



1. Pendahuluan

1.1 Latar Belakang

Berdasarkan jenis representasinya citra dibagi menjadi dua yaitu citra analog dan citra digital. Perubahan citra analog ke citra digital terkadang menimbulkan *noise*. *Noise* adalah bagian pada citra yang mengalami kerusakan dan berbeda dari citra aslinya sehingga mengurangi kualitas dari citra tersebut.

Image denoising adalah suatu teknik yang digunakan untuk menghilangkan *noise* dari suatu citra digital ternoise. Salah satu jenis dari image denoising adalah menggunakan domain frekuensi. Transformasi wavelet efektif digunakan untuk mengubah citra digital ke dalam domain frekuensi. Pada jenis ini, citra digital terlebih dahulu diubah ke dalam domain frekuensi (dalam bentuk *wavelet coefficient*) sebelum proses denoising. Threshold digunakan untuk memfilter *noise* dari citra yang sudah diubah ke dalam domain frekuensi tersebut.

Penentuan nilai threshold yang optimum merupakan salah satu permasalahan pada image denoising dalam domain frekuensi. Penggunaan metoda seperti Normal Shrinkage (Normalshrink) pada soft thresholding dan hard thresholding dapat memecahkan permasalahan ini yaitu dengan mencari nilai threshold yang optimum berdasarkan subband-subband [7]. Namun metoda ini mengasumsikan bahwa *wavelet coefficient* bersifat *independent*.

Metoda Bivariate Shrinkage dalam melakukan proses denoising mengasumsikan sifat *dependent* antara *wavelet coefficient* dengan parent dari *wavelet coefficient* tersebut yaitu pada dekomposisi wavelet yang lebih tinggi. Pada [14], sifat *dependent* ini bisa meningkatkan performansi denoising.

Performansi denoising metoda Bivariate Shrinkage dimungkinkan untuk ditingkatkan yaitu dengan pemodelan perhitungan signal secara local neighborhood pada perhitungan *marginal variance* citra ternoise[10]. Varian metoda Bivariate Shrinkage yang menggunakan pemodelan perhitungan secara local neighborhood untuk menghitung variansi dari citra ternoise disebut sebagai metoda Bivariate Shrinkage Dengan Local Variance Estimation.

1.2 Perumusan Masalah

Perumusan masalah pada Tugas Akhir ini adalah :

1. Bagaimana performansi image denoising berupa nilai Peak Signal to Noise Ratio (PSNR) menggunakan metoda Bivariate Shrinkage Dengan Local Variance Estimation.
2. Bagaimana pengaruh perubahan *window size* terhadap nilai threshold pada bivariate shrinkage function yang mempengaruhi hasil denoising dan waktu komputasi pada citra grayscale.

Untuk menghindari meluasnya pembahasan pada Tugas Akhir ini, maka permasalahan dibatasi pada :

1. Jenis transformasi wavelet yang digunakan adalah separable discrete wavelet transform dengan farras[1][4] sebagai wavelet filter. Tidak dilakukan pembahasan detail dari transformasi dan filter wavelet jenis tersebut.
2. Ukuran citra grayscale bujur sangkar sama sisi. Ukuran piksel 256x256.
3. Ukuran *window size* $N \times N$ dengan N bilangan ganjil lebih besar dari 1. N yang digunakan antara 3 sampai 51.

4. Jenis *noise* yang digunakan adalah *additive gaussian noise*, *impulsive noise*, *laplacian noise*.

1.3 Tujuan Pembahasan

Tujuan pembahasan Tugas Akhir ini adalah untuk :

1. Menganalisis hasil uji pengaruh perubahan window size terhadap hasil denoising pada citra grayscale sehingga diketahui pengaruh perubahannya terhadap nilai PSNR dan waktu komputasi.
2. Mengetahui bagaimana performansi image denoising menggunakan metoda Bivariate Shrinkage Dengan Local Variance Estimation, dengan parameter tingkat hasil denoising berupa nilai PSNR.

