenggangan kontras (*image stretching*), pengubahan histogram citra, penghilangan noise (*denoising*), penajaman tepi (*edge sharpening*), pewarnaan semu (*pseudocolouring*) dan pengubahan geometric. Salah satu cara yang digunakan untuk *image processing* adalah dengan mereduksi noise (*noise reduction*) yang terdapat pada citra sehingga menjadi citra lain yang kualitasnya lebih baik dibandingkan dengan kondisi awal, sehingga mudah diinterpretasi baik itu oleh manusia maupun mesin (Murinto, 2007).

Terdapat beberapa metode yang digunakan dalam penghilangan noise. Salah satunya adalah penggunaan *transformasi wavelet*. *Transformasi wavelet* telah banyak digunakan dalam berbagai bidang ilmu salah satunya *image denoising*. Terdapat beberapa

metode penghilangan noise yang berbasis wavelet. Proses denoising ini dikenal sebagai *wavelet shrinkage* atau *thresholding* yang dikembangkan oleh Donoho dkk (Taswell, 1998). *Shrinkage* sendiri terbagi menjadi dua, yaitu *soft-shrinkage* (*soft-thresholding*) dan *hard-shrinkage* (*hard-thresholding*). Mereka juga mengembangkan metode *wavelet shrinkage* lain yang dikenal dengan *VisuShrink* dan *SureShrink*. *VisuShrink* menggunakan *universal threshold* berdasar aturan *soft-shrinkage*. *SureShrink* juga dikembangkan dengan aturan *soft-shrinkage* dengan pemilihan threshold secara independen berdasar *Stein's unbiased risk estimate (SURE)*. *SureShrink* menghasilkan hasil yang lebih baik dariada *VisuShrink* (Dengwen dkk, 2008).

Cai dan Silverman (Cai,2001) mengajukan dua metode shrinkage yang berbeda, yaitu *Neighblock* dan *Neighcoeff*. Dari hasil penelitian tersebut, diperoleh hasil bahwa kedua metode yang diajukan Cai dkk menghasilkan performa yang lebih baik daripada *VisuShrink* dan *SureShrink*. Chen dkk (Chen, 2005) mengaplikasikan *Neighcoeff* yang diajukan Cai dan Silverman pada *image denoising* yang selanjutnya dikenal dengan *Neighshrink*. *Neighshrink* menggunakan konsep ketetanggaan. Menurut Chen, koefisien wavelet saling berhubungan pada neighbour yang kecil. Neighboorhood window sekitar koefisien wavelet yang akan dithreshold memiliki pengaruh terhadap kemampuan algoritma *Neighshrink*. Pada ukuran window yang luas, maka kemungkinan akan memiliki koefisien noise yang besar pada neighbournya. Ukuran neighbouring window yang baik yaitu 3x3 dan 5x5. Dari hasil penelitian yang dilakukan Chen dkk tersebut, didapatkan hasil bahwa hasil metode *NeighCoeff* yaitu *Neighshrink* memiliki hasil lebih baik dari *VisuShrink* dan *SureShrink*.

Dari latar belakang yang telah diuraikan, maka akan dilakukan penelitian mengenai image denoising pada citra ber-ISO tinggi dengan menggunakan metode *Neighshrink* dengan penggunaan Universal Threshold dan SURE Threshold pada wavelet domain dan pengaruh ukuran *neighbouring window* pada proses denoising dengan judul “Denoising Pada Citra Ber-ISO Tinggi Dengan Menggunakan Konsep *Neighbouring Window*”.

1.2 Rumusan Masalah

Permasalahan yang diangkat dalam pembuatan tugas akhir ini adalah :

1. Bagaimana tingkat keberhasilan *denoising* pada citra ber-ISO tinggi dengan menggunakan metode *Neighshrink* dengan menggunakan universal threshold pada ukuran neighbouring window 3x3 dan 5x5.
2. Bagaimana tingkat keberhasilan *denoising* pada citra ber-ISO tinggi dengan menggunakan metode *Neighshrink* dengan menggunakan SURE threshold pada ukuran neighbouring window 3x3 dan 5x5.
3. Bagaimana tingkat keberhasilan metode *Neighshrink* yang universal threshold dan SURE threshold dalam menghilangkan *noise* pada ukuran neighbouring window 5x5.

1.3 Batasan Masalah

Agar penulisan dan penyelesaian penulisan ini tidak melebar lebih luas akan diberikan batasan-batasan sebagai berikut :

1. Metode yang digunakan diadopsi dari jurnal Dengwen dkk (Dengwen, 2008) yaitu menggunakan metode wavelet *Neighshrink* yang berbasis SURE serta jurnal Chen dkk (Chen, 2005) yaitu menggunakan *Neighshrink* menggunakan universal threshold.
2. Citra yang digunakan adalah citra grayscale dengan ISO 1600 dalam format bmp(.bmp) dengan ukuran NxN.
3. Metode pengujian tingkat keberhasilan dari metode *Neighshrink* pada penggunaan Universal Threshold dengan *Neighshrink* pada penggunaan SURE Threshold dengan menggunakan *Peak Signal to Noise Ratio (PNSR)* dan *MSE (Mean Square Error)*.

1.4 Tujuan Penelitian

Tujuan dari pembuatan tugas akhir ini adalah :

1. Mengimplementasikan *Neighshrink* universal threshold dan SURE threshold pada citra ber-ISO tinggi untuk menghilangkan *noise* pada ukuran neighbouring window yang berbeda.
2. Mengetahui tingkat keberhasilan dari kedua metode tersebut.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diambil dalam penulisan tugas akhir kali ini adalah dapat menghasilkan suatu aplikasi *denoising* pada citra ber-ISO tinggi yang dapat digunakan untuk keperluan dalam perbaikan kualitas citra ber-ISO tinggi yang telah diperoleh.

1.6 Metode Penelitian

Dalam pencapaian tujuan yang dirumuskan sebagaimana yang telah dipaparkan sebelumnya, maka metodologi yang digunakan dalam penulisan skripsi ini adalah:

1. Studi Literatur.

Mempelajari teori-teori dan konsep-konsep dasar serta identifikasi masalah image denoising dan mengenai konsep citra ber-ISO tinggi, transformasi wavelet berbasis *shrinkage*, *Neighshrink* berupa buku, jurnal ilmiah maupun melalui internet.

2. Pendefinisian dan analisis masalah

Menganalisis permasalahan dan memilih solusi yang tepat.

3. Perancangan dan implementasi sistem

Membuat perancangan perangkat lunak dan mengimplementasikan hasil rancangan tersebut yaitu membuat aplikasi denoising pada citra ber-iso tinggi menggunakan konsep *neighbouring window*.

4. Uji coba dan analisis hasil implementasi

Menguji perangkat lunak dan menganalisis hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya kemudian dilakukan evaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Dalam pembuatan tugas akhir ini seluruh tahapan kegiatan penelitian akan dijelaskan secara terperinci dengan pembagian bab sebagai berikut:

BAB I: PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, tujuan yang ingin dicapai, batasan masalah dari permasalahan, manfaat, metodologi pemecahan masalah, serta sistematika penulisan

BAB II: TINJAUAN PUSTAKA

Pada bab ini menjelaskan tentang citra digital secara umum, definisi ISO, teori *wavelet*, teori konsep *wavelet shrinkage*, teori *wavelet neighshrink* dan *SURE*. Adapun literature yang digunakan meliputi buku referensi dan dokumentasi internet.

BAB III: METODOLOGI DAN PERANCANGAN

Bab ini menjelaskan metode-metode yang digunakan dalam menyelesaikan masalah *denoising* pada citra ber-ISO tinggi dengan menggunakan konsep neighbouring window. Adapun bagian yang akan dibahas adalah: Deskripsi Umum Sistem, Mekanisme kerja sistem, Perancangan uji coba dan Evaluasi hasil.

BAB IV: HASIL DAN PEMBAHASAN

Pada bab ini akan menjelaskan proses implementasi dari rancangan sistem yang telah dijelaskan pada BAB III. Penjelasan Implementasi terdiri dari implementasi program, penerapan aplikasi, hasil uji coba dan analisa hasil percobaan berdasarkan nilai MSE dan PSNR.

BAB V: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan tugas akhir selanjutnya.

UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

2.1 Pengolahan Citra Digital

Citra (*image*) merupakan istilah lain untuk gambar. Sebagai salah satu komponen multimedia, citra memegang peranan sangat penting sebagai bentuk informasi visual. Seiring dengan perkembangan teknologi dalam bidang komputerisasi, teknologi pengolahan citra (*image processing*) telah banyak dipakai di berbagai bidang.

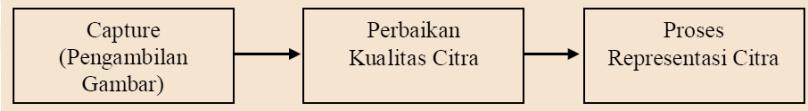
Dalam pengolahan maupun pengenalan citra, masalah persepsi visual, yaitu apa yang dilihat manusia mempunyai peranan penting. Penentuan apa yang dapat dilihat itu tidak dapat hanya ditentukan oleh manusia sendiri.

Pengolahan citra digital (PCD) adalah pemrosesan citra dengan menggunakan teknologi komputer. Pengolahan citra dapat dilakukan oleh komputer pada domain spasial maupun domain frekuensi. Dengan menggunakan teknologi komputer, pengolahan citra mempunyai tujuan yaitu :

1. Proses memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau komputer
2. Teknik pengolahan citra dengan mentrasformasikan citra menjadi citra lain
3. Pengolahan citra merupakan proses awal dari komputer visi

Ada beberapa hal yang penting didalam pengolahan citra digital, antara lain teknik-teknik pengambilan citra, model citra digital, sampling dan kuantitasi, histogram, proses filtering, perbaikan citra sampai pada pengolahan citra digital yang lebih lanjut seperti segmentasi, *image clustering* dan ekstrasi ciri.

Proses pengolahan citra secara diagram yaitu proses dimulai dari pengambilan citra, perbaikan citra sampai dengan pernyataan representatif citra digambarkan dengan Gambar 2.1 :



Gambar 2.1 Proses Pengolahan Citra

2.2 Pengertian dan Macam Noise

Setiap gangguan pada citra disebut *noise*. *Noise* bisa terjadi pada saat proses pengambilan gambar, ada beberapa gangguan yang mungkin terjadi, seperti kamera tidak focus, munculnya bintik-bintik yang bisa jadi disebabkan oleh proses pengambilan gambar yang tidak sempurna, adanya kotoran-kotoran yang terjadi pada citra.

Berdasarkan bentuk dan karakteristiknya, *noise* pada citra dibedakan menjadi beberapa macam, sebagai berikut (Bovik, 2009):

- Gaussian

Merupakan model noise yang mengikuti distribusi normal standart dengan rata-rata nol dan standart deviasi 1. Efek dari noise ini adalah munculnya titik-titik berwarna yang jumlahnya sama dengan persentase noise.

- Speckle

Merupakan model noise yang memberikan warna hitam pada titik yang terkena noise.

- Salt and Pepper

Merupakan model noise seperti taburan garam dan merica yang memberikan warna putih pada titik yang terkena noise.

- Noise uniform

Merupakan noise sintetis yang sebenarnya jarang digunakan dalam penerapan, tetapi secara pemrograman pembangkitan noise uniform ini merupakan jenis pembangkitan noise yang paling mudah.

Contoh Gaussian Noise dan Salt and Pepper Noise ditunjukkan oleh Gambar 2.2 :



a.



b.

Gambar 2.2 a. Salt and pepper Noise b. Gaussian Noise

2.3 Pengertian ISO

Secara definisi, ISO adalah ukuran tingkat kepekaan sensor kamera terhadap cahaya (King, 2010). Dalam fotografi, ISO dan ASA merupakan standar untuk mengukur kecepatan film atau mengukur kepekaan film terhadap cahaya. ASA adalah standar yang dibuat oleh American Standards Association sedangkan ISO merupakan standar yang dibuat oleh International Organization for Standardization. Dulu ketika kamera film masih berjaya maka satuan kepekaan film terhadap cahaya yang sering digunakan adalah ASA. Saat ini, kamera digital lebih banyak digunakan sehingga istilah ISO lebih populer dibanding ASA karena ISO merujuk pada kepekaan sensor kamera digital sedangkan ASA merujuk pada kepekaan film.

Ada banyak sistem pengukuran kepekaan film, tetapi dua yang paling banyak digunakan sebelum tahun 1987 adalah ASA dan DIN (dibuat oleh Institut Jerman untuk Standardisasi). ASA menggunakan skala aritmatika yang berarti menggandakan sensitivitas film berarti juga akan menggandakan nilai ASA. Sedangkan DIN adalah skala logaritma, sehingga menggandakan sensitivitas akan menambahkan nilai 3 derajat ke nomor DIN. ISO pada dasarnya menggabungkan kedua metode tersebut dan kini menjadi standar tunggal yang berlaku di seluruh dunia.

Dengan adanya ISO maka kita dapat mengatur kepekaan sensor kamera digital terhadap cahaya. Semakin tinggi nilai ISO maka

semakin sensitif sensor tersebut sehingga dibutuhkan lebih sedikit cahaya untuk menghasilkan suatu image.

Untuk mendapatkan gambaran yang jelas tentang setting ISO di kamera digital, maka dapat diumpamakan sebagai suatu komunitas lebah. Sebuah ISO adalah seekor lebah pekerja. Jika kamera digital diatur menggunakan ISO 100, artinya terdapat 100 lebah pekerja. Jika menggunakan ISO 200 maka setara dengan 200 lebah pekerja dan seterusnya. Tugas setiap lebah pekerja adalah memungut cahaya yang masuk melalui lensa kamera dan membuat gambar. Jika 2 kamera dengan setting yang sama digunakan untuk mengambil gambar sebuah objek pada kondisi yang sama dengan salah satu kamera menggunakan ISO 100 dan kamera lainnya menggunakan ISO 200, maka kamera dengan ISO 200 akan lebih cepat selesai dalam membuat image.

Sayangnya peningkatan ISO juga akan membawa efek negatif yang tidak diinginkan. Meningkatkan ISO berarti meningkatkan sensitivitas sensor, sehingga sinyal yang lemah pun dapat menjadi kuat. Masalahnya, pada proses kerja sensor juga menghasilkan *noise* yang mengiringi sinyal aslinya. Bila ISO dinaikkan, *noise* yang awalnya kecil pun akan ikut menjadi tinggi (King, 2010). *Noise* yang tinggi akan tampak mengganggu pada hasil foto dan muncul berupa titik-titik warna yang tidak enak untuk dilihat. Masalah *noise* ini akan lebih parah apabila jenis sensor yang digunakan adalah sensor berukuran kecil. Karena sensor kecil memiliki ukuran titik/piksel yang kecil juga, dan secara teori piksel kecil lebih rentan terhadap *noise* dibandingkan piksel berukuran lebih besar.

2.4 Metode Denoising

Telah dijelaskan sebelumnya bahwa noise pada citra dapat menyebabkan menurunnya kualitas citra dan hilangnya beberapa detail informasi citra. Oleh karena itu, proses degradasi citra untuk menghilangkan noise merupakan proses penting dalam bidang pengolahan citra. Proses degradasi tersebut disebut *denoising*. *Denoising* citra dapat dilakukan dengan berbagai macam metode. Metode-metode tersebut antara lain :

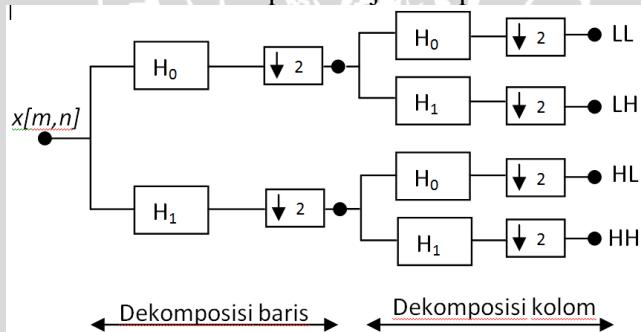
- Operasi aritmatik (aljabar),
- Transformasi wavelet
- Metode contour

- Metode intensity filtering (murinto,2007)
- Metode frequency filtering (murinto,2007)
- Dan lain sebagainya.

Teknik yang sesuai untuk denoising citra dengan menggunakan wavelet dikenal dengan metode *shrinkage wavelet* yang diperkenalkan oleh *Donoho dan Johnstone* pada tahun 1994, dimana shrinkage function digunakan pada perubahan koefisien dari image yang bernoise (Rahmawati,2010). Operasi *denoise* dilakukan untuk menekan komponen yang berfrekuensi tinggi dan meloloskan komponen yang berfrekuensi rendah (Munir, R., 2004). Pada Tugas Akhir ini digunakan metode transformasi wavelet untuk denoising citra.

2.5 Transformasi Wavelet (Wavelet Transform)

Secara umum transformasi wavelet diskrit merupakan dekomposisi citra pada frekuensi subband pada citra tersebut dimana komponen-komponen yang dihasilkan diperoleh dari penurunan level dekomposisi (Novamizanti, 2009). Pada citra 2D, proses transformasi wavelet dilakukan pada baris terlebih dahulu, kemudian dilanjutkan dengan transformasi pada kolom. Implementasi transformasi wavelet diskrit dapat ditunjukkan pada Gambar 2.3.

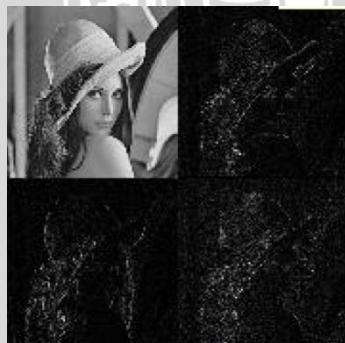


Gambar 2.3 Transformasi Wavelet 2D 1 level (Putra, 2010)

2.6 Dekomposisi Wavelet

Proses dekomposisi wavelet diperoleh dengan melewatan sinyal frekuensi tinggi atau highpass filter dan frekuensi rendah atau lowpass filter (Novamizanti, 2010). Citra semula yang ditransformasi dibagi atau didekomposisi menjadi 4 sub-image baru. Setiap sub-image baru tersebut berukuran $\frac{1}{4}$ kali dari citra asli. 3 sub-image pada posisi atas kanan, bawah kiri dan bawah kanan tampak seperti versi kasar citra asli karena ketiga sub-image tersebut mengandung komponen berfrekuensi tinggi. Sedangkan sub-image posisi atas kiri tampak seperti citra asli dan agak halus karena mengandung komponen frekuensi rendah.

