



TESIS-SM 142501

PENERAPAN TRANSFORMASI WAVELET DAUBECHIES UNTUK REDUKSI NOISE HUJAN PADA VIDEO

Siti Khotijah
06111450012004

Dosen Pembimbing
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.

**PROGRAM MAGISTER
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**



THESIS-SM 142501

APPLICATION OF DAUBECHIES WAVELET TRANSFORMATION FOR NOISE RAIN REDUCTION ON THE VIDEO

Siti Khotijah
06111450012004

Supervisor
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T

**MASTER'S DEGREE
MATHEMATICS DEPARTMENT
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2018**

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sains (M.Si.)
di
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

oleh:

SITI KHOTIJAH

NRP. 06111450012004

Tanggal Ujian : 15 Januari 2018

Periode Wisuda

: Maret 2018

Disetujui oleh:

Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.
NIP. 19690405 199403 2 003

(Pembimbing)

Dr. Mahmud Yunus, M.Si.
NIP. 19620407 198703 1 005

(Penguji)

Dr. Budi Setiyono, S.Si., M.T.
NIP. 19720207 199702 1 001

(Penguji)

Dr. Imam Mukhlash, S.Si., M.T.
NIP. 19700831 199403 1 003

(Penguji)



PENERAPAN TRANSFORMASI WAVELET DAUBECHIES UNTUK REDUKSI NOISE HUJAN PADA VIDEO

Nama mahasiswa : Siti Khotijah
NRP : 06111450012004
Pembimbing : Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.

ABSTRAK

Saat ini penggunaan video digital di bidang ilmu komputer semakin luas, seperti proses *tracking object*, penghitungan jumlah kendaraan, klasifikasi jenis kendaraan, estimasi kecepatan kendaraan dan sebagainya. Proses pengambilan video digital tersebut seringkali dipengaruhi oleh cuaca buruk, seperti pengambilan pada saat hujan. Hujan dalam video digital dianggap *noise* karena mampu menghalangi objek yang sedang di amati. Oleh karena itu, dibutuhkan suatu proses reduksi *noise* hujan yang terdapat pada video tersebut.

Pada penelitian ini reduksi *noise* hujan pada video digital dilakukan dengan menerapkan Transformasi wavelet *Daubechies* yang melewati beberapa proses yaitu: Dekomposisi wavelet, proses fusi, proses *thresholding* dan proses rekonstruksi. Nilai *threshold* yang digunakan dalam proses *thresholding* yaitu *VishuShrink*, *BayesShrink*, dan *NormalShrink* dan diuji coba menggunakan data hujan deras, sedang dan rendah. Hasil Implementasi dan uji coba diperoleh bahwa reduksi *noise* untuk data hujan deras, perbandingan nilai PSNR tertinggi yaitu 27,3138 dB dengan waktu komputasi 29,4475 detik menggunakan wavelet *Daubechies* db2 level 3 *VishuShrink*. Untuk data hujan sedang, perbandingan nilai PSNR tertinggi yaitu 34,6985 dB dengan waktu komputasi 38,3172 detik menggunakan wavelet *Daubechies* db4 level 3 *VishuShrink*. Sedangkan untuk data hujan rendah, perbandingan nilai PSNR tertinggi yaitu 38,7916 dB dengan waktu komputasi 34,5306 detik.

Kata Kunci : Transformasi wavelet *daubechies*, *fusi*, *VishuShrink*, *BayesShrink* *NormalShrink*.

APPLICATION OF DAUBECHIES WAVELET TRANSFORMATION FOR NOISE (RAIN) REDUCTION ON VIDEO

By : Siti Khotijah
Student Identity Number : 06111450012004
Supervisor : Dr. Dwi Ratna Sulistyaningrum, S. Si., M.T.

ABSTRACT

Currently the use of digital video in the field of computer science increasingly widespread, such as the process of tracking objects, the calculation of the number of vehicles, classification of vehicle types, vehicle speed estimates and so forth. The process of taking digital video is often influenced by bad weather, such as taking in the rain. Rain in digital video is considered *noise* because it is able to block objects being observed. Therefore, a rainfall *noise* reduction process is required in the video.

In this study, the reduction of rain *noise* in digital video is done by applying *Daubechies* wavelet transformation through several processes, namely: wavelet decomposition, fusion process, thresholding process and reconstruction process. The threshold values used in the thresholding process are *VishuShrink*, *BayesShrink*, and *NormalShrink* and are tested using heavy, medium and low rainfall data. The results of the implementation and test show that *noise* reduction for rainfall data, the highest PSNR value is 27.3138 dB with computation time of 29.4475 seconds using *Daubechies* db2 level 3 *VishuShrink* wavelet. For moderate rainfall data, the highest PSNR score was 34.6985 dB with computation time 38.3172 sec using *Daubechies* db4 level 3 *VishuShrink* wavelet. As for the low rainfall data, the highest PSNR value is 38.7916 dB with computation time of 34.5306 seconds.