1.4 Metodologi Penyelesaian Masalah

Metodologi penyelesaian masalah pada Tugas Akhir ini adalah :

1. Studi Literatur.
Studi literatur yang diperlukan untuk memecahkan permasalahan pada Tugas Akhir ini dari paper, jurnal ataupun dokumen yang membahas konsep image denoising, wavelet thresholding, Bivariate Shrinkage Dengan Local Variance Estimation dan ukuran performansi PSNR.
2. Analisis Kebutuhan dan perancangan sistem.
Analisis masalah dan kebutuhan perangkat lunak yaitu analisis metoda Bivariate Shrinkage Dengan Local Variance Estimation. Berisi Diagram Alir Data (DAD) dari Analisis kebutuhan.
3. Implementasi.
Tahap pembuatan perangkat lunak image denoising domain frekuensi basis wavelet menggunakan metoda Bivariate Shrinkage Dengan Local Variance Estimation. Implementasi dilakukan menggunakan Matlab 7.
4. Pengujian.
Citra uji dibubuhi *additive gaussian/additive laplacian/impulsive noise* untuk mendapatkan citra ternoise. Dilakukan proses denoising pada citra ternoise tersebut menggunakan metoda Bivariate Shrinkage Dengan Local Variance Estimation sehingga menghasilkan citra hasil denoise. Hitung PSNR dari citra ternoise dan citra hasil denoise tersebut. Pengujian dilakukan dengan mengubah ukuran window size, jumlah dekomposisi dan konstanta *noise* yang berbeda. Pengujian juga memperhatikan perubahan waktu pada setiap perubahan ukuran window size. Pengujian dilakukan terhadap citra grayscale dengan karakteristik histogram yang berbeda. Untuk keperluan perbandingan dilakukan juga denoising menggunakan metoda Bivariate Shrinkage dan metoda Normalshrink.
5. Analisis
Melakukan analisis dari hasil pengujian sehingga didapatkan kesimpulan akhir dari rumusan masalah pada Tugas Akhir ini.
6. Kesimpulan dan Saran.
Mendapatkan kesimpulan akhir dan saran untuk pengembangan selanjutnya.

Lampiran A : Spesifikasi Proses

Nama	=	Proses 1.1 Baca Citra Clean
Deskripsi	=	Proses ini membaca file citra grayscale yang belum terkena noise (Citra clean). Setelah dibaca file ini dirubah ke dalam bentuk elemen matrik. Output dari proses ini adalah matrik dari citra asli (Data citra clean).
Input	=	Citra clean
Output	=	Data citra clean
Logika Proses	=	{I.S.: Citra clean dalam bentuk file bitmap} {F.S.: Citra clean sudah dipetakan ke dalam matrik citra} Open(citra clean) Data citra clean = read(citra clean)

Nama	=	Proses 1.2 Generate Noise
Deskripsi	=	Proses ini melakukan penggenerate-an noise terhadap data citra clean berdasarkan jenis <i>noise</i> dan ukuran <i>noise</i> yang diinputkan user. Jenis noise yang bisa digenerate adalah additive gaussian, additive laplacian, impulsive noise. Untuk <i>additive gaussian noise</i> , konstanta <i>noise</i> dinyatakan dengan standar deviasi, <i>additive laplacian noise</i> dengan konstanta laplace, sedangkan <i>impulsive noise</i> dinyatakan dengan probabilitas noise.
Input	=	Jenis noise, konstanta noise, data citra clean
Output	=	Data citra ter-noise.
Logika Proses	=	{I.S : Data Citra belum terkena noise} {F.S.: Data Citra sudah terkena noise} if jenis noise = gaussian then x=random distribusi normal sebesar matrik citra. Matrik Noise=Matrik citra asli+x*ukuran noise. Else If jenis noise=impulsive then Matrik noise = matrik citra asli x=random matrik sebesar matrik citra. d= index koefisien matrik x < ukuran noise/2 matrik noise(d)=0 d= index koefisien matrik x >= ukuran noise/2 dan index koefisien matrik x < ukuran noise matrik noise(d)=255 Else x=random matrik sebesar citra matrik noise=matrik citra d=index koefisien matrik x < 0.5 matrik noise(d)=matrikcitra asli(d)+ukuran noise*(2*x(d)) d=index koefisien matrik x < 1 matrik noise(d)=matrik citra asli(d)-ukuran noise*(2*x(d)) end if matrik citra ter-noise=matrik noise