Pada Gambar 2.3, LL menyatakan bagian komponen yang diperoleh melalui proses Lowpass filter dilanjutkan dengan lowpass filter sehingga bagian ini sering disebut komponen aproksimasi. LH menyatakan bagian koefisien yang diperoleh melalui proses Lowpass filter dilanjutkan dengan Highpass filter. Koefisien pada bagian ini menunjukkan citra tepi dalam arah horizontal. Bagian HL merupakan bagian yang diperoleh melalui proses Highpass filter dilanjutkan dengan Lowpass filter. Koefisien pada bagian ini menunjukkan citra tepi dalam arah vertikal. Bagian HH merupakan bagian yang diperoleh melalui proses Highpass filter dilanjutkan dengan Highpass filter. Koefisien pada bagian ini menunjukkan citra tepi dalam arah diagonal. Ketiga komponen LH, HL dan HH disebut juga komponen detail (Putra, 2010). Contoh hasil dekomposisi 1 level ditunjukkan oleh Gambar 2.4 :



Gambar 2.4 Dekomposisi Lena 1 tingkat

Hasil transformasi wavelet 2D 1 level, sering dibuat dalam bentuk skema sebagai berikut (Putra, 2010)

$$\begin{array}{c|c} \text{LL} & \text{HL} \\ \hline \end{array} = \begin{array}{c|c} \text{Approximation} & \text{Vertical details} \\ \hline \end{array} = \begin{array}{c|c} \text{C}_A & \text{C}_V \\ \hline \end{array}$$

$$\begin{array}{c|c} \text{LH} & \text{HH} \\ \hline \end{array} = \begin{array}{c|c} \text{Horizontal detail} & \text{Digonal details} \\ \hline \end{array} = \begin{array}{c|c} \text{C}_H & \text{C}_D \\ \hline \end{array}$$

CA, CV, CH dan CD berturut-turut menyatakan komponen aproksimasi, vertical, horizontal dan diagonal.

Dekomposisi satu tingkat dapat dituliskan dengan ekspresi matematika pada persamaan 2.1 dan 2.2 (Alfatwa,2008).

dengan $y_{tinggi}[k]$ dan $y_{rendah}[k]$ adalah hasil dari *highpass filter* dan *lowpass filter*, $x[n]$ merupakan sinyal asal, $h[n]$ adalah *highpass filter*, dan $g[n]$ adalah *lowpass filter*. Untuk dekomposisi lebih dari satu tingkat, persamaan 2.1 dan 2.2 dapat diterapkan pada masing-masing tingkat.

2.7 Skema Denoising Dengan Wavelet

Transformasi wavelet telah banyak digunakan dalam berbagai bidang ilmu salah satunya *image denoising*. Terdapat beberapa metode penghilangan noise yang berbasis wavelet. Proses denoising ini dikenal sebagai *wavelet shrinkage* atau *thresholding* yang dikembangkan oleh Donoho dkk (Taswell, 1998). *Shrinkage* sendiri terbagi menjadi dua, yaitu *soft-shrinkage* (*soft-thresholding*) dan *hard-shrinkage* (*hard-thresholding*) (Gunara, 2007). Anggap $A(i,j)$ adalah citra bebas noise dengan ukuran NxN pixel dan $B(i,j)$ adalah citra dengan noise $Z(i,j)$ sehingga $B(i,j) = A(i,j) + \sigma Z(i,j)$. Skema denoising wavelet Donoho adalah sebagai berikut (Chen, 2007):

1. Transformasi citra $B(i,j)$ dengan menggunakan transformasi diskrit wavelet

2. Menerapkan soft atau hard thresholding pada koefisien wavelet yang dihasilkan dengan menggunakan threshold $\lambda = \sqrt{2 \sigma^2 \log N^2}$ 2.3
3. Terapkan invers transformasi diskrit wavelet untuk memperoleh citra bebas noise.

2.8 Pendekatan Shrinkage Untuk Denoising

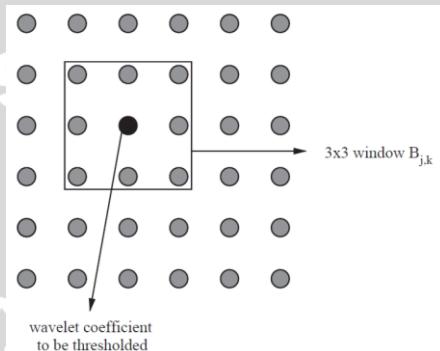
Transformasi wavelet dapat dilakukan dengan menerapkan lowpass dan highpass filter pada koefisien berfrekuensi rendah. Beberapa metode telah dikembangkan dengan menganggap pengaruh koefisien wavelet lain pada koefisien wavelet yang akan dithreshold sangatlah penting. Suatu koefisien wavelet yang besar kemungkinan akan memiliki koefisien wavelet yang besar pula pada neighbournya. Hal itu disebabkan karena transformasi wavelet menghasilkan koefisien wavelet yang saling berhubungan. Menurut Chen (2007), koefisien wavelet saling berhubungan pada neighbour yang kecil. Neighbourhood window sekitar koefisien wavelet yang akan dithreshold memiliki pengaruh terhadap kemampuan algoritma *Neighshrink*. Pada ukuran window yang luas, maka kemungkinan akan memiliki koefisien noise yang besar pada neighbournya. Ukuran neighbouring window yang baik yaitu 3×3 dan 5×5 .

Algoritma denoising neighbour wavelet adalah sebagai berikut (Chen, 2007) :

1. Menerapkan dekomposisi wavelet 2D pada citra ternoise
2. Menerapkan skema shrinkage untuk men-threshold koefisien wavelet dengan menggunakan *neighbouring window* $B_{j,k}$ dan *universal threshold* $\lambda = \sqrt{2 \sigma^2 \log N^2}$
3. Menerapkan invers transformasi diskrit wavelet pada wavelet koefisien yang telah *dithreshold*.

Untuk setiap level dekomposisi wavelet akan dihasilkan 4 frekuensi subband, yaitu LL, LH, HL dan HH. Karena nilai koefisien wavelet pada frekuensi rendah harus dijaga, maka hanya level frekuensi tinggi yang akan dikenai *thresholding* yaitu subband LH, HL dan HH. Untuk setiap koefisien wavelet $d_{j,k}$ yang akan diubah, perlu diperhatikan *neighbouring window* $B_{j,k}$ disekitarnya (Chen, 2007). Ukuran neighbouring window dapat direpresentasikan sebagai $L \times L$ dimana L adalah bilangan ganjil positive lebih dari 1

(Dengwen, 2008). Menurut hasil penelitian Chen dkk (Chen, 2007) ukuran window 3×3 dan 5×5 adalah ukuran yang disarankan dalam algoritma *Neighbouring*. Jika ukuran *neighbouring window* terlalu besar, maka kemampuan denoising menjadi lemah sehingga hasil yang diperoleh kurang maksimal. Pada Gambar 2.5 ditunjukkan *neighbouring window* dengan ukuran 3×3 dengan koefisien wavelet yang akan dithreshold.



Gambar 2.5 Contoh neighbouring window ukuran 3×3
(Dengwen,2008)

Nilai kuadrat *neighbouring window* dapat dihitung dengan menggunakan persamaan 2.4.

$$S_{j,k}^2 = \sum_{(i,l) \in B_{j,k}} d_{i,l}^2 \quad 2.4$$

Apabila $S_{j,k}$ memiliki pixel yang keluar dari range subband wavelet, maka aturan dalam penjumlahan tersebut diabaikan(Dengwen, 2008). Untuk wavelet koefisien, $d_{j,k}$,yang akan dithreshold, koefisien tersebut di-shrinkage dengan persamaan 2.5 (Zhou, 2005) :

$$d_{j,k} = d_{j,k} B_{j,k} \quad 2.5$$

dengan faktor shrinkage pada persamaan 2.6 :

$$B_{j,k} = \left(1 - \frac{\lambda^2}{S_{j,k}^2} \right) \quad 2.6$$

Proses thresholding dilakukan dengan aturan apabila nilai jumlah kuadrat *neighbouring window* (pada persamaan 2.4) kurang dari atau sama dengan nilai kuadrat threshold, maka nilai koefisien

wavelet akan diset menjadi nol, sebaliknya koefisien wavelet akan dishrink dengan menggunakan faktor shrinkage pada persamaan 2.6.

Satu diantara metode optimal threshold yang ada adalah dengan meminimalkan *Stein's unbiased risk estimation* (SURE) untuk setiap level dekomposisi wavelet.

SURE digunakan untuk mengestimasi optimal threshold pada neighbouring window L pada *Neighshrink* pada setiap wavelet subband. Nilai estimator yang baik untuk σ (noise level) pada citra yang tidak diketahui nilai varian noisenya adalah median absolute deviation (MAD) menggunakan koefisien wavelet pada subband HH pada level paling tinggi (Donoho dan Johnstone,1994). Nilai estimator noise level ditunjukkan pada persamaan 2.7 :

$$\sigma = \frac{\text{median}(|w_s|)}{0,6745} \quad (w_s \in \text{subband HH}) \quad 2.7$$

Untuk memudahkan dalam penulisan, maka koefisien wavelet pada subband s , dapat dilakukan penulisan dalam bentuk 1 dimensi sebagai $w_s = \{w_n : n = 1, \dots, N_s\}$.

Ketika w_j adalah subband wavelet NxN pada level j, SURE threshold λ^s dapat diperoleh dengan persamaan 2.8 :

$$\text{SURE}(w_s, \lambda, L) = N_s + \sum_n \left\| g_n(w_n) \right\|_2^2 + 2 \sum_n \frac{\partial g_n}{\partial w_n} \quad 2.8$$

dengan nilai $\frac{\partial g_n}{\partial w_n}$ dan $\left\| g_n(w_n) \right\|_2^2$ dapat diperoleh dengan persamaan 2.9 dan 2.10 (Dengwen,2008) :

$$\frac{\partial g_n}{\partial w_n} = \begin{cases} -\lambda^2 * \frac{S_n^2 - 2w_n^2}{S_n^4} & (\lambda^2 < S_n) \\ -1 & \text{sebaliknya} \end{cases} \quad 2.9$$

$$\left\| g_n(w_n) \right\|_2^2 = \begin{cases} \frac{\lambda^4}{S_n^4} w_n^2 & (\lambda^2 < S_n) \\ w_n^2 & \text{sebaliknya.} \end{cases} \quad 2.10$$

Kemudian untuk menentukan nilai threshold dipilih dari nilai $\text{SURE}(w_s, \lambda, L)$ yang paling kecil.

2.9 Inverse Discrete Wavelet Transform (IDWT)

Proses rekonstruksi diawali dengan menggabungkan koefisien DWT dari yang berada pada akhir dekomposisi melalui *highpass filter* dan *lowpass filter*. Proses rekonstruksi ini merupakan kebalikan dari proses dekomposisi.

Matrik pengali untuk lowpass L^* dan highpass H^* untuk proses rekonstruksi merupakan invers dari matriks pengali filter pada dekomposisi. Matrik L^* dan H^* untuk basis Haar ditunjukkan pada persamaan 2.11 dan 2.12 :

$$L^* = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix} \quad 2.11$$

$$H^* = \begin{bmatrix} 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} \\ 0 & -1/\sqrt{2} \end{bmatrix} \quad 2.12$$

2.10 Kualitas Citra

Salah satu cara yang digunakan untuk perhitungan kualitas citra dapat dilakukan dengan menggunakan *Peak Signal to Noise Ratio (PSNR)* sebagai pembanding atas citra hasil *denoising* dengan citra asal.

1. Peak Signal to Noise Ratio(PSNR)

PSNR digunakan sebagai dasar matematis untuk mengukur perbedaan secara objektif diantara dua citra dengan satuan *decibel* (dB). PSNR menaksir kualitas dari hasil restorasi citra terhadap citra asli. Untuk citra hasil restorasi, nilai PSNR biasanya berkisar antara 25 – 50 dB. Nilai ini dihitung berdasarkan perbandingan antara sinyal dengan nilai noise. Semakin besar nilai SNR semakin baik kualitas sinyal yang dihasilkan.

Pertama yang dilakukan adalah menghitung nilai *mean squared error (MSE)* dari suatu citra hasil rekonstruksi. MSE dihitung untuk seluruh pixel dalam citra. *Root mean squared error (RMSE)* adalah akar dari *MSE*. Perhitungan nilai *MSE* ditunjukkan oleh persamaan 2.13 :

$$MSE = \frac{\sum [f(i,j) - F(i,j)]^2}{N^2} \quad 2.13$$

N^2 menyatakan hasil perkalian panjang dan lebar citra dalam pixel. $F(i,j)$ merupakan citra hasil denoising, sedangkan $f(i,j)$ adalah citra asal. Berdasarkan persamaan MSE tersebut, maka nilai PNSR direpresentasikan dalam skala decibel (dB) dan dapat dihitung dengan persamaan 2.14 :

$$PSNR = 10 \log_{10} \left(\frac{255^2}{RMSE} \right) \quad 2.14$$

Nilai 255 dalam persamaan 2.14 merupakan batas atas dari sebuah nilai pixel. Nilai MSE yang paling baik nilai MSE paling rendah sedangkan nilai PSNR yang paling baik adalah nilai PSNR yang paling tinggi (Linda S, 2005).



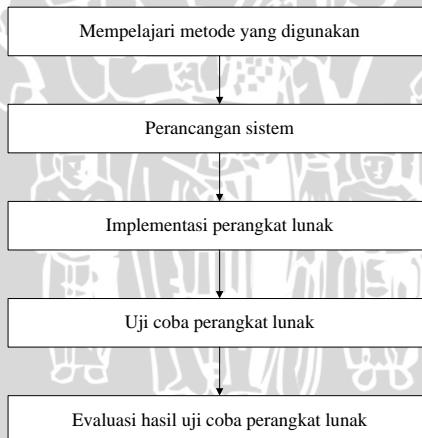
BAB III

METODOLOGI DAN PERANCANGAN SISTEM

Pada bab metodologi dan perancangan akan dibahas mengenai penggunaan metode yang digunakan dalam pembuatan perangkat penghilangan noise. Adapun tahapan yang dilakukan adalah sebagai berikut:

1. Mempelajari metode yang digunakan dari berbagai sumber seperti yang telah dijelaskan pada bab 2.
2. Merancang perangkat lunak dengan metode yang akan digunakan.
3. Membuat perangkat lunak berdasarkan perancangan yang dilakukan.
4. Uji coba perbaikan citra ber-ISO tinggi menggunakan perangkat lunak yang telah dibuat.
5. Evaluasi hasil uji coba dan analisis.

Langkah-langkah dalam pembuatan perangkat lunak tersebut ditunjukkan pada Gambar 3.1.



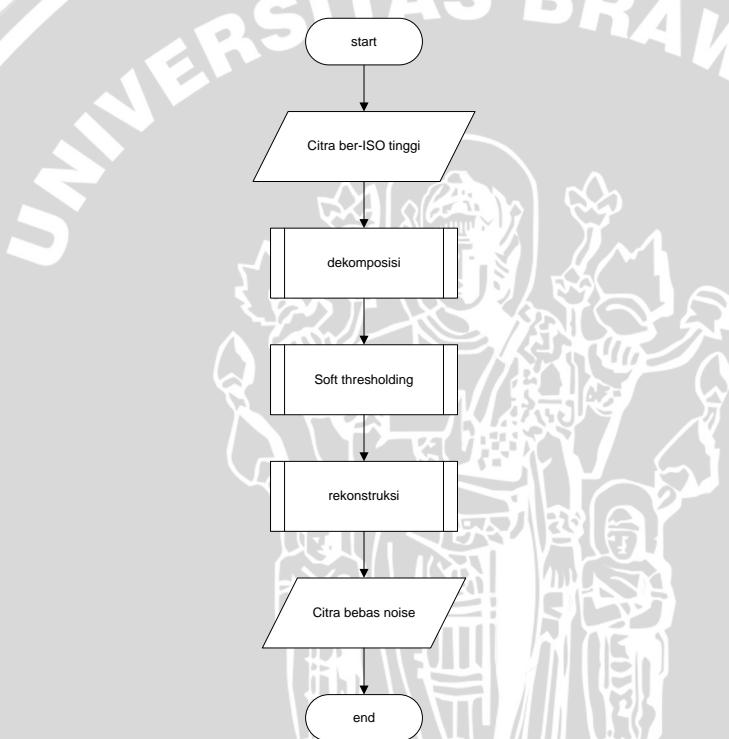
Gambar 3.1 Tahapan Penelitian

3.1 Perencanaan Sistem

3.1.1 Deskripsi Umum Sistem

Sistem ini dibangun untuk menghilangkan noise dari suatu citra dengan menggunakan konsep neighbouring window. Data masukkan dalam sistem ini merupakan citra ber-ISO tinggi. Citra masukkan akan mengalami beberapa proses pengolahan citra sehingga akan menghasilkan citra yang lebih baik dari sebelumnya.

Deskripsi umum sistem dapat dijabarkan pada gambar 3.2.



Gambar 3.2 Deskripsi Umum Sistem

Pada Gambar 3.2 citra input akan mengalami proses transformasi untuk diubah ke domain frekuensi dengan menggunakan Haar Wavelet. Setelah itu akan dilakukan perhitungan thresholding dengan menggunakan soft thresholding.

Deskripsi umum sistem dapat dijelaskan sebagai berikut :

1. Citra inputan merupakan citra ber-ISO tinggi
2. Citra akan mengalami transformasi dengan menggunakan wavelet haar
3. Selanjutnya citra dikenai proses thresholding dengan menggunakan soft threshold dengan nilai threshold yang diperoleh dari perhitungan
4. Citra dikenai rekonstruksi
5. Hasil akhir dari keseluruhan proses adalah citra bebas noise.

3.1.2 Batasan Sistem

Perancangan sistem ini dibatasi dengan menggunakan citra ber-ISO tinggi (ISO 1600) yang merupakan citra grayscale dengan format bmp, dengan ukuran citra 256x256.

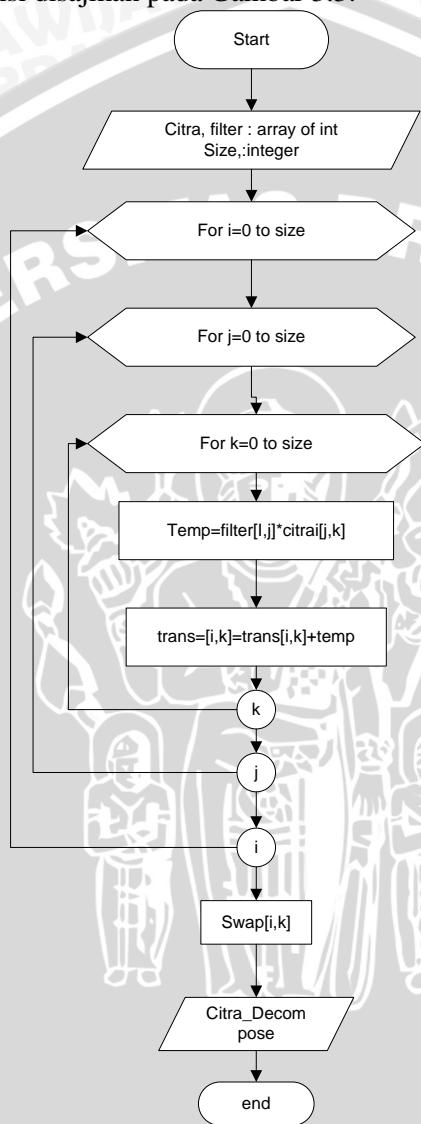
3.2 Perancangan Proses

3.2.1 Dekomposisi

Proses dekomposisi wavelet diperoleh dengan melewatkannya sinyal frekuensi tinggi atau *highpass* filter dan frekuensi rendah atau *lowpass* filter. Proses transformasi wavelet dilakukan pada baris terlebih dahulu, kemudian dilanjutkan dengan transformasi pada kolom. Langkah-langkah yang dilakukan adalah :

1. Memasukkan matriks citra yang akan didenoise
2. Mendefinisikan ukuran matriks citra dengan matriks filter
3. Melakukan lowpass filter dan highpass filter terhadap baris(mengalikan matrik filter dengan matrik citra)
4. Melakukan swap baris (dengan tujuan mengelompokkan bagian aproksimasi dan detail)
5. Melakukan lowpass filter dan highpass filter terhadap kolom (mengalikan matrik citra dengan matrik filter)
6. Melakukan swap kolom(dengan tujuan mengelompokkan bagian aproksimasi dan detail)

Langkah dekomposisi disajikan pada Gambar 3.3.



Gambar 3.3 Diagram Alir Dekomposisi dengan Wavelet Haar

3.2.2 Perhitungan Shrinkage

Untuk melakukan thresholding, dibutuhkan nilai thresholding. Untuk itu dilakukan perhitungan nilai threshold tersebut adalah :

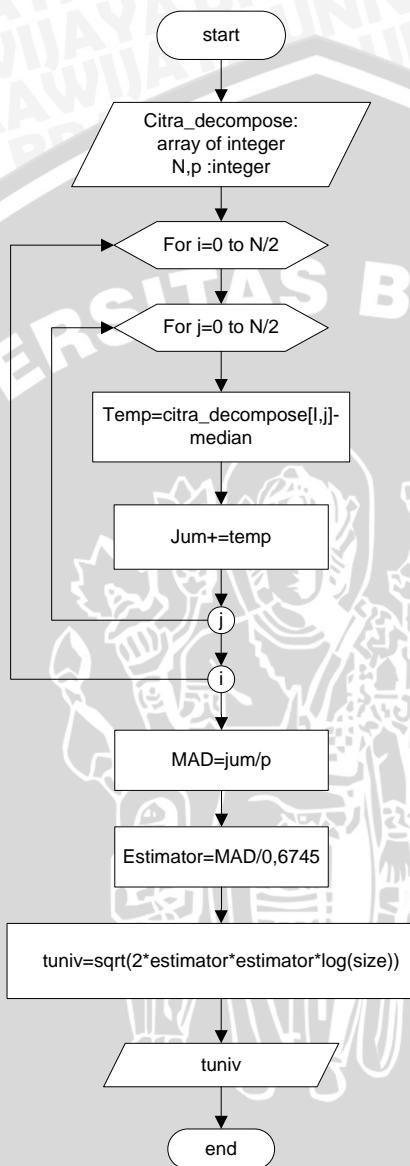
1. Masukan berupa data citra hasil dekomposisi(koefisien wavelet)
2. Menghitung median dari koefisien wavelet pada subband HH
3. Mengurangi tiap koefisien wavelet pada subband HH dengan median
4. Menghitung jumlah hasil pengurangan dan membagi dengan jumlah data (MAD)
5. Menghitung nilai estimator (σ) dengan cara membagi MAD dengan 0,6745
6. Menghitung nilai threshold dengan mengalikan σ dengan akar dari $2*\log(\text{jumlah data})$
7. Keluaran berupa nilai threshold universal λ

Diagram penentuan nilai threshold universal terdapat pada Gambar 3.4.

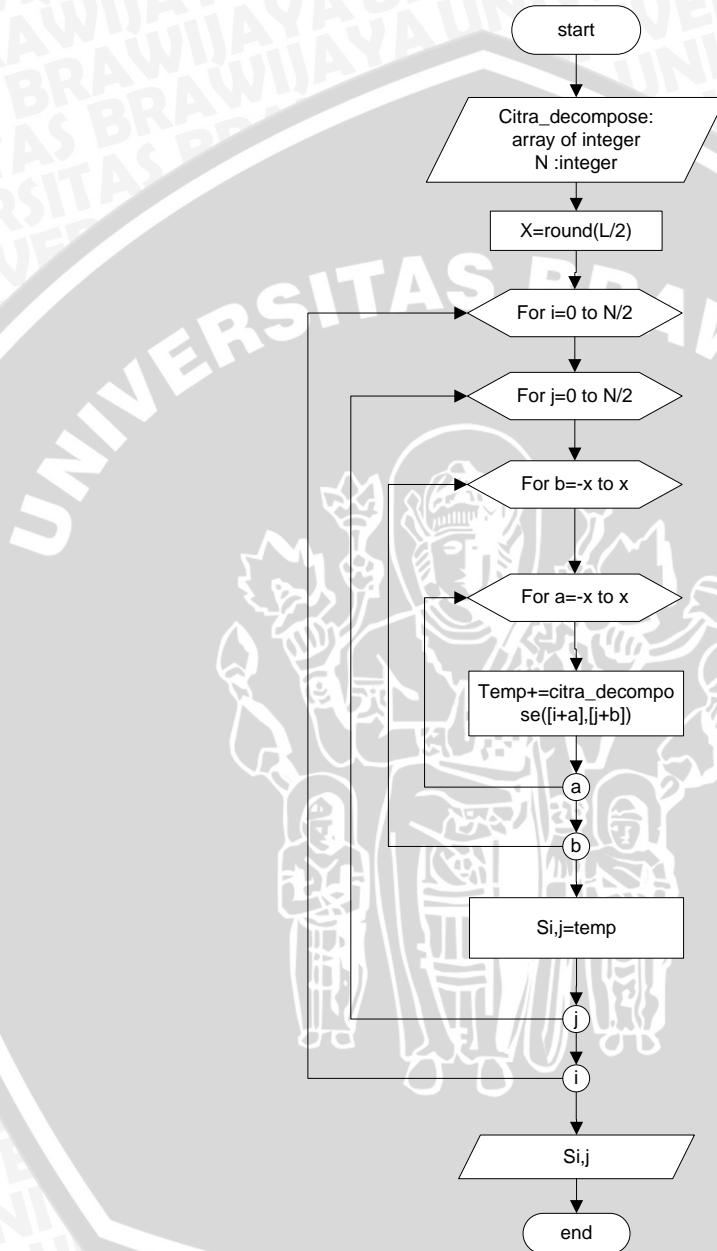
Langkah selanjutnya yang harus dilakukan adalah menentukan summation neighbour window dari koefisiien wavelet pada tiap subband. Berikut adalah langkah-langkahnya:

1. Masukan berupa data citra hasil dekomposisi(koefisien wavelet)
2. Menentukan ukuran neighbour window
3. Melakukan perulangan untuk menghitung jumlah kuadrat dari koefisien wavelet sesuai dengan ukuran neighbor window pada tiap subband yang digunakan pada proses selanjutnya
4. Hasil berupa $\sum \text{neighbour} (S_{i,j}^2)$

Diagram penentuan $\sum \text{neighbour} (S_{i,j}^2)$ terdapat pada Gambar 3.5.



Gambar 3.4 Diagram Alir Penentuan Threshold Universal dengan Estimator



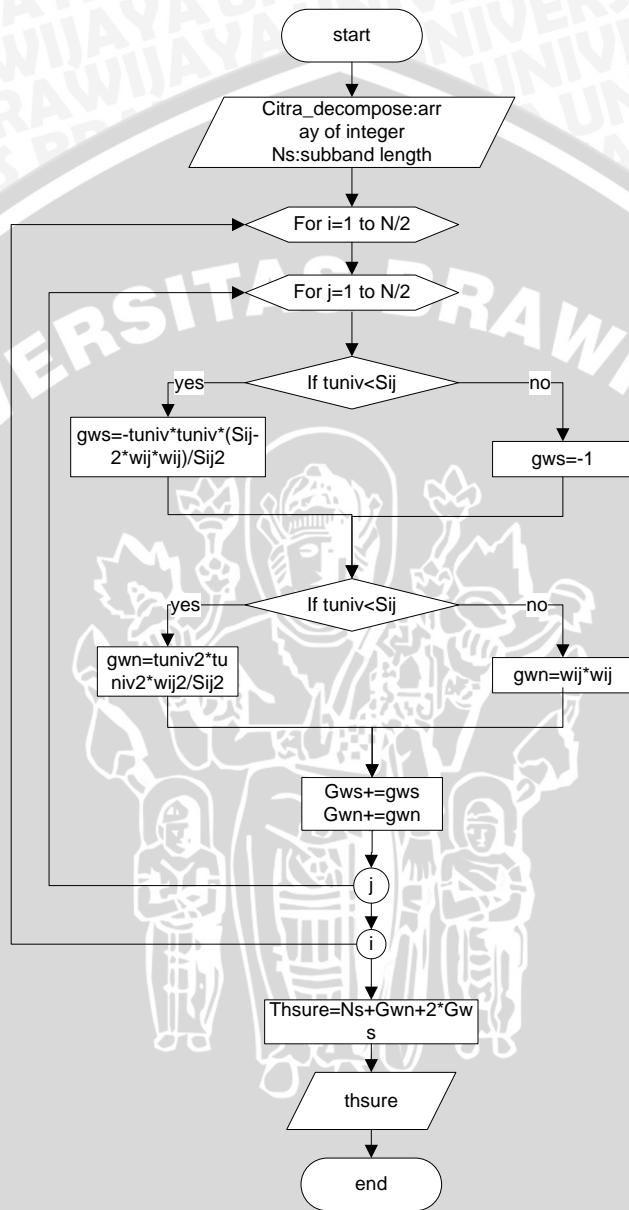
Gambar 3.5 Diagram Alir Penentuan $\sum \text{neighbour}$ ($S_{i,j}^2$)

Langkah selanjutnya adalah menentukan *Stein's Unbiased Risk Estimator (SURE)* pada tiap subband untuk memperoleh optimal threshold. Berikut adalah langkah-langkahnya :

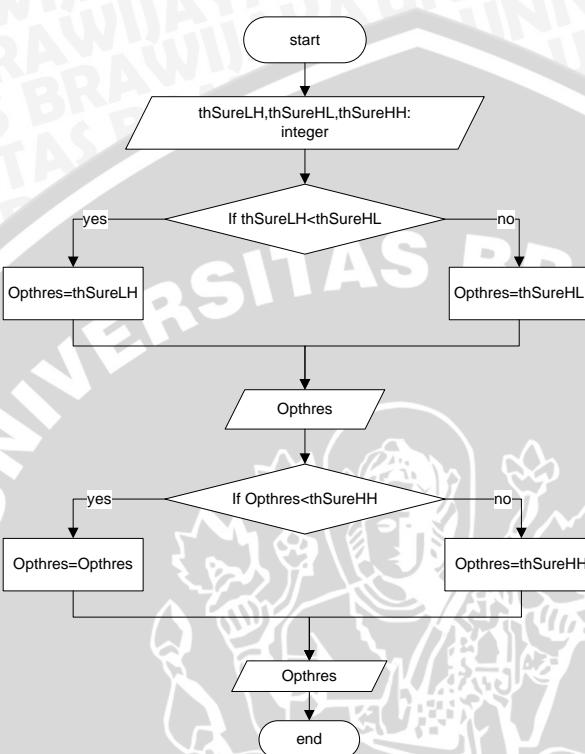
1. Masukan berupa data citra hasil dekomposisi (koefisien wavelet)
2. Melakukan perulangan data sebanyak ukuran tiap subband
3. Menjumlah $\sum gwn$ dengan perhitungan $(-\lambda^2 * \frac{S_{i,j}^2 - 2w_{i,j}^2}{S_{i,j}^4})$ jika $(\lambda^2 < S_{i,j}^2)$, dan -1 jika $(\lambda^2 > S_{i,j}^2)$
4. Menjumlah ∂gn dengan perhitungan $\frac{\lambda^2}{S_{i,j}^4} * w_{i,j}^2$ jika $(\lambda^2 < S_{i,j}^2)$, dan $w_{i,j}^2$ jika $(\lambda^2 > S_{i,j}^2)$
5. Menghitung nilai threshold SURE (λ^s) dengan menjumlah panjang subband (N_s) + $\sum gwn + \partial gn$
6. Hasil keluaran berupa threshold SURE (λ^s)

Diagram perhitungan threshold sure terdapat pada Gambar 3.6.

Setelah nilai threshold SURE tiap subband diketahui, maka untuk memperoleh nilai optimal threshold, maka threshold SURE tiap subband diabndingkan dan diambil nilai paling minimum. Diagram perbandingan threshold SURE untuk memperoleh SURE threshold terdapat pada Gambar 3.7.



Gambar 3.6 Diagram Alir Perhitungan Threshold Sure

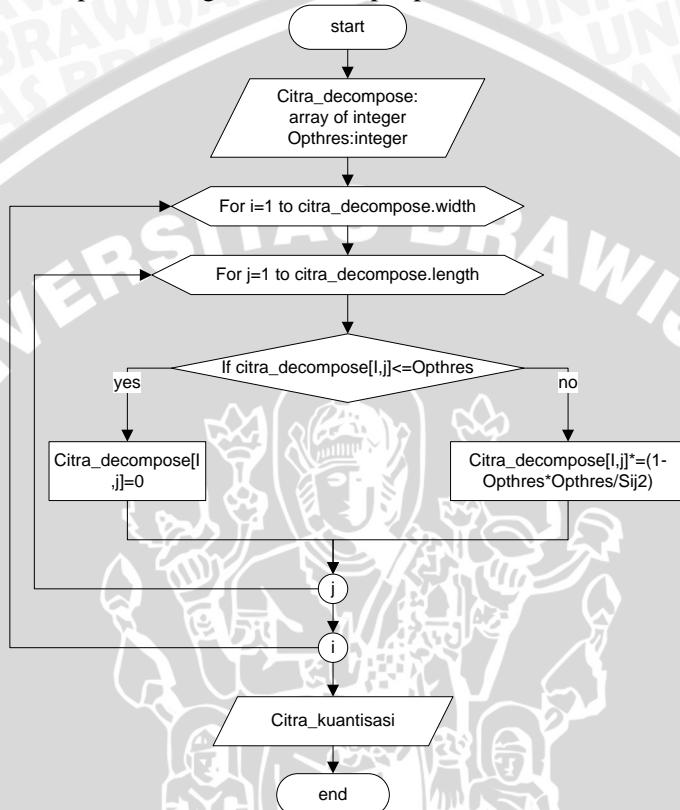


Gambar 3.7 Diagram Alir Perbandingan Threshold SURE

Setelah nilai optimal threshold dihitung, dapat dilakukan proses soft thresholding dengan menggunakan Neighshrink. Berikut adalah langkah neighshrink :

1. Masukan berupa data citra hasil dekomposisi
2. Melakukan perulangan sepanjang ukuran citra hasil dekomposisi
3. Membandingkan nilai koefisien wavelet dengan λ^s
4. Merubah nilai koefisien wavelet dengan 0 jika nilai koefisien wavelet dibawah λ^s , dan merubah nilai koefisien wavelet dengan mengalikan nilai koefisien wavelet dengan $1 - \frac{\lambda^s}{S_{i,j}^2}$ jika nilai koefisien wavelet diatas λ^s
5. Keluaran berupa citra hasil kuantisasi

Diagram alir proses Neighshrink terdapat pada Gambar 3.8.



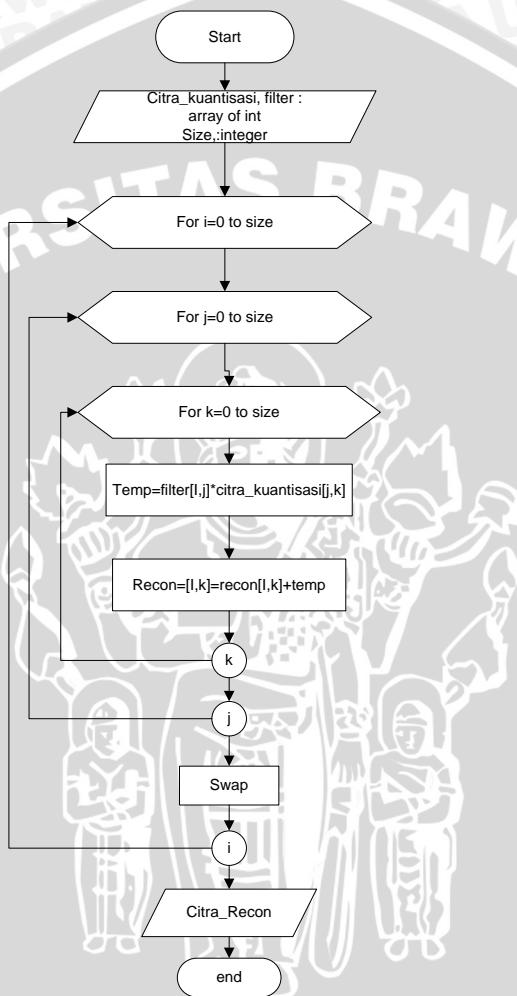
Gambar 3.8 Diagram Alir Neighshrink

3.2.3 Invers Haar Wavelet Transform

Proses invers transformasi wavelet Haar merupakan proses kebalikan dari proses dekomposisi. Pada proses ini citra hasil dekomposisi dikalikan dengan invers dari matrik filter. Langkah-langkah yang dilakukan adalah sebagai berikut :

1. Memasukkan citra hasil perhitungan neighshrink
2. Mengalikan matrik citra dengan invers matrik filter
3. Melakukan swap kolom
4. Mengalikan invers matrik filter dengan matrik citra
5. Melakukan swap baris

Proses invers Haar Wavelet Transform ditunjukkan pada Gambar 3.9.