Nama	=	Proses 1.3 Baca Citra Ternoise
Deskripsi	=	Proses ini membaca file citra grayscale yang sudah terkena noise (Citra ternoise). Setelah dibaca file ini dirubah ke dalam bentuk elemen matrik. Output dari proses ini adalah matrik dari citra ternoise (Data citra ternoise).
Input	=	Citra ternoise
Output	=	Data citra ternoise
Logika Proses	=	{I.S.: Citra ternoise dalam bentuk file bitmap} {F.S.: Citra ternoise sudah dipetakan ke dalam matrik citra}

		Open(citra ternoise) Data citra ternoise = read(citra ternoise)
Nama	=	Proses 2.1 Forward Transformasi Wavelet
Deskripsi	=	Proses ini mengubah data citra ternoise ke dalam bentuk koefisien wavelet.
Input	=	Data citra ternoise, Jumlah dekomposisi
Output	=	Koefisien wavelet ternoise
Logika Proses	=	{I.S.: Data citra ternoise dalam bentuk elemen matrik} {F.S.: Data citra ternoise sudah dalam bentuk koefisien wavelet} Koefisien wavelet ternoise = FDWT(Jumlah dekomposisi, Data citra ternoise)
Nama	=	Proses 2.3 Inverse Transformasi Wavelet
Deskripsi	=	Proses ini mengubah Koefisien wavelet hasil denoising koefisien wavelet ternoise ke dalam bentuk data citra hasil (elemen matrik).
Input	=	Koefisien wavelet hasil, Jumlah dekomposisi
Output	=	Data citra hasil
Logika Proses	=	{I.S.: Koefisien wavelet hasil denoising} {F.S.: Koefisien wavelet hasil denoising yang sudah dirubah representasinya ke dalam bentuk elemen matrik kembali} Data citra hasil = IDWT(Jumlah dekomposisi, Koefisien wavelet hasil)
Nama	=	Proses 3.1 Hitung PSNR
Deskripsi		Proses ini menghitung nilai Peak Signal to Noise Ratio (PSNR) dari citra ternoise dan citra hasil denoising
Input	=	Data citra clean, Data citra ternoise, Data citra hasil
Output	=	PSNR ternoise, PSNR hasil
Logika Proses	=	{I.S. matrik citra clean, matrik citra ter-noise, matrik citra hasil denoising} {F.S. Nilai PSNR citra noise dan citra denoising} If matrik citra clean = ada then Selisih citra ternoise = Data citra clean – Data citra ternoise PSNR ternoise = $20 \cdot \log_{10}(256 / \text{std}(\text{Selisih citra ternoise}(:)))$ Selisih citra hasil = Data citra clean – Data citra hasil PSNR hasil = $20 \cdot \log_{10}(256 / \text{std}(\text{Selisih citra hasil}(:)))$ Else PSNR ternoise = tidak ada PSNR hasil = tidak ada End if
Nama	=	Proses 3.2 Representasi Matrik
Deskripsi	=	Pada proses ini dilakukan penulisan data citra kedalam file dalam bentuk file bitmap.
Input	=	Data citra clean, Data citra ternoise, Data citra hasil
Output	=	file bitmap citra clean, citra ternoise, dan citra hasil
Logika Proses	=	{I.S. Data citra clean, Data citra ternoise, Data citra hasil} {F.S. File citra clean, citra ternoise, citra hasil} File citra clean=write(Data citra clean) File citra ternoise=write(Data citra ternoise) File citra hasil=write(Data citra hasil)
Nama	=	Proses 2.2.2 BiShrink
Deskripsi	=	Pada proses ini dilakukan image denoising menggunakan metoda Bivariate Shrinkage.