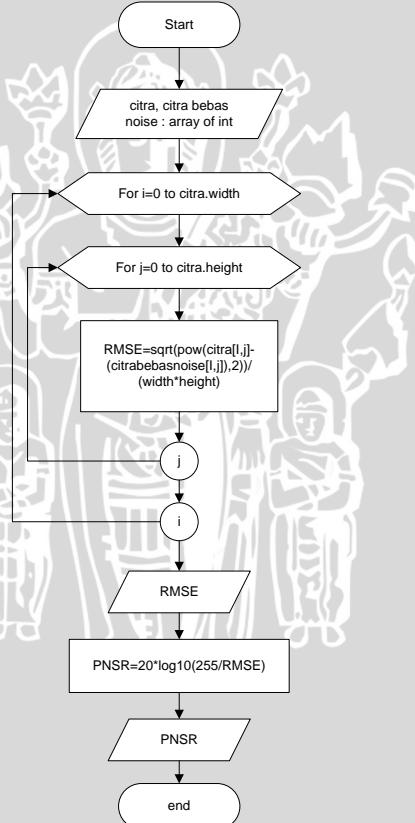
Gambar 3.9 Diagram Alir Proses Rekonstruksi Wavelet Haar

3.2.4 Perancangan Pengujian

Pengujian kesalahan dilakukan dengan menghitung PNSR pada persamaan 2.10. langkahnya adalah sebagai berikut :

1. Memasukkan citra hasil rekonstruksi dan citra bebas noise
2. Melakukan perulangan sepanjang ukuran citra
3. Menghitung akar dari pangkat hasil pengurangan nilai hasil rekonstruksi dengan nilai citra bebas noise yang dibagi dengan ukuran citra (RMSE)
4. Menghitung perkalian 20 dengan \log_{10} dari 255 dibagi dengan nilai RMSE
5. Keluaran berupa nilai RMSE dan PNSR.

Langkah perhitungan PNSR ditunjukkan pada Gambar 3.10.



Gambar 3.10 Diagram Alir Proses PNSR

3.3 Perhitungan Manual

3.3.1 Proses Dekomposisi Citra

Misal terdapat citra berukuran 8x8 dengan nilai seperti pada Tabel 3.1 :

Tabel 3.1 Nilai Citra Asli

(i,j)	1	2	3	4	5	6	7	8
1	79	81	82	91	101	102	82	69
2	71	81	74	90	78	90	70	69
3	74	81	75	85	73	79	81	83
4	82	86	83	82	74	78	86	99
5	92	92	93	81	84	85	80	93
6	99	96	100	83	85	83	85	91
7	107	104	108	87	92	80	96	95
8	116	125	122	96	96	83	101	108

Langkah dekomposisi citra adalah sebagai berikut :

1. Menentukan matriks filter yang ditunjukkan Tabel 3.2 :

Tabel 3.2 Nilai Matrik Filter

(i,j)	1	2	3	4	5	6	7	8
1	0,707	0,707	0	0	0	0	0	0
2	0,707	-0,71	0	0	0	0	0	0
3	0	0	0,707	0,707	0	0	0	0
4	0	0	0,707	-0,71	0	0	0	0
5	0	0	0	0	0,707	0,707	0	0
6	0	0	0	0	0,707	-0,71	0	0
7	0	0	0	0	0	0	0,707	0,707
8	0	0	0	0	0	0	0,707	-0,71

2. Melakukan perkalian matriks filter LH dengan matriks citra. Hasilnya berupa nilai dekomposisi citra (koefisien wavelet) yang ditunjukkan pada Tabel 3.3 :

Tabel 3.3 Hasil Perkalian Baris Matrik Filter dengan Kolom Citra

(i,i)	1	2	3	4	5	6	7	8
1	106,07	114,55	110,31	127,99	126,57	135,76	107,48	97,581
2	5,6569	0	5,6569	0,7071	16,263	8,4853	8,4853	0
3	110,31	118,09	111,72	118,09	103,94	111,02	118,09	128,69
4	-5,657	-3,536	-5,657	2,1213	-0,707	0,7071	-3,536	-11,31
5	135,06	132,94	136,47	115,97	119,5	118,79	116,67	130,11
6	-4,95	-2,828	-4,95	-1,414	-0,707	1,4142	-3,536	1,4142
7	157,68	161,93	162,63	129,4	132,94	115,26	139,3	143,54
8	-6,364	-14,85	-9,899	-6,364	-2,828	-2,121	-3,536	-9,192

Keterangan :

- Warna hitam merupakan nilai hasil aproksimasi (dikenai lowpass filter)
 - Warna merah merupakan nilai hasil detail (dikenai highpass filter)
3. Mengumpulkan nilai aproksimasi pada bagian atas dan detail pada bagian bawah. Pengelompokan hasil Aproksimasi dan Detail ditunjukkan pada Tabel 3.4.

Tabel 3.4 Hasil Pengelompokan Aproksimasi dan Detail

(i,j)	1	2	3	4	5	6	7	8
1	106,07	114,55	110,31	127,99	126,57	135,76	107,48	97,581
2	110,31	118,09	111,72	118,09	103,94	111,02	118,09	128,69
3	135,06	132,94	136,47	115,97	119,5	118,79	116,67	130,11
4	157,68	161,93	162,63	129,4	132,94	115,26	139,3	143,54
5	5,6569	0	5,6569	0,7071	16,263	8,4853	8,4853	0
6	-5,657	-3,536	-5,657	2,1213	-0,707	0,7071	-3,536	-11,31
7	-4,95	-2,828	-4,95	-1,414	-0,707	1,4142	-3,536	1,4142
8	-6,364	-14,85	-9,899	-6,364	-2,828	-2,121	-3,536	-9,192

4. Melakukan perkalian matriks dekomposisi LH dengan matriks hasil perkalian baris pada poin 3 dan mengelompokkan hasil aproksimasi dan detail. Hasil pengelompokan tersebut ditunjukkan oleh Tabel 3.5 :

Tabel 3.5 Hasil Pengelompokan bagian Aproksimasi dan Detail dari Perkalian Kolom Matrik Filter dengan Baris Citra

(i,j)	1	2	3	4	5	6	7	8
1	156	168,5	185,5	145	-6	-12,5	-6,5	7
2	161,5	162,5	152	174,5	-5,5	-4,5	-5	-7,5
3	189,5	178,5	168,5	174,5	1,5	14,5	0,5	-9,5
4	226	206,5	175,5	200	-3	23,5	12,5	-3
5	4	4,5	17,5	6	4	3,5	5,5	6
6	-6,5	-2,5	0	-10,5	-1,5	-5,5	-1	5,5
7	-5,5	-4,5	0,5	-1,5	-1,5	-2,5	-1,5	-3,5
8	-15	-11,5	-3,5	-9	6	-2,5	-0,5	4

Hasil proses dekomposisi 1 tingkat adalah hasil pada poin 4.

Hasil tersebut dapat dilihat pembagian subband seperti pada Tabel 3.6 :

Tabel 3.6 Hasil Dekomposisi Tingkat 1

(i,j)	1	2	3	4	5	6	7	8
1	156	168,5	185,5	145	-6	-12,5	-6,5	7
2	161,5	162,5	152	174,5	-5,5	-4,5	-5	-7,5
3	189,5	178,5	168,5	174,5	1,5	14,5	0,5	-9,5
4	226	206,5	175,5	200	-3	23,5	12,5	-3
5	4	4,5	17,5	6	4	3,5	5,5	6
6	-6,5	-2,5	0	-10,5	-1,5	-5,5	-1	5,5
7	-5,5	-4,5	0,5	-1,5	-1,5	-2,5	-1,5	-3,5
8	-15	-11,5	-3,5	-9	6	-2,5	-0,5	4

Keterangan :

- Warna hitam adalah bagian aproksimasi
- Warna biru adalah bagian detail vertical
- Warna orange adalah bagian detail horizontal
- Warna merah adalah bagian detail diagonal

Sedangkan Dekomposisi 3 Tingkat disajikan pada Tabel 3.7 :

Tabel 3.7 Hasil Dekomposisi 3 Tingkat

(i,j)	1	2	3	4	5	6	7	8
1	706,13	18,375	-6,75	9	-6	-12,5	-6,5	7
2	-53,38	-22,63	15,25	-15,25	-5,5	-4,5	-5	-7,5
3	0,25	2	-5,75	31,5	1,5	14,5	0,5	-9,5
4	-32,25	-16,25	-4,25	9,25	-3	23,5	12,5	-3
5	4	4,5	17,5	6	4	3,5	5,5	6
6	-6,5	-2,5	0	-10,5	-1,5	-5,5	-1	5,5
7	-5,5	-4,5	0,5	-1,5	-1,5	-2,5	-1,5	-3,5
8	-15	-11,5	-3,5	-9	6	-2,5	-0,5	4

3.3.2 Perhitungan Shrinkage

Sebelum dilakukan proses thresholding, nilai universal threshold harus dihitung terlebih dulu untuk selanjutnya digunakan dalam perhitungan untuk menentukan optimal threshold berdasarkan Stein Unbiased Risk Estimated (SURE). Perhitungan nilai universal threshold dimulai dengan menghitung nilai estimator dengan menggunakan median absolute deviation dengan persamaan 2.7.

$$\sigma = \frac{MAD(|ws|)}{0,6745} = \frac{1,65625}{0,6745} = 2,45552$$

Setelah nilai estimator ditemukan, maka universal threshold dapat dihitung dengan persamaan 2.3.

$$\lambda = \sqrt{2 * \sigma^2 * \log n} = \sqrt{2 * 2,4552^2 * \log 64} = 4,66702$$

Selanjutnya, menghitung jumlah kuadrat tetangga dari koefisien wavelet pada tiap subband. Berikut table hasil perhitungan jumlah kuadrat tetangga dari masing-masing koefisien wavelet :

Jumlah kuadrat tetangga pada level 1 ditunjukkan pada Tabel 3.8 :

Tabel 3.8 Jumlah kuadrat tetangga pada level 1

	LH1	HL1	HH1
S1,1	84,75	242,75	60,75
S1,2	391	310	92
S1,3	479	349	140
S1,4	452,5	172,5	97,5
S2,1	135,25	455,25	69,25
S2,2	441,75	522,75	102,75
S2,3	501,75	649,75	160,75
S2,4	455	263	112
S3,1	456,25	824,25	83,25
S3,2	468,75	1005,8	86,75
S3,3	364,75	1119,8	104,75
S3,4	206	337	62
S4,1	407,75	773,75	50,75
S4,2	420,25	930,25	53,25
S4,3	248,25	1018,3	43,25
S4,4	95,75	255,75	30,75
84,75	391	479	452,5
135,25	441,75	501,75	455
456,25	468,75	364,75	206
407,75	420,25	248,25	95,75

Jumlah kuadrat tetangga pada level 2 ditunjukkan pada Tabel 3.9 :

Tabel 3.9 Jumlah kuadrat tetangga level 2

	LH2	HL2	HH2
S1,1	1308,2	591,69	1128,9
S1,2	1308,2	591,69	1128,9
S1,3	1308,2	591,69	1128,9
S1,4	1308,2	591,69	1128,9

Jumlah kuadrat pada level 3 ditunjukkan pada Tabel 3.10 :

Tabel 3.10 Jumlah kuadrat tetangga level 3

	LH3	HL3	HH3
S1,1	2848,9	337,64	511,89

Langkah selanjutnya adalah menghitung nilai SURE dengan menghitung terlebih dahulu nilai $\frac{\partial g_n}{\partial w_n}$ dan $\|g_n(w_n)\|_2^2$ masing-masing koefisien wavelet sesuai persamaan 2.9 dan 2.10. Hasil perhitungan nilai $\frac{\partial g_n}{\partial w_n}$ dan $\|g_n(w_n)\|_2^2$ pada masing-masing subband LH, HL dan HH ditunjukkan oleh Tabel 3.11, Tabel 3.12 dan Tabel 3.13 :

Tabel 3.11 Nilai $\frac{\partial g_n}{\partial w_n}$ dan $\|g_n(w_n)\|_2^2$ pada subband LH

LH	$\frac{\partial g_n}{\partial w_n}$	$\ g_n(w_n)\ _2^2$
	-0,1599639	1,056814
	-0,049936	0,062839
	0,01267328	0,633232
	-0,0404759	0,083411
	-0,0604283	1,095746
	-0,0479111	0,015194
	-0,0434102	0
	-0,0246717	0,252647
	-0,0414089	0,068941
	-0,0424516	0,043722
	-0,0596332	0,000891
	-0,1034236	0,025154
	0,005535	0,642026
	-0,0192084	0,355253
	-0,0790794	0,094301
	0,15739365	4,191459
Σ	-0,5964001	8,62163

Tabel 3.12 Nilai $\frac{\partial g_n}{\partial w_n}$ dan $\|g_n(w_n)\|_2^2$ pada subband HL

	$\frac{\partial g_n}{\partial w_n}$	$\ g_n(w_n)\ _2$
HL	-0,06311	0,289829
	0,00057	0,771355
	-0,0473	0,164564
	-0,05453	0,781224
	-0,04149	0,069244
	-0,03844	0,035156
	-0,03094	0,028093
	-0,04739	0,385806
	-0,02628	0,001571
	-0,0126	0,098608
	-0,01944	9,46E-05
	-0,03001	0,377003
	-0,0275	0,007132
	0,00439	0,302757
	-0,01483	0,071494
	-0,07917	0,065278
Σ	-0,52808	3,449209

Tabel 3.13 Nilai $\frac{\partial g_n}{\partial w_n}$ dan $\|g_n(w_n)\|_2^2$ pada subband HH

	$\frac{\partial g_n}{\partial w_n}$	$\ g_n(w_n)\ _2^2$
HH	-0,16968	2,0567675
	-0,1737	0,6866228
	-0,08835	0,7321956
	-0,05843	1,7965986
	-0,29409	0,2225875
	-0,08716	1,3593131
	-0,13381	0,0183593
	-0,08942	1,1440556
	-0,24749	0,1540181
	-0,2149	0,3940024
	-0,199	0,0972819
	-0,21248	1,5118562
	0,179707	6,6311399
	-0,31302	1,0456795
	-0,49779	0,0634053
	0,028794	8,0276333
Σ	-2,57082	25,941517

Setelah jumlah masing-masing nilai $\frac{\partial g_n}{\partial w_n}$ dan $\|g_n(w_n)\|_2^2$ diketahui, selanjutnya menghitung nilai SURE dengan persamaan 2.8 sebagai berikut :

$$SURE = N_s + \sum_n \|g_n(w_n)\|_2^2 + 2 \sum_n \frac{\partial g_n}{\partial w_n}$$

$$SURE LH = 16 + 8,62163 + 2 * -0,5964001 = 23,4288297$$

$$SURE HL = 16 + 3,449209 + 2 * -0,52808 = 18,393$$

$$SURE HH = 16 + 25,941517 + 2 * -2,57082 = 36,79988$$

Nilai SURE pada masing-masing subband telah dihitung dan untuk menentukan nilai optimal threshold, maka dipilih nilai

threshold yang paling kecil yaitu 18,393 sehingga SURE threshold yang dihasilkan adalah $\lambda^s = 18,393$.

Langkah selanjutnya adalah menerapkan aturan soft shrinkage pada koefisien wavelet hasil dekomposisi dengan menggunakan nilai $\lambda^s = 18,393$. Hasil dari proses thresholding dengan menggunakan SURE Threshold pada aturan soft Shrinkage dengan aturan Neighshrink ditunjukkan pada Tabel 3.14 dan Tabel 3.15 :

Tabel 3.14 Soft Shrinkage dengan menggunakan Sure threshold

(i,j)	1	2	3	4	5	6	7	8
1	706,13	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	31,5	0	0	0	0
4	0	0	0	0	0	23,5	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Setelah aturan soft shrinkage diterapkan, aturan neighshrink dapat diterapkan dengan mengalikan koefisien wavelet yang memiliki nilai diatas $\lambda^s = 18,393$ harus di-shrink terlebih dahulu sebelum dilakukan proses rekonstruksi. Berikut contoh perhitungan neighshrink dengan penggabungan persamaan 2.5 dan 2.6 pada koefisien wavelet pada posisi 3,4 dan hasil neighshrink adalah:

$$\begin{aligned}
 w_{3,4} &= w_{3,4} * \left(1 - \frac{\lambda^s}{S_{3,4}} \right) \\
 &= 31,5 * \left(1 - \frac{18,393^2}{930,25} \right) \\
 &= 22,061
 \end{aligned}$$

Tabel 3.15 Neighshrink dengan menggunakan optimal threshold Sure

(i,j)	1	2	3	4	5	6	7	8
1	706,13	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	22,061	0	0	0	0
4	0	0	0	0	0	14,954	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Untuk penggunaan Universal Threshold, dengan menggunakan nilai $\lambda^s = 4,66702$, hasil dari proses thresholding pada aturan soft Shrinkage dengan aturan Neighshrink ditunjukkan pada Tabel 3.16 dan Tabel 3.17 :

Tabel 3.16 Soft Shrinkage dengan menggunakan universal threshold

(i,j)	1	2	3	4	5	6	7	8
1	706,13	18,38	0	9	0	0	0	7
2	0	0	15,25	0	0	0	0	0
3	0	0	0	31,5	0	14,5	0	0
4	0	0	0	9,25	0	23,5	12,5	0
5	0	0	17,5	6	0	0	5,5	6
6	0	0	0	0	0	0	0	5,5
7	0	0	0	0	0	0	0	0
8	0	0	0	0	6	0	0	0

Dan hasil dari soft thresholding tersebut, koefisien wavelet yang memiliki nilai diatas $\lambda=4,66702$ harus di-shrink terlebih dahulu sebelum dilakukan proses rekonstruksi seperti pada Sure threshold. Berikut contoh perhitungan neighshrink pada koefisien wavelet pada posisi 3,4 dan hasil neighshrink adalah:

$$\begin{aligned}
 w_{3,4} &= w_{3,4} * \left(1 - \frac{\lambda^2}{S_{3,4}}\right) \\
 &= 31,5 * \left(1 - \frac{4,66702^2}{930,25}\right) \\
 &= 30,89
 \end{aligned}$$

Tabel 3.17 Neighshrink dengan menggunakan universal threshold

(i,j)	1	2	3	4	5	6	7	8
1	706,13	17,19	0	9	0	0	0	6,12
2	0	0	14,69	0	0	0	0	0
3	0	0	0	30,89	0	14,22	0	0
4	0	0	0	9,072	0	22,95	12,2	0
5	0	0	16,7	5,711	0	0	4,64	4,66
6	0	0	0	0	0	0	0	4,43
7	0	0	0	0	0	0	0	0
8	0	0	0	0	3,42	0	0	0

3.3.3 Rekonstruksi

Proses rekonstruksi dilakukan berkebalikan dengan proses dekomposisi. Proses dekomposisi dilakukan dengan mengalikan hasil perhitungan shrinkage pada proses sebelumnya dengan invers matrik wavelet Haar. Invers matrik waveler Haar dan hasil rekonstruksi ditunjukkan pada Tabel 3.18 :

Tabel 3.18 Invers Matrik filter LH

LH*	0,7071	0,7071	0	0	0	0	0	0
	0,7071	-0,7071	0	0	0	0	0	0
	0	0	0,7071	0,7071	0	0	0	0
	0	0	0,7071	-0,707	0	0	0	0
	0	0	0	0	0,7071	0,7071	0	0
	0	0	0	0	0,7071	-0,7071	0	0
	0	0	0	0	0	0	0,7071	0,7071
	0	0	0	0	0	0	0,7071	-0,7071

Sedangkan citra hasil rekonstruksi dengan menggunakan SURE Threshold dan Universal Threshold ditunjukkan pada Tabel 3.19 dan 3.20 :

Tabel 3.19 Citra Hasil rekonstruksi dengan Sure Threshold

(i,j)	1	2	3	4	5	6	7	8
1	88,27	88,27	88,27	88,27	88,27	88,27	88,27	88,27
2	88,27	88,27	88,27	88,27	88,27	88,27	88,27	88,27
3	88,27	88,27	88,27	88,27	88,27	88,27	88,27	88,27
4	88,27	88,27	88,27	88,27	88,27	88,27	88,27	88,27
5	88,27	88,27	88,27	88,27	103	80,97	84,53	84,53
6	88,27	88,27	88,27	88,27	103	80,97	84,53	84,53
7	88,27	88,27	88,27	88,27	84,53	84,53	92	92
8	88,27	88,27	88,27	88,27	84,53	84,53	92	92

Tabel 3.20 Citra Hasil rekonstruksi dengan Universal Threshold

(l,j)	1	2	3	4	5	6	7	8
1	96,62	79,91	90,59	85,94	99,2	84,46	90,1	79,3
2	79,91	96,62	85,94	90,59	93,5	90,18	85,4	84
3	95,61	80,92	88,27	88,27	84,7	84,71	94	89,6
4	95,61	80,92	88,27	88,27	84,7	84,71	89,6	94
5	88,27	88,27	88,27	88,27	109	78,56	82,5	82,5
6	88,27	88,27	88,27	88,27	109	78,56	82,5	82,5
7	88,27	88,27	96,09	83,86	87,1	77,99	94	94
8	88,27	88,27	92,67	80,44	87,1	77,99	94	94

3.3.4 Perhitungan Peak Signal to Noise Ratio (PSNR)

PNSR digunakan sebagai parameter evaluasi hasil denoising yang dilakukan. Nilai MSE harus dihitung terlebih dahulu agar nilai PNSR dapat diperoleh. Perhitungan nilai MSE dan PSNR pada penggunaan SURE Threshold dan Universal Threshold masing-masing ditunjukkan pada poin 1 dan 2 :

1. Metode Neighshrink dengan Sure Threshold

$$\begin{aligned}MSE &= \frac{\sum [f(i,j) - F(i,j)]^2}{N^2} \\&= \frac{9864,061083}{64} \\&= 154,126\end{aligned}$$

$$\begin{aligned}PSNR &= 10 \log_{10} \left(\frac{255^2}{RMSE} \right) \\&= 10 \log_{10} \left(\frac{255^2}{12,4147} \right) \\&= 37,191425\end{aligned}$$

Nilai MSE dan PSNR dengan menggunakan Sure Threshold adalah 154,126 dan 37,191425dB.

2. Metode Neighshrink dengan universal Threshold

$$\begin{aligned}MSE &= \frac{\sum [f(i,j) - F(i,j)]^2}{N^2} \\&= \frac{9873,2291}{64} \\&= 154,2692\end{aligned}$$

$$\begin{aligned}PSNR &= 10 \log_{10} \left(\frac{255^2}{RMSE} \right) \\&= 10 \log_{10} \left(\frac{255^2}{12,420515} \right) \\&= 37,189407\end{aligned}$$

Nilai MSE dan PSNR dengan menggunakan Universal Threshold adalah 154,2692 dan 37,189407 dB.

Dari dua hasil diatas dapat disimpulkan bahwa metode neighshrink dengan menggunakan optimal threshold lebih baik dari metode neighshrink dengan menggunakan universal threshold.

3.4 Pengujian

3.4.1 Pengujian Kesalahan

Pengujian kesalahan dilakukan dengan menghitung RMSE dan membandingkan antara nilai Neighgshrink dengan universal threshold dan SURE threshold pada masing-masing citra. Keluaran dari hasil pengujian dengan menggunakan RMSE ini berupa nilai real. Rancangan pengujian MSE tampak pada Tabel 3.21.

Tabel 3.21 Rancangan RMSE Denoising Citra

No	Citra	Tingkat Kesalahan (RMSE)	
		Threshold (λ)	
		Universal threshold	Sure threshold
1	Citra 1		
2	Citra 2		
...			

3.4.2 Pengujian Rasio

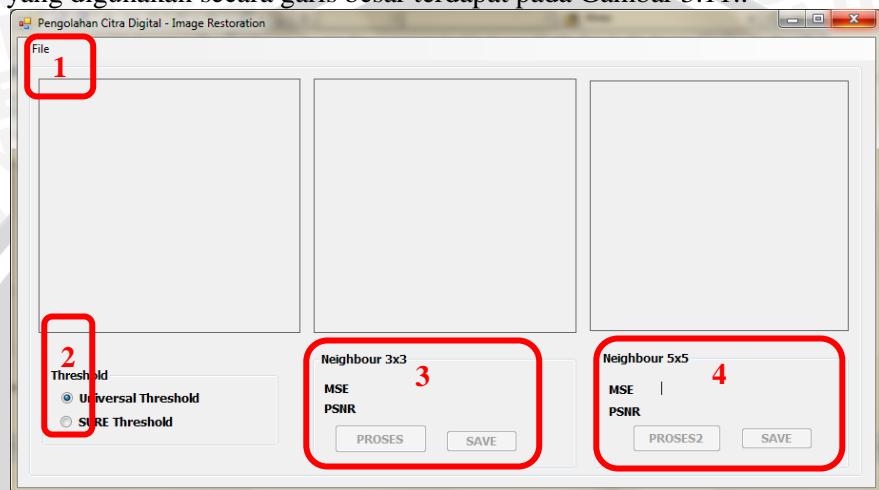
Pengujian rasio digunakan untuk melakukan pengujian terhadap nilai citra hasil perbaikan dengan citra asli. Masukan dari pengujian berupa nilai pixel dari citra asli dan citra hasil perbaikan. Keluaran dari hasil pengujian ini berskala decibel (dB). Rancangan hasil pengujian rasio menggunakan RMSE tampak pada Tabel 3.22.

Tabel 3.22 Rancangan PSNR Denoising Citra

No	Citra	Tingkat Kesalahan (PSNR)	
		Threshold (λ)	
		Universal threshold	Sure threshold
1	Citra 1		
2	Citra 2		
...			

3.5 Rancangan Antarmuka

Rancangan antarmuka yang meliputi proses dekomposisi, thresholding, neighshrink dan rekonstruksi serta pengujian metode yang digunakan secara garis besar terdapat pada Gambar 3.11.:



Gambar 3.11 Rancangan Antarmuka

Keterangan gambar :

1. Menu File : berisi menu input citra dan menutup form
2. Pilihan threshold : terdiri dari pilihan universal threshold dan SURE threshold
3. Neighbouring 3x3 :
Tombol PROSES : untuk menjalankan program meliputi dekomposisi, thresholding (universal atau SURE), dan rekonstruksi dengan ukuran neighbouring window 3x3
4. Neighbouring 5x5 :
Tombol SAVE : untuk menyimpan hasil proses Thresholding dengan ukuran neighbouring window 3x3
4. Neighbouring 5x5 :
Tombol PROSES : untuk menjalankan program meliputi dekomposisi, thresholding (universal atau

UNIVERSITAS BRAWIJAYA

Tombol SAVE

SURE), dan rekonstruksi dengan ukuran neigbouring window 5x5 : untuk menyimpan hasil proses Thresholding dengan ukuran neigbouring window 3x3

UNIVERSITAS BRAWIJAYA



BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Bab ini akan lebih lanjut menjelaskan tentang proses-proses yang telah dirancang pada bab tiga. Hal-hal yang akan dibahas meliputi implementasi proses-proses yang telah dirancang, tampilan *interface*, dan bagian-bagian *source code* yang ada dalam implementasi program. Pada bab ini juga dilakukan analisa terhadap hasil yang diperoleh dari proses denoising citra uji.

4.1 Lingkungan Implementasi

Proses implementasi merupakan tahapan penerapan rancangan pada bahasa pemrograman yang dapat dimengerti oleh komputer. Lingkungan implementasi yang akan dijelaskan pada bab ini meliputi lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan perangkat lunak ini adalah:

1. Processor Intel Core i3 2,53 GHz.
2. RAM 2 GB.
3. Harddisk dengan kapasitas 300 GB.
4. Monitor 14”.
5. Keyboard.
6. Mouse.

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi *Microsoft Windows 7 Ultimate*.
2. Microsoft Visual Studio Express 2008

4.2 Implementasi Program

4.2.1 Implementasi Antarmuka

Implementasi antarmuka terdiri dari satu form utama. Untuk dapat melaksanakan proses restorasi, maka terlebih dahulu dipilih citra *noise* yang telah diperoleh. Setelah citra muncul pada panel yang telah ditentukan, kemudian proses restorasi dapat dilakukan. Hasil citra *output* selama implementasi proses denoising akan muncul pada panel citra di panel utama. Implementasi antarmuka ditampilkan pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka

Untuk melakukan proses denoising terhadap sebuah inputan gambar yang ada pada *picturebox* yang ditunjukkan oleh nomor 2, terlebih dahulu dipilih inputan dengan menggunakan menu file "Open" yang ada pada menu item yang ditunjukkan nomor 1. Setelah inputan ditampilkan, selanjutkan dipilih metode apa yang ingin digunakan untuk mengolah citra tersebut (ditunjukkan oleh nomor 3). Setelah metode terpilih, proses denoising akan dilakukan setelah menekan tombol "Proses" yang ditunjukkan oleh nomor 4 untuk proses dengan ukuran neighbouring window 3x3, sedangkan tombol "Proses2" yang ditunjukkan oleh nomor 6 akan melakukan proses denoising dengan ukuran neighbouring window 5x5 . Setelah

proses selesai, akan ditampilkan gambar hasil denoising pada *picturebox* yang ditandai dengan nomor 5(untuk window 3x3) dan nomor 7(untuk window 5x5). Apabila ingin menyimpan hasil dari proses denoising dapat dilakukan dengan menekan tombol "Save" pada groupbox neighbouring 3x3 seperti pada nomor 4 untuk menyimpan hasil proses dengan ukuran neighbouring window 3x3, sedangkan tombol "Save" pada groupbox neighbouring 5x5 yang ditunjukkan oleh nomor 6 akan melakukan proses penyimpanan dengan ukuran neighbouring window 5x5.

4.2.2 Implementasi Program

Berdasarkan perancangan proses yang terdapat pada Bab 3, maka pada subbab ini akan dijelaskan implementasi proses-proses tersebut. Secara garis besar proses denoising ini terdiri dari 3 tahap utama, diantaranya adalah tahap dekomposisi citra, tahap thresholding dan tahap rekonstruksi.

4.2.2.1 Dekomposisi

Seperti yang dijelaskan pada bab sebelumnya bahwa wavelet Haar digunakan untuk melakukan proses dekomposisi citra. Proses dekomposisi dilakukan melalui proses pertukaran (swapping) baris dan kolom ketika dikenai perkalian dengan matriks filter. Dalam program, proses dekomposisi diberi nama *dekomposisi()*. Proses dekomposisi dengan menggunakan wavelet Haar ditunjukkan pada *Source code 4.1*.

Dekomposisi
<pre>private double[,] dekomposisi(double[,] filter, byte[,] bil, int size) { double[,] hasil = new double[size, size]; double[,] baris = new double[size, size]; double z = 0; parent.Text = "Do Decompostion.."; for (int i = 0; i < size; i++) { for (int j = 0; j < size; j++) { for (int k = 0; k < size; k++) { z = filter[i, k] * bil[k, j]; hasil[i, j] = hasil[i, j] +</pre>

```

z;
        }
    }
    double[,] decomp = new double[size,
size];
    decomp = swapBaris(hasil, size);
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            for (int k = 0; k < size; k++)
            {
                z = decomp[i, k] * filter[k,
j];
                baris[i, j] = baris[i, j] +
z;
            }
        }
    }
    double[,] decompose = new double[size,
size];
    decompose = swapKolom(baris, size);
    return decompose;
} //end of dekomposisi

```

Source code 4.1 Proses Dekomposisi.

4.2.2.2 Menghitung Universal Threshold

Karena nilai noise level pada citra uji tidak diketahui, maka diperlukan perhitungan untuk memperkirakan nilai noise level yang terkandung pada citra uji. Perhitungan yang digunakan menggunakan Median Absolut Deviation untuk memperoleh perkiraan noise level yang akan digunakan untuk menentukan nilai universal threshold. Proses perhitungan untuk mendapatkan universal threshold dengan menggunakan MAD ditunjukkan pada Source code 4.2.

```

getUniversal Threshold
private    double getUnivThreshold(double[,] d2, int
size)
{
    parent.Text      =      "Get      Universal
Threshold..";
    double[] tamp = new double[(size / 2) *
(size / 2)];
    int v = 0;
    for (int i = size / 2; i < size; i++)
    {
        for (int j = size / 2; j < size; j++)
        {
            double temp;
            temp = d2[i, j];
            tamp[v] = temp;
            v++;
            //Application.DoEvents();
        }
    }
    int a = size / 2;
    int t = a * a;
    double[] num = new double[t];
    num = urutkan(tamp, t);
    double median, MAD, UnThreshold;
    median=(num[t / 2]+num[t / 2 - 1]) / 2;

    double[] sel = new double[t];
    for (int i = 0; i < t; i++)
    {
        sel[i] = num[i] - median;
        if (sel[i] < 0)
            sel[i] = sel[i] * (-1);
    }

    double[] rawMAD = new double[t];
    rawMAD = urutkan(sel, t);

    MAD = (rawMAD[t/2]+rawMAD[t/2 - 1])/ 2;
    double estimator;
    estimator = MAD / 0.6745;
    UnThreshold      =      Math.Sqrt(2      *
Math.Pow(estimator, 2) * Math.Log(size * size));
    return UnThreshold;
}

```

Source code 4.2. Proses GetUniversal Threshold.

4.2.2.3 Perhitungan Jumlah Kuadrat Tetangga

Jumlah kuadrat tetangga masing-masing koefisien wavelet diperlukan untuk penentuan faktor *shrinkage* (penyusutan) yang akan digunakan pada proses thresholding berikutnya. Proses perhitungan jumlah kuadrat tetangga ditunjukkan pada *Source code 4.3.*

```
getSumKuadrat
private    double[,] GetSumKuadrat(double[,] d, int
r, int t)
{
    int p = r + 2;
    double[,] m = new double[p, p];
    parent.Text = "Do Get Sum Kuadrat..";
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < p; j++)
        {
            if ((i == 0) || (i == p - 1) ||
(j == 0) || (j == p - 1))
            {
                m[i, j] = 0;
            }
            else
            {
                m[i, j]=d[i - 1, j - 1];
            }
        }
    }
    double[,] sum = new double[r, r];
    int u, v;
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < r; j++)
        {
            u = i + 1;
            v = j + 1;
            sum[i, j] = m[u, v] + m[u - 1,
v - 1] + m[u - 1, v] + m[u - 1, v + 1] + m[u, v +
1] + m[u, v + 1] + m[u + 1, v - 1] + m[u + 1, v] +
m[u + 1, v + 1];
        }
    }
    return sum;
}
```

Source code 4.3. Proses Perhitungan Jumlah Kuadrat Tetangga

4.2.2.4 Perhitungan Faktor Shrinkage

Faktor *shrinkage* merupakan nilai yang dibutuhkan untuk perhitungan nilai yang akan dithreshold pada proses thresholding. Inputan nilai yang dibutuhkan untuk memperoleh faktor shrinkage tiap koefisien wavelet antara lain nilai threshold dan jumlah kuadrat tetangga yang telah diperoleh pada proses sebelumnya. Perolehan nilai faktor shrinkage ditunjukkan pada *Source code 4.4*.

```
Factor
private double[,] Factor(double[,] S, double T,
int size)
{
    parent.Text = "Get Factor..";
    double[,] b = new double[size, size];
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if ((i < size / 2) && (j < size
/ 2))
                b[i, j] = 0;
            else
            {
                b[i, j] = 1 - (T / S[i, j]);
                if(b[i,j]<0)
                    b[i,j]=0;
            }
        }
    }
    return b;
}
```

Source code 4.4. Proses Faktor Shrinkage

4.2.2.5 Perhitungan Shrinkage

Pada proses thresholding dibutuhkan inputan nilai threshold. Nilai threshold digunakan untuk menentukan nilai yang akan dikenai proses thresholding, dimana apabila nilai koefisien wavelet melebihi nilai threshold maka akan dilakukan proses thresholding terhadap koefisien tersebut. Proses Thresholding ditunjukkan pada *Source code 4.5*.

```

Shrinkage
private double[,] Shrink(double[,] dec, double[,] B, double[,] S, double T, int size)
{
    parent.Text = "Get Shrink..";
    double[,] rec = new double[size, size];
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if ((i < size / 2) && (j < size / 2))
                rec[i, j] = dec[i, j];
            else
            {
                if (S[i, j] <= T)
                    rec[i, j] = 0;
                else
                {
                    rec[i, j] = dec[i, j] *
B[i, j];
                }
            }
        }
    }
    //Application.DoEvents();
}
return rec;
}

```

Source code 4.5. Proses Shrinkage(Thresholding)

4.2.2.6 Proses Rekonstruksi

Proses rekonstruksi adalah proses terakhir dalam urutan program. Proses ini berfungsi untuk mengembalikan citra kembali ke domain spatial agar dapat dibaca. Seperti telah dijelaskan dalam proses dekomposisi, baris dan kolom dalam citra telah saling ditukar, dan dalam proses rekonstruksi nantinya baris dan kolom tersebut harus dikembalikan pada posisi semula agar informasi citra dapat dibaca kembali, untuk itu diperlukan proses pembalik. Proses Rekonstruksi dan Swapping ditunjukkan pada *Source code 4.6* dan *Source code 4.7*.

Reconstruction

```
private    double[,] rekonstruksi(double[,] filter,
double[,] rekons, int s, int size)
{
    double[,] hasil = new double[s, s];
    double[,] hasil2 = new double[s, s];
    double[,] kolom = new double[s, s];
    double[,] swap1 = new double[s, s];

    parent.Text = "Reconstruction..";
    kolom = swapReverse(rekons,s,"kolom");
    double z = 0;
    for (int i = 0; i < s; i++)
    {
        for (int j = 0; j < s; j++)
        {
            for (int k = 0; k < s; k++)
            {
                z = kolom[i, k]*filter[k,j];
                swap1[i, j] = swap1[i, j] + z;
            }
        }
    }
    double[,] decomp = new double[s, s];
    decomp = swapReverse(swap1,s,"baris");

    for (int i = 0; i < s; i++)
    {
        for (int j = 0; j < s; j++)
        {
            for (int k = 0; k < s; k++)
            {
                z =filter[i,k]*decomp[k,j];
                hasil2[i,j] = hasil2[i,j]+z;
            }
        }
    }
    double[,] recon =new double[size,size];
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if ((i < s) && (j < s))
                recon[i, j] = hasil2[i, j];
            else
                recon[i, j] = rekons[i, j];
        }
    }
}
```

```
    }
    return recon;
}//end of rekonstruksi
```

Source code 4.6. Proses Rekonstruksi

SwapReverse

```
private static double[,] swapReverse(double[,] c,
int size, string elemen)
{
    double[,] hasill=new double[size,size];
    int half = size / 2;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            int k;
            if (elemen == "baris")
            {
                if (i < half)
                {
                    k = i * 2;
                }
                else
                {
                    k = 2 * (i - half) + 1;
                }
                hasill[k, j] = c[i, j];
            }
            else if (elemen == "kolom")
            {
                if (j < half)
                {
                    k = j * 2;
                }
                else
                {
                    k = 2 * (j - half) + 1;
                }
                hasill[i, k] = c[i, j];
            }
        }
    }
    return hasill;
}
```

Source code 4.7. Proses Swapping

4.2.2.7 Perhitungan Threshold SURE (Stein's Unbiased Risk Estimate)

Stein's Unbiased Risk Estimate digunakan untuk menghitung nilai threshold pada metode SURE Threshold agar diperoleh nilai *MSE* yang paling kecil. Hal yang berpengaruh pada perhitungan ini adalah ukuran subband pada tiap level dekomposisi. Proses perhitungan threshold SURE ditunjukkan *Source code 4.8*.

```
Sure
private static double Sure(double[,] S,
double[,] W, double T, int m)
{
    double t2=T*T,temp1=0,temp2= 0,
temp3 = 0; double sure;
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (S[i, j] > t2)
            {
                temp1 = temp1 + (1 /
S[i, j]) - ((2 * W[i, j] * W[i, j]) / (S[i,
j] * S[i, j]));
                temp2 = temp2 +
((W[i, j] * W[i, j]) / (S[i, j] * S[i, j]));
            }
            else
            {
                temp3 = temp3 + (W[i,
j] * W[i, j]) - 2;
            }
        }
    }
    sure = (m * m) - (2 * t2 * temp1)
+ (t2 * t2 * temp2) + temp3;
    return sure;
} //end of Sure
```

Source code 4.8. Proses SURE

4.2.3 Implementasi Pengujian

Setelah dilakukan proses denoising, akan diperoleh nilai citra keluaran dari masing-masing ukuran window. Nilai keluaran ini digunakan sebagai pembanding antara nilai yang dihasilkan. Berikut proses pengujian kesalahan dengan menggunakan *MSE(Mean Square Error)* dan *PSNR(Peak Signal to Noise Ratio)*. Proses perhitungan nilai MSE dan PSNR ditunjukkan *Source code 4.9*.

```
Calculate
public void calculate()
{
    byte[] ori = getData(Origin);
    byte[] eks = getData(Eks);
    double temp = 0;
    for (int x = 0; x < ori.Length; x++)
    {
        temp += Math.Pow(ori[x] - eks[x], 2);
    }
    MSE = temp / ori.Length;
    if (MSE != 0)
    {
        PSNR = 10 * Math.Log10(Math.Pow(255, 2)
/ MSE);
    }
    else
    {
        PSNR = 0;
    }
}
```

Source code 4.9. Proses Calculate

4.3 Skenario Pengujian

Pada sistem denoising citra untuk mengimplementasikan metode Shrinkage, pengujian dilakukan untuk mencari nilai *PSNR* citra ber-ISO tinggi dengan citra *output* (citra hasil denoising). Pengujian dilakukan pada citra uji dengan *impulse noise* yang tidak diketahui prosentase *noise*-nya. Pada pengujian ini parameter evaluasi yang digunakan adalah nilai *MSE* (*Mean Square Error*) dan juga nilai *PSNR* (*Peak Signal to Noise Ratio*) seperti pada persamaan 2.13 dan 2.14.

4.3.1 Data Pengujian

Citra sebagai data uji diambil penulis dengan menggunakan kamera digital KODAK EASYSHARE C142 DIGITAL CAMERA dengan pengaturan ISO 1600. Citra yang digunakan adalah citra grayscale. Ukuran citra uji adalah 256x256 piksel serta menggunakan format BMP (.bmp). Secara keseluruhan jumlah citra uji adalah 30 buah.

4.3.2 Hasil Pengujian

Metode denoising yang diterapkan pada citra, yang menghasilkan nilai keluaran yang akan dihitung nilai kesalahannya dengan menggunakan *MSE*. Nilai *MSE* akan dihitung terhadap kedua metode pengujian. *MSE* merupakan perbandingan nilai piksel citra hasil perbaikan dengan nilai citra yang mengandung *noise*, selanjutnya nilai *RMSE* digunakan untuk menghitung nilai *PSNR*(dalam dB/desibel) yang menunjukkan rasio kekuatan sinyal citra terhadap *noise*.

Pengujian ini bertujuan untuk memperoleh nilai *PSNR* citra hasil *denoising* setelah metode diterapkan, sehingga dapat diketahui efisiensi metode tersebut dalam merestorasi citra dari *impulse noise* tanpa pengetahuan awal mengenai besarnya *noise* yang terkandung di dalamnya.

Citra uji yang digunakan berjumlah 30 buah namun yang disajikan disini hanya 5 untuk memudahkan pengamatan dan pembahasan. Hasil citra uji dengan menggunakan Universal Threshold pada penggunaan neighbouring window 3x3 dan 5x5 ditunjukkan pada Lampiran 1, sedangkan citra hasil pengujian dengan penggunaan SURE Threshold pada kedua ukuran neighbouring window ditunjukkan pada Lampiran 2, sedangkan

perbandingan pada penggunaan Universal dan SURE Threshold dapat dilihat ditunjukkan pada Lampiran 3. Sedangkan hasil pengolahan citra uji secara visual ditunjukkan pada Lampiran 4 dan 5. Tabel 4.1 menunjukkan contoh perbandingan penggunaan ukuran *neighbouring window* hasil dari pengujian *MSE* dan *PSNR* dengan menggunakan metode *Universal Threshold* dan Tabel 4.2 menunjukkan contoh perbandingan penggunaan ukuran *neighbouring window* hasil dari pengujian *MSE* dan *PSNR* dengan menggunakan *SURE Threshold*. Tabel 4.3 menunjukkan perbandingan hasil pengujian *MSE* dan *PSNR* penggunaan Universal Threshold dan *SURE Threshold*. Sedangkan contoh hasil pengolahan citra secara visual ditampilkan dalam Tabel 4.4.

Tabel 4.1 Hasil pengujian MSE dan PSNR metode Universal Threshold

No	Nama Citra Uji(.bmp)	Window 3x3		Window 5x5	
		MSE	PSNR	MSE	PSNR
1	white leaf_010	55,816235	30,663198	55,816143	30,663205
2	sedotan_008	55,946399	30,653082	55,946262	30,653093
3	teratai_015	59,421467	30,391370	59,421421	30,391373
4	tudung_011	58,672142	30,446484	58,672668	30,446445
5	tusuk_002	37,055557	32,442270	37,055901	32,442230

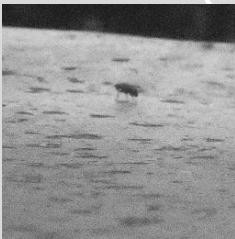
Tabel 4.2 Hasil pengujian MSE dan PSNR metode SURE Threshold

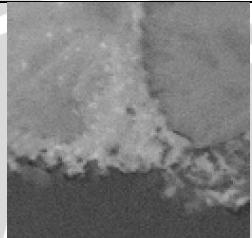
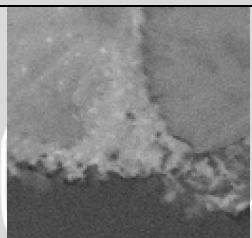
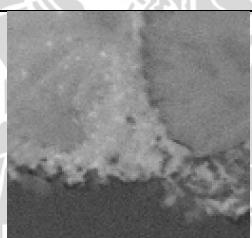
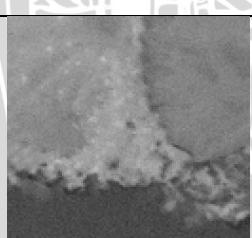
No	Nama Citra Uji(.bmp)	Window 3x3		Window 5x5	
		MSE	PSNR	MSE	PSNR
1	white leaf_010	59,169777	30,409804	59,169777	30,409804
2	sedotan_008	55,975353	30,650835	55,975353	30,650835
3	teratai_015	59,618855	30,376967	59,618855	30,376967
4	tudung_011	62,147392	30,196575	62,147392	30,196575
5	tusuk_002	41,651688	31,934478	41,651688	31,934478

Tabel 4.3 Hasil pengujian MSE dan PSNR metode Neighshrink dengan menggunakan Universal Threshold dan SURE Threshold pada ukuran Neighbouring window 5x5

No	Nama Citra Uji(.bmp)	Universal Threshold Window 5x5		SURE Threshold Window 5x5	
		MSE	PSNR	MSE	PSNR
1	white leaf_010	55,816143	30,663205	56,035801	30,646148
2	sedotan_008	55,946262	30,653093	55,975353	30,650835
3	teratai_015	59,421421	30,391373	59,618855	30,376967
4	tudung_011	58,672668	30,446445	62,147392	30,196575
5	tusuk_002	37,055901	32,442230	41,651688	31,934478

Tabel 4.4 Hasil pengujian secara visual

No.	Citra masukan	Citra Keluaran	Keterangan
1		 	Universal Threshold dengan ukuran window 3x3 Universal Threshold dengan ukuran window 5x5
			SURE Threshold dengan ukuran window 3x3
			SURE Threshold dengan ukuran window 5x5

No.	Citra masukan	Citra Keluaran	Keterangan
2	<i>teratai_015.bmp</i>		Universal Threshold dengan ukuran window 3x3
			Universal Threshold dengan ukuran window 5x5
			SURE Threshold dengan ukuran window 3x3
			SURE Threshold dengan ukuran window 5x5

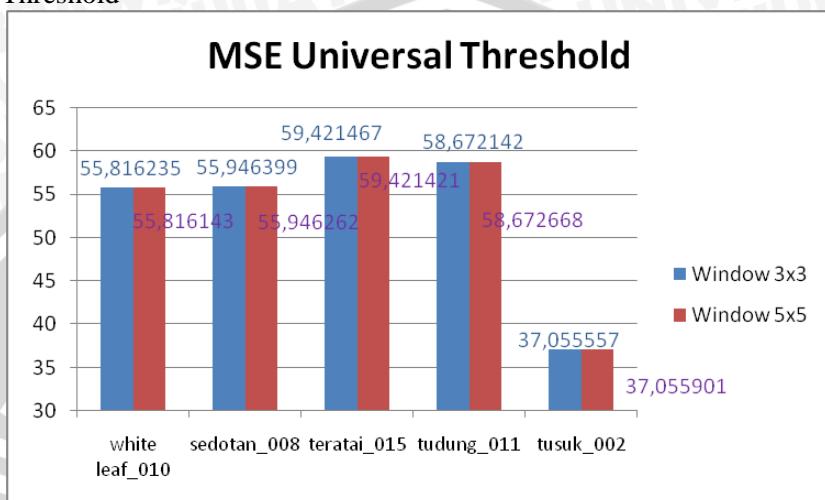
Dari hasil pada Tabel 4.1, pada metode *Neighshrink* dengan menggunakan *Universal Threshold* dengan ukuran *neighbouring window* 3×3 , nilai *MSE* hasil *denoising* citra yang paling besar adalah 59,421467 yaitu pada citra *teratai_015.bmp*, sedangkan nilai *MSE* terkecil dimiliki oleh citra *tusuk_002.bmp* dengan nilai 37,055557. Sedangkan pada ukuran *neighbouring window* 5×5 , nilai *MSE* hasil denoising citra yang paling besar adalah 59,421421 yaitu pada citra uji yang sama yaitu *teratai_015.bmp*, sedangkan nilai *MSE* terkecil dimiliki oleh citra *tusuk_002.bmp* dengan nilai *MSE* 37,055901. Untuk nilai *PSNR* pada penggunaan *Universal Threshold* dengan ukuran *neighbouring window* 3×3 , hasil denoising tertinggi dicapai oleh citra uji *tusuk_002.bmp* yaitu 32,442270 dB dan yang terendah dicapai oleh citra uji *teratai_015.bmp* dengan nilai 30,391370 dB. Sedangkan pada ukuran *neighbouring window* 5×5 , nilai *PSNR* hasil denoising citra yang paling besar adalah 32,442230 dB yaitu pada citra uji yang sama yaitu *tusuk_002.bmp*, sedangkan nilai *PSNR* terkecil dimiliki oleh citra *teratai_015.bmp* dengan nilai 30,391373 dB.

Sedangkan dari hasil Tabel 4.2, pada metode *neighshrink SURE Threshold* dengan ukuran *neighbouring window* 3×3 maupun dengan ukuran *neighbouring window* 5×5 , nilai *MSE* yang paling besar adalah 62,147392 yaitu pada citra *tudung_011.bmp*, sedangkan nilai *MSE* terkecil dimiliki oleh citra *tusuk_002.bmp* dengan nilai 41,651688. Begitu pula pada metode *neighshrink SURE Threshold* dengan ukuran *neighbouring window* 3×3 maupun dengan ukuran *neighbouring window* 5×5 , nilai *PSNR* yang dihasilkan juga sama yaitu nilai *PSNR* yang paling besar adalah 31,934478 dB yaitu pada citra *tusuk_002.bmp*, sedangkan nilai *PSNR* terkecil dimiliki oleh citra *tudung_011.bmp* dengan nilai 30,196575 dB.

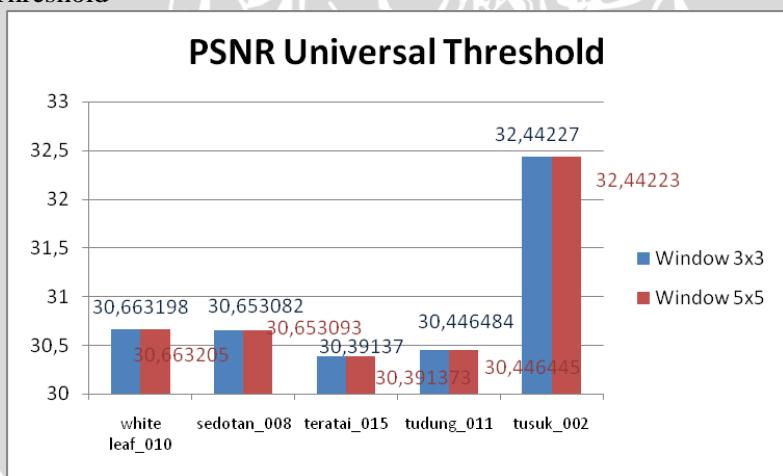
Pada hasil Tabel 4.3, terlihat bahwa nilai yang dihasilkan oleh penggunaan *Universal Threshold* menghasilkan nilai *PSNR* tertinggi daripada nilai *PSNR* penggunaan *SURE Threshold*.

Dari ketiga table diatas, informasi yang telah diperoleh dapat disajikan dalam grafik untuk mempermudah mengetahui keunggulan dari kedua metode tersebut.

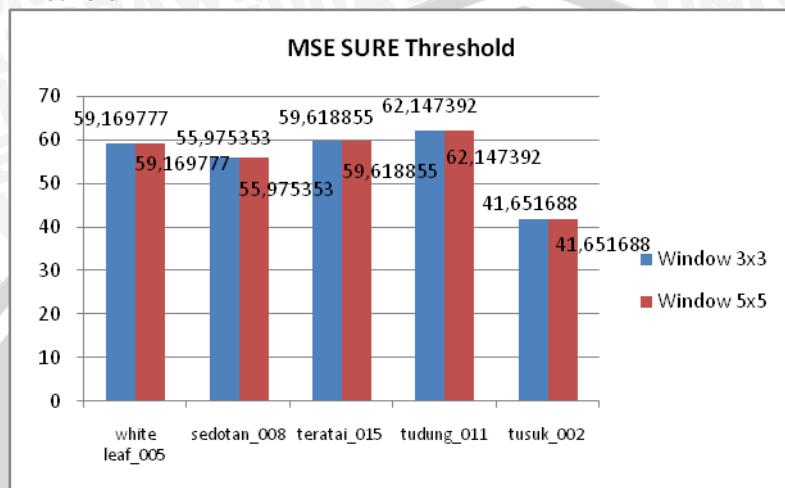
Gambar 4.2 Grafik nilai MSE citra pada penggunaan Universal Threshold



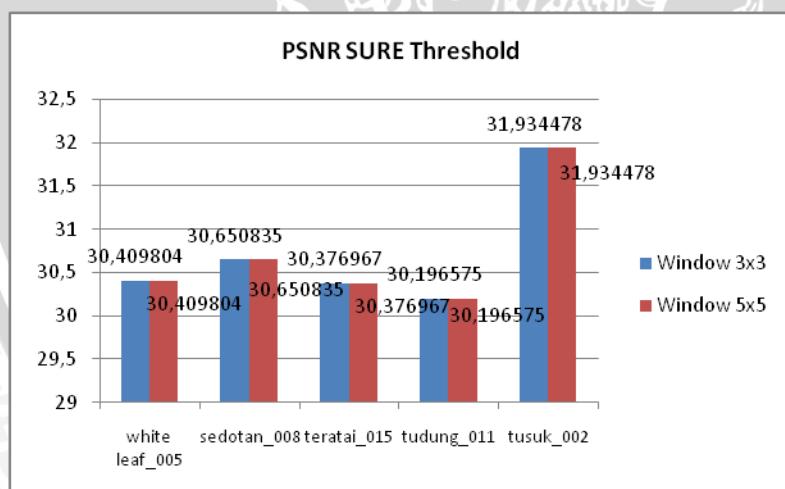
Gambar 4.3 Grafik Nilai PSNR Citra Pada Penggunaan Universal Threshold



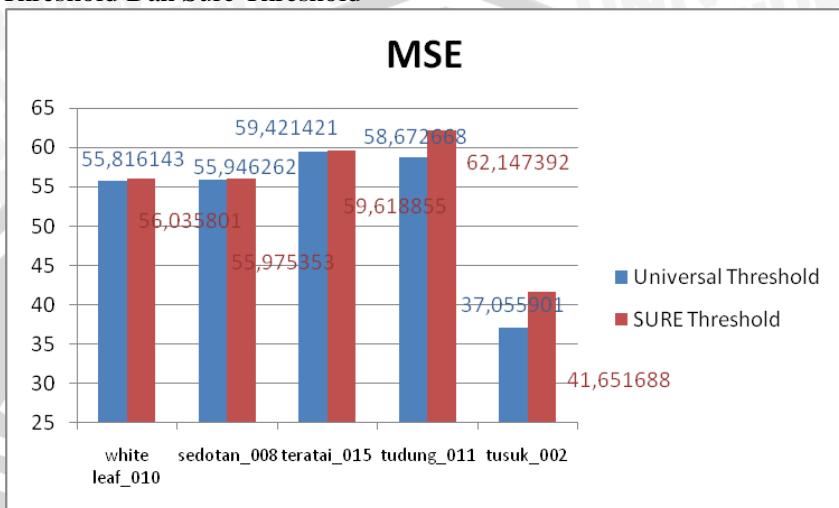
Gambar 4.4 Grafik nilai MSE citra pada penggunaan SURE Threshold



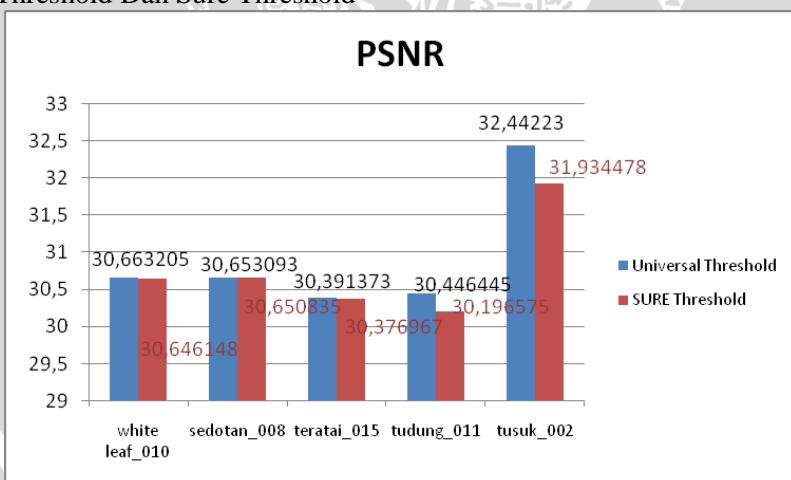
Gambar 4.5 Grafik nilai PSNR citra pada penggunaan SURE Threshold



Gambar 4.6 Grafik Perbandingan Nilai Mse Citra Universal Threshold Dan Sure Threshold



Gambar 4.7 Grafik Perbandingan Nilai Psnr Citra Universal Threshold Dan Sure Threshold



Dari hasil pada grafik 4.2-4.7, dapat terlihat bahwa metode *Universal threshold* memiliki nilai yang lebih unggul dibanding dengan metode *SURE threshold*. Sebagai contoh, untuk citra *sedotan_008.bmp*, nilai nyata PSNR antara kedua ukuran neighbouring window adalah sama, namun dua nilai terakhir memberikan arti bahwa metode *universal threshold* lebih unggul dibandingkan metode *SURE threshold*.

4.4 Analisa Hasil Pengujian

Dari hasil pengujian dapat diketahui bahwa dari seluruh citra uji yang berjumlah 30 buah, 15 citra uji dengan menggunakan metode *Universal Threshold* pada ukuran *neighbouring window* 5x5 memiliki nilai PSNR lebih tinggi daripada nilai PSNR pada ukuran window 3x3 (13 buah). Hal ini berarti, metode *universal threshold* dengan menggunakan ukuran window 5x5 memiliki keunggulan daripada metode *universal threshold* pada ukuran *neighbouring* yang lebih rendah yaitu ukuran 3x3. Sehingga dapat diambil kesimpulan bahwa pada ukuran *neighbouring window* 5x5, semakin tinggi kualitas hasil denoising pada citra. Pengaruh ukuran window yang besar pada koefisien wavelet mempengaruhi kemampuan denoising pada Neighshrink tidak berlaku pada ukuran neighbouring window 5x5. Sedangkan 2 citra yang lain yaitu citra *tudung_014.bmp* dan *bot_1234_001.bmp* memiliki nilai PSNR yang sama. Hal ini berarti pada kedua citra uji memiliki kualitas citra yang sama setelah dilakukan proses denoising.

Pada metode *SURE threshold*, nilai PSNR dari kedua ukuran window bernilai sama. Hal ini terjadi karena pada sistem dilakukan dekomposisi 1 kali terhadap citra sehingga nilai Optimal Threshold yang diinginkan belum tercapai. Nilai optimal threshold dipengaruhi ukuran subband pada dekomposisi yang dilakukan. Semakin dalam dekomposisi yang dilakukan, semakin memungkinkan perolehan optimal threshold.

Sedangkan untuk perbandingan metode dengan penggunaan *Universal Threshold* dan *SURE Threshold*, dari data hasil yang diperoleh terlihat bahwa metode dengan menggunakan *Universal Threshold* lebih unggul daripada penggunaan *SURE Threshold*. Namun hal ini tidak berlaku pada citra *daun_009.bmp* dimana pada hasil yang diperoleh menunjukkan nilai PSNR pada penggunaan

SURE Threshold memiliki nilai yang lebih tinggi dari pada penggunaan Universal Threshold. Seperti yang telah dijelaskan pada paragraf sebelumnya, hal ini tidak berarti sepenuhnya bahwa metode universal threshold lebih unggul daripada SURE threshold dikarenakan pada metode SURE threshold yang digunakan oleh sistem hanya menggunakan dekomposisi satu tingkat sehingga nilai optimal yang diinginkan belum terlihat jelas. Semakin kecil ukuran subband pada level dekomposisi, maka akan diperoleh nilai threshold yang mampu meminimalkan nilai MSE. Apabila dilakukan dekomposisi hingga level-N, kemungkinan besar nilai SURE threshold akan menjadi Optimal threshold.



UNIVERSITAS BRAWIJAYA



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian tentang denoising pada citra ber-ISO tinggi pada metode *Neighshrink* dengan menggunakan Universal Threshold dan SURE Threshold pada ukuran neighbouring window yang berbeda, dapat diambil kesimpulan sebagai berikut :

1. Metode Neighshrink pada citra ber-ISO tinggi dengan penggunaan Universal Threshold dengan ukuran neighbouring window 3×3 menghasilkan nilai PSNR yang lebih rendah dibandingkan dengan penggunaan Universal Threshold pada ukuran neighbouring window 5×5 .
2. Metode Neighshrink pada citra ber-ISO tinggi dengan penggunaan SURE Threshold pada ukuran neighbouring window 3×3 dan 5×5 menghasilkan nilai PSNR yang sama sehingga pengaruh kedua ukuran neighbouring window tidak tampak. Hal ini dikarenakan proses dekomposisi pada sistem hanya dilakukan satu kali sehingga pengaruh yang diharapkan dari proses thresholding yang dilakukan dengan penggunaan kedua ukuran window tidak terlihat.
3. Pada sistem, metode *Neighshrink* pada citra ber-ISO tinggi dengan menggunakan Universal Threshold lebih unggul dibandingkan metode *Neighshrink* dengan menggunakan SURE threshold dikarenakan proses thresholding pada penggunaan SURE threshold tidak nampak terjadi yang disebabkan oleh proses dekomposisi yang hanya dilakukan satu kali. Sedangkan pada penggunaan SURE threshold sangat dipengaruhi dengan ukuran subband pada proses dekomposisi.
4. Ukuran neigbouring window 5×5 memperoleh hasil yang lebih baik daripada ukuran 3×3 . Pengaruh ukuran neighbouring window yang terlalu besar pada pemilihan ukuran neighbouring window, dimana ukuran window yang terlalu besar akan menyimpan informasi noise, tidak berlaku pada neighbouring window dengan ukuran 5×5 .

5.2 Saran

Beberapa saran yang mungkin menjadi pertimbangan untuk penelitian selanjutnya adalah sebagai berikut :

1. Perlu dilakukan pengujian terhadap jenis citra dengan ISO yang lebih rendah yang dapat digunakan sebagai pembanding untuk mengetahui keakuratan hasil dari sistem denoising yang diteliti.
2. Perlu dilakukan proses dekomposisi secara penuh agar nilai optimal threshold pada penggunaan SURE Threshold dapat dicapai.
3. Perlu dilakukan penelitian terhadap pengaruh *neighbouring window* pada ukuran *neighbouring window* yang lebih luas agar hasil semakin akurat.



DAFTAR PUSTAKA

- Bovik ,Alan Conrad.2009. *The Essensial Guide To Image Processing.*texas:academic press.
- Cai, T.T., Silverman, B.W., 2001. *Incorporating information on neighbouring coefficients into wavelet estimation.* Sankhya, Ser. B 63, 127–148.
- Chen Gy, Bui Td, Kryzak A.2005. *image denoising using neighbouring wavelet coefficients.*
- Chen, G.Y., Bui, T.D., Krzyzak, A., 2005. *Image denoising with neighbour dependency and customized wavelet and threshold.* Pattern Recognition 38, 115–124.
- Cho Dongwook, Tien D Bui, Guangyi Chen. 2009. *Image Denoising Bades On Wavelet Shrinkage Using Neighbor And Level Dependency.* Canada: world scientific publishing company.
- Ciptaningrum, Mega Satya. 2010. *Denoising Gangguan Awan Pada Citra Satelit Landsat Menggunakan Transformasi Wavelet Diskrit Haar.* Skripsi. Malang : Universitas Brawijaya.
- Dengwen, Zhou dan Wengang Cheng. 2008. *Image Denoising With An Optimal Threshold And Neighbouring Window.* Pattern recognition 29, 1694-1697.
- Donoho,D.L, Johnstone, I.M.1994. *Ideal Spatial adaptation via Wavelet Shrinkage.* J.Amer.Statist.Assoc. 90,1200-1224.
- Gunara Andra, Tritoasomoro Iwan Imut, Raharjo Jangkung.2007.*Analisa Perbandingan Reduksi Pada Citra Antara Disrete Wavelet Transform (Dwt) Dengan Dual Tree Complex Wavelet Transform.* Seminar Nasional Sistem Dan Informatika:Bali
- King, Julie Adair. 2010. *Nikom D3000 for Dummies.* Indianapolis : Wiley Publishing,Inc.
- Linda S., Agustina.2005. *Penerapan Region of Interest (ROI) pada Metode Kompresi JPEG2000.* Departemen Teknik Infomatika, Institut Teknologi Bandung.
- Munir ,Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik.* Bandung: Penerbit Informatika.
- Murinto,Eko Aribowo, Risnadi Syazali .2007.*Analisis Perbandingan Metode Intensity Filtering Dengan Metode Frequency Filtering Sebagai Reduksi Noise Pada Citra*

- Digital. Seminar Nasional Aplikasi Teknologi Informasi 2007. Yogyakarta.
- Novamizanti, Ledy.2010.*Identifikasi Pola Iris Mata Menggunakan Dekomposisi Transformasi Wavelet*.ITB:bandung.
- Polikar, Robi.1988. *Multi Resolution Analysis : The Discrete Wavelet Transform*. Durham Computation Center, Iowa state University.
- Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: C.V. Andi Offset.
- Rahmawati, Mona.2010 .*Penerapan Fungsi Pemetaan Dengan Pendekatan Diskriminatif Pada Denoising Citra*. DigilibITS:surabaya. (<http://digilib.its.ac.id/>)
- Taswell, Carl. 1998. *The What, How, and Why of Wavelet Shrinkage Denoising*. Stanford : Computational Toolsmiths.



UNIVERSITAS BRAWIJAYA



Lampiran 1 Nilai MSE dan PSNR Hasil Pengujian Metode Neighshrink dengan menggunakan Universal Threshold pada ukuran Neighbouring window 3x3 dan 5x5

Nama citra uji (.bmp)	Neighbouring window 3x3		Neighbouring window 5x5	
	MSE	PSNR	MSE	PSNR
bot_1234_001	69,09352112	29,73643035	69,09352112	29,73643035
bot_1234_008	55,92602921	30,65466376	55,92628098	30,65464421
daun_009	91,77730179	28,50345075	91,77179718	28,50371124
daun_011	66,38889313	29,90984933	66,38952255	29,90980815
dino_1263_006	60,803936	30,29148668	60,80382156	30,29149485
dus_1259_008	62,8838768	30,14541053	62,88315582	30,14546032
libom_1233_001	41,30900574	31,97035619	41,30918884	31,97033694
libom_1233_002	39,87652588	32,12363046	39,87643433	32,12364043
libom_1233_006	78,77442169	29,16695137	78,77463913	29,16693939
libom_1233_007	57,24481201	30,55344227	57,24383926	30,55351607
libom_1233_009	51,24225998	31,03452085	51,96892548	30,97336624
libom_1233_012	51,62837219	31,00191929	51,62839508	31,00191736
mag_1255_011	66,11761093	29,92763208	66,11758804	29,92763359
mag_1255_013	45,58178329	31,54289049	45,58170319	31,54289812
mag_1255_014	54,1467247	30,79508169	54,14681625	30,79507435

pet_1251_008	46,31126404	31,47393726	46,31212234	31,47385677
pet_1251_013	73,95420456	29,44117491	73,95413589	29,44117894
race_1267_012	76,72734833	29,28130171	76,72705078	29,28131856
race_1267_013	81,08369064	29,04146853	81,08510971	29,04139252
race_1267_014	60,49845886	30,31336049	60,49866486	30,31334571
sedotan_008	55,94639969	30,65308217	55,94626236	30,65309283
teratai_013	70,07733536	29,67502781	70,07723236	29,67503419
teratai_015	59,42146683	30,39136993	59,42142105	30,39137327
tudung_011	58,67214203	30,44648417	58,67266846	30,4464452
tudung_012	67,42760468	29,84242629	67,42749023	29,84243366
tudung_014	73,33216095	29,47785878	73,33216095	29,47785878
tusuk_002	37,05555725	32,44227012	37,05590057	32,44222988
tusuk_015	46,66497803	31,44089295	46,6648407	31,44090573
tv_1247_003	31,36023331	33,16701076	31,36325455	33,16659238
whiteleaf_010	55,81623459	30,66319826	55,81614304	30,66320538

Lampiran 2 Nilai MSE dan PSNR Hasil Pengujian Metode Neighshrink dengan menggunakan SURE Threshold pada ukuran Neighbouring window 3x3 dan 5x5

Nama citra uji (.bmp)	Neighbouring window 3x3		Neighbouring window 5x5	
	MSE	PSNR	MSE	PSNR
bot_1234_001	69,09352112	29,73643035	69,09352112	29,73643035
bot_1234_008	57,65492249	30,52243968	57,65492249	30,52243968
daun_009	86,23868179	28,77378251	86,23868179	28,77378251
daun_011	71,93767548	29,56123961	71,93767548	29,56123961
dino_1263_006	62,16970825	30,19501532	62,16970825	30,19501532
dus_1259_008	69,24097824	29,72717166	69,24097824	29,72717166
libom_1233_001	41,98564911	31,89979489	41,98564911	31,89979489
libom_1233_002	40,41963959	32,06487924	40,41963959	32,06487924
libom_1233_006	82,91962051	28,94423055	82,91962051	28,94423055
libom_1233_007	59,7118721	30,37019673	59,7118721	30,37019673
libom_1233_009	56,32580566	30,62372948	56,32580566	30,62372948
libom_1233_012	51,85890198	30,98257044	51,85890198	30,98257044
mag_1255_011	67,04980087	29,86682868	67,04980087	29,86682868

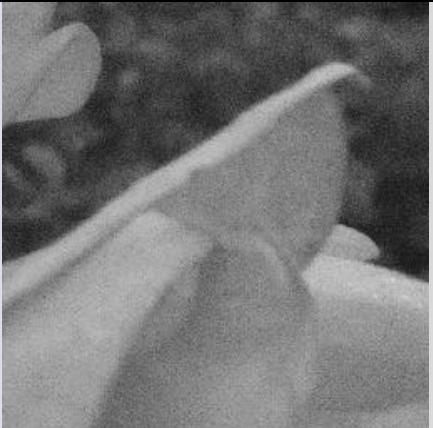
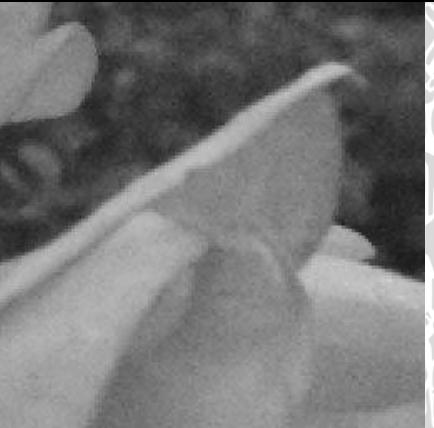
mag_1255_013	46,53326797	31,45316807	46,53326797	31,45316807
mag_1255_014	59,19676208	30,40782408	59,19676208	30,40782408
pet_1251_008	55,33601761	30,7007246	55,33601761	30,7007246
pet_1251_013	75,38727951	29,3578229	75,38727951	29,3578229
race_1267_012	82,99764633	28,94014584	82,99764633	28,94014584
race_1267_013	90,62688446	28,55823311	90,62688446	28,55823311
race_1267_014	68,50090027	29,77384082	68,50090027	29,77384082
sedotan_008	55,97535324	30,65083518	55,97535324	30,65083518
teratai_013	72,63662338	29,51924714	72,63662338	29,51924714
teratai_015	59,61885452	30,37696733	59,61885452	30,37696733
tudung_011	62,14739227	30,19657451	62,14739227	30,19657451
tudung_012	68,45967865	29,77645505	68,45967865	29,77645505
tudung_014	73,85538483	29,44698195	73,85538483	29,44698195
tusuk_002	41,65168762	31,93447758	41,65168762	31,93447758
tusuk_015	47,14215088	31,39670967	47,14215088	31,39670967
tv_1247_003	47,05458069	31,40478453	47,05458069	31,40478453
whiteleaf_010	56,03580093	30,64614777	56,03580093	30,64614777

Lampiran 3 Perbandingan Nilai MSE dan PSNR Hasil Pengujian Metode Neighshrink dengan menggunakan Universal Threshold dan SURE Threshold pada ukuran Neighbouring window 5x5

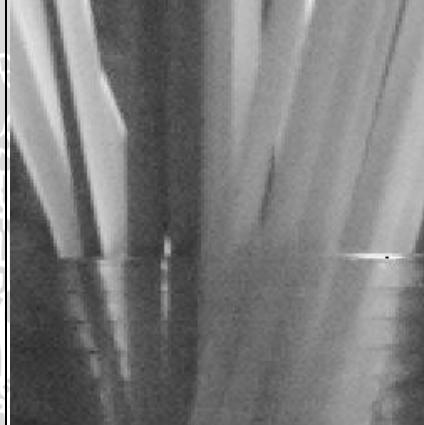
Nama citra uji (.bmp)	UNIVERSAL THRESHOLD		SURE THRESHOLD	
	MSE	PSNR	MSE	PSNR
bot_1234_001	69,09352112	29,73643035	69,09352112	29,73643035
bot_1234_008	55,92628098	30,65464421	57,65492249	30,52243968
daun_009	91,77179718	28,50371124	86,23868179	28,77378251
daun_011	66,38952255	29,90980815	71,93767548	29,56123961
dino_1263_006	60,80382156	30,29149485	62,16970825	30,19501532
dus_1259_008	62,88315582	30,14546032	69,24097824	29,72717166
libom_1233_001	41,30918884	31,97033694	41,98564911	31,89979489
libom_1233_002	39,87643433	32,12364043	40,41963959	32,06487924
libom_1233_006	78,77463913	29,16693939	82,91962051	28,94423055
libom_1233_007	57,24383926	30,55351607	59,7118721	30,37019673
libom_1233_009	51,96892548	30,97336624	56,32580566	30,62372948
libom_1233_012	51,62839508	31,00191736	51,85890198	30,98257044
mag_1255_011	66,11758804	29,92763359	67,04980087	29,86682868

mag_1255_013	45,58170319	31,54289812	46,53326797	31,45316807
mag_1255_014	54,14681625	30,79507435	59,19676208	30,40782408
pet_1251_008	46,31212234	31,47385677	55,33601761	30,7007246
pet_1251_013	73,95413589	29,44117894	75,38727951	29,3578229
race_1267_012	76,72705078	29,28131856	82,99764633	28,94014584
race_1267_013	81,08510971	29,04139252	90,62688446	28,55823311
race_1267_014	60,49866486	30,31334571	68,50090027	29,77384082
sedotan_008	55,94626236	30,65309283	55,97535324	30,65083518
teratai_013	70,07723236	29,67503419	72,63662338	29,51924714
teratai_015	59,42142105	30,39137327	59,61885452	30,37696733
tudung_011	58,67266846	30,4464452	62,14739227	30,19657451
tudung_012	67,42749023	29,84243366	68,45967865	29,77645505
tudung_014	73,33216095	29,47785878	73,85538483	29,44698195
tusuk_002	37,05590057	32,44222988	41,65168762	31,93447758
tusuk_015	46,6648407	31,44090573	47,14215088	31,39670967
tv_1247_003	31,36325455	33,16659238	47,05458069	31,40478453
whiteleaf_010	55,81614304	30,66320538	56,03580093	30,64614777

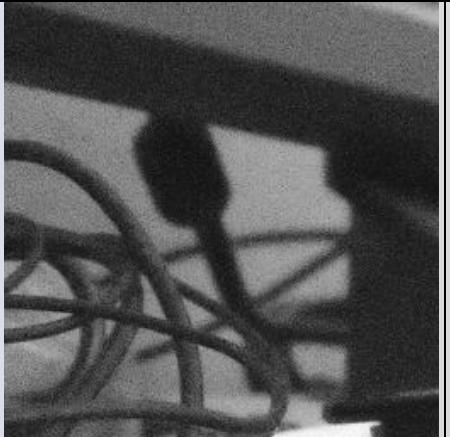
Lampiran 4 Contoh Hasil Denoising Citra Uji secara Visual pada metode Neighshrink dengan menggunakan Universal Threshold

Citra Asli (<i>whiteleaf_010.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

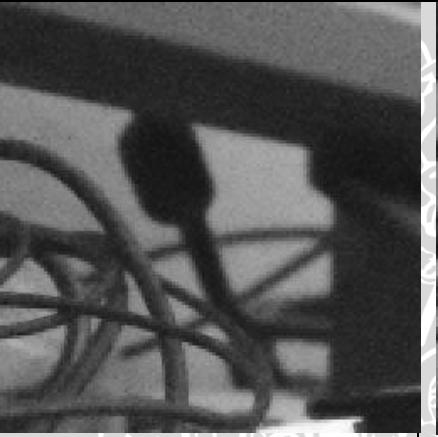
UNIVERSITAS BRAWA

Citra Asli (<i>sedotan_008.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

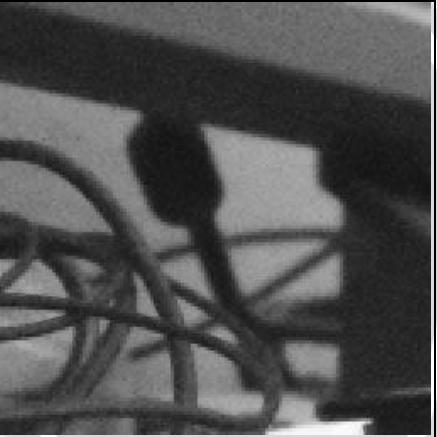
Citra Asli (*dino_006.bmp*)



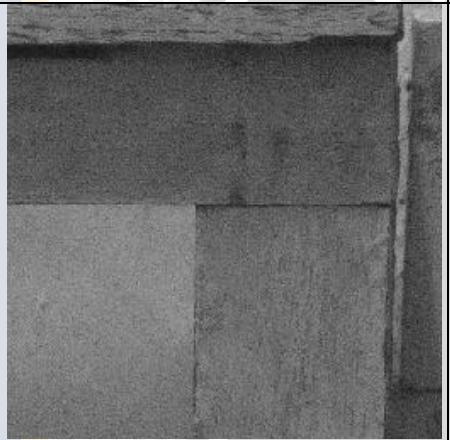
Denoising dengan ukuran
neighbouring window 3x3



Denoising dengan ukuran
neighbouring window 5x5



Citra Asli (*pet_013.bmp*)



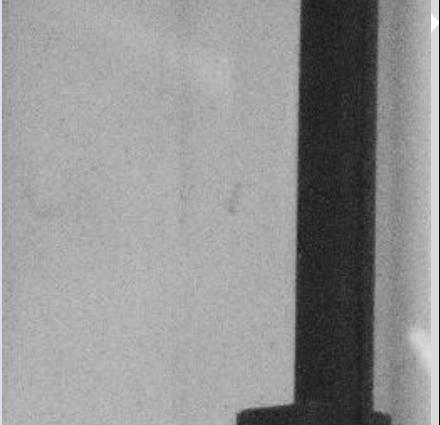
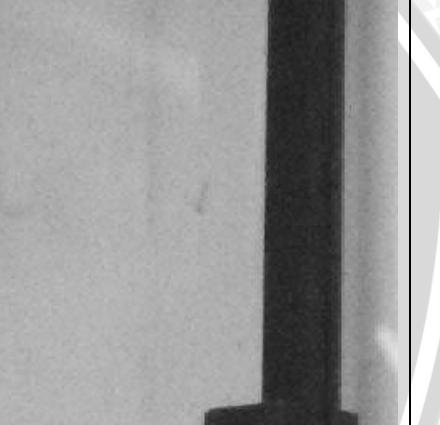
Denoising dengan ukuran neighbouring window 3x3

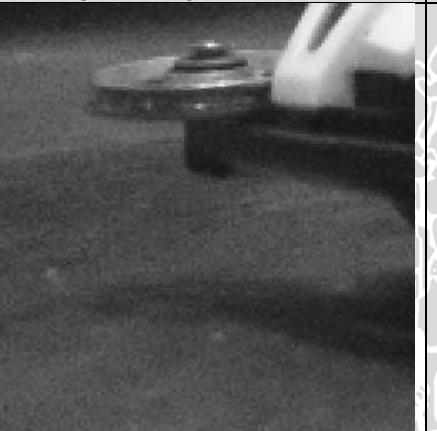


Denoising dengan ukuran neighbouring window 5x5

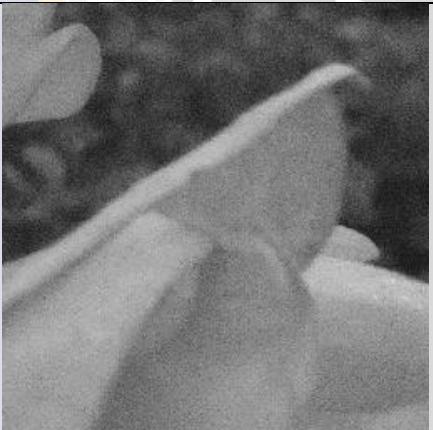
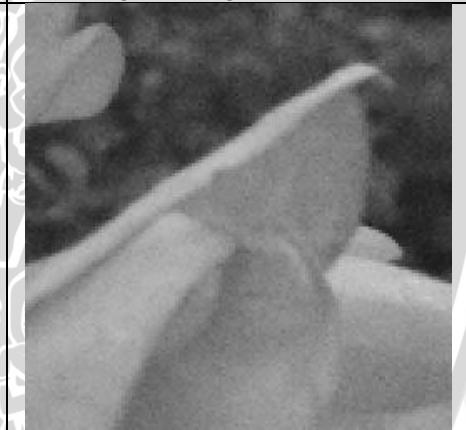


PERBEDAAN DENGAN UKURAN NEIGHBOURING WINDOW

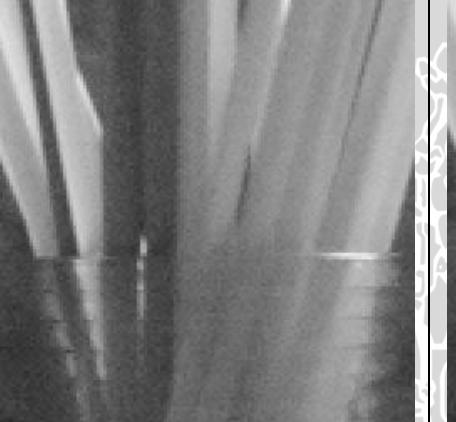
Citra Asli (<i>tusuk_002.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

Citra Asli (<i>mag_011.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

Lampiran 5 Contoh Hasil Denoising Citra Uji secara Visual pada metode Neighshrink dengan menggunakan SURE Threshold

Citra Asli (<i>whiteleaf_010.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

UNIVERSITAS BRAWA

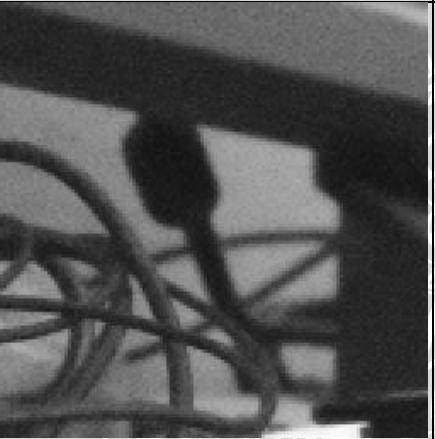
Citra Asli (<i>sedotan_008.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		



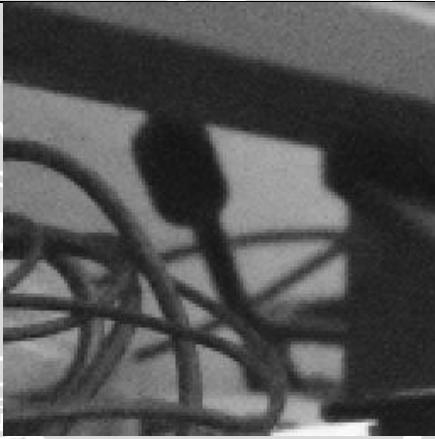
Citra Asli (*dino_006.bmp*)

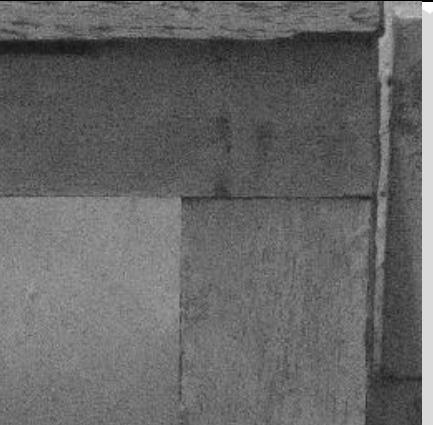


Denoising dengan ukuran
neighbouring window 3x3

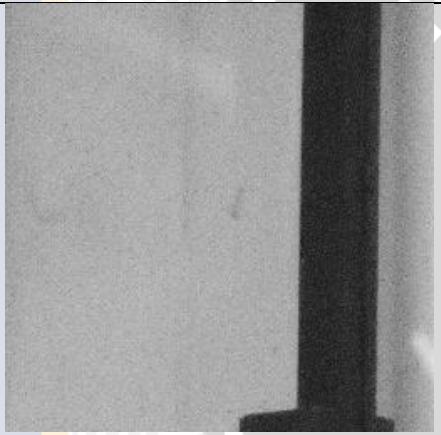


Denoising dengan ukuran
neighbouring window 5x5

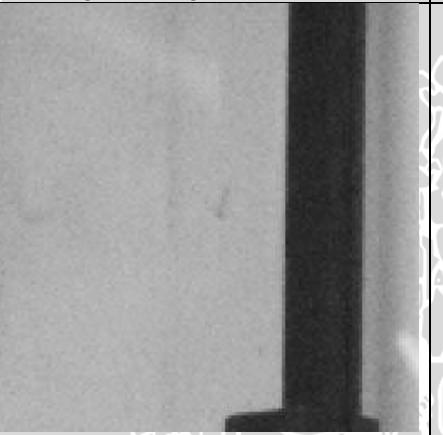


Citra Asli (<i>pet_013.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

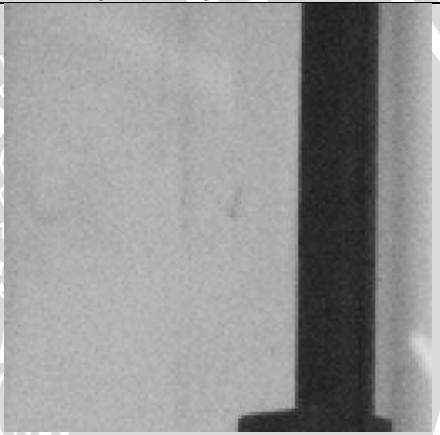
Citra Asli (*tusuk_002.bmp*)



Denoising dengan ukuran
neighbouring window 3x3



Denoising dengan ukuran
neighbouring window 5x5



Citra Asli (<i>mag_011.bmp</i>)	Denoising dengan ukuran neighbouring window 3x3	Denoising dengan ukuran neighbouring window 5x5
		

UNIVERSITAS BRAWIJAYA