Keyword : *Daubechies Wavelet Transformation, fusion, VishuShrink, BayesShrink, NormalShrink*

KATA PENGANTAR

Dengan rahmat Allah SWT, syukur Alhamdulillah penulis dapat menyelesaikan tesis yang berjudul:

Penerapan Transformasi Wavelet Daubechies Untuk Reduksi Noise Hujan Pada Video

Tesis ini disusun sebagai salah satu syarat kelulusan Program Studi Strata2 (S-2) Program Magister Departemen Matematika Fakultas Matematika, Komputasi dan Sains Data (FMKSD) Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penulisan tesis ini, tidak akan terselesaikan dengan baik tanpa adanya bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Kedua orang tua dengan segala kasih sayangnya selalu memberikan motivasi, saran, dukungan, dan doa kepada penulis sehingga dapat menyelesaikan tesis ini.
2. Bapak Prof. Ir. Joni Hermana, M.Sc.ES., Ph.D., selaku Rektor Institut Teknologi Sepuluh Nopember yang telah memberikan kesempatan dan fasilitas kepada penulis untuk menyelesaikan tesis ini.
3. Bapak Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D., selaku Direktur Program Pascasarjana Institut Teknologi Sepuluh Nopember.
4. Bapak Prof. Dr. Basuki Widodo, M.Sc., selaku Dekan Fakultas Matematika, Komputasi dan Sains Data (FMKSD).
5. Bapak Dr. Imam Mukhlash, S.Si., M.T., selaku Ketua Jurusan Matematika Institut Teknologi Sepuluh Nopember.
6. Bapak Dr. Mahmud Yunus, M.Si., selaku koordinator Program Studi Pascasarjana Matematika ITS.
7. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si, M.T., selaku dosen pembimbing tesis yang telah banyak mengarahkan, membimbing, dan memberikan motivasi sehingga tesis ini dapat terselesaikan.

8. Bapak Dr. Budi Setiyono, S.Si, M.T., Bapak Dr. Mahmud Yunus, M.Si., dan Bapak Dr. Imam Mukhlash, S.Si., M.T., selaku dosen penguji tesis yang telah banyak memberikan saran dan masukan sehingga tesis ini dapat diselesaikan dengan baik.
9. Seluruh dosen, staf, dan karyawan Jurusan Matematika ITS yang telah memberikan bantuan selama penulis menempuh perkuliahan ini.
10. Suamiku Andi Sholihin yang selalu mendampingi dan membantu ketika ada kesulitan.
11. Teman-teman S2 Matematika ITS angkatan 2014.
12. Seluruh pihak yang telah memberikan saran, dukungan, dan motivasi kepada penulis dalam penyelesaian tesis ini.

Penulis menyadari bahwa dalam penulisan ini masih banyak kekurangan, kesalahan, dan masih jauh dari kata sempurna sehingga segala kritik dan saran yang sifatnya membangun sangat diperlukan. Penulis berharap tesis ini dapat memberikan manfaat bagi penulis pada khususnya dan pembaca pada umumnya.

Surabaya, Januari 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1 Penelitian Terdahulu	5
2.2 Pengertian Citra Digital.....	6
2.3 Video	6
2.4 <i>Noise Hujan</i>	7
2.5 Transformasi Wavelet	8
2.5.1 Fungsi skala (<i>Scaling function</i>)	9
2.5.2. Wavelet <i>Daubechies</i>	10
2.5.3 Proses Dekomposisi	11
2.5 Wavelet <i>fusion</i>	13
2.6. Thresholding.....	16
2.7 Mean Filter	18
2.8 Rekonstruksi Wavelet	18
2.7 Peak Signal to Noise Ratio(PSNR)	19
BAB 3 METODE PENELITIAN.....	21

3.1	Peralatan Penelitian	21
3.2	Tahapan Penelitian.....	21
3.3	Block Diagram Reduksi <i>NoiseHujan</i> pada Video	22
BAB 4	PENERAPAN TRANSFORMASI WAVELET <i>DAUBECHIES</i> UNTUK REDUKSI <i>NOISE HUJAN</i> PADA VIDEO.....	25
4.1	Perancangan Proses	25
4.1.1	Dekomposisi Wavelet.....	27
4.1.2	Aturan Wavelet Fusion	32
4.1.3	Proses <i>thresholding</i>	34
4.1.5.	Mengukur Kualitas Video dengan Membandingkan Nilai PSNR.	36
4.2	Implementasi Proses Reduksi <i>noise</i> Hujan Pada Video	37
4.2.2	Konversi video ke <i>grayscale</i>	37
4.2.3	Ekstraksi video menjadi urutan <i>frame</i>	38
4.2.4	Dekomposisi wavelet <i>Daubechies</i> dari tiap-tiap <i>frame</i>	38
4.2.5	Wavelet <i>fusion</i>	39
BAB 5	UJI COBA DAN ANALISIS HASIL.....	41
5.1	Data Uji Coba	41
5.2	Uji Coba reduksi <i>noise</i> hujan pada video dan Analisis Hasil.....	44
5.2.1	Uji coba reduksi <i>noise</i> hujan deras <i>background</i> diam objek bergerak... ..	44
5.2.2	Uji coba reduksi <i>noise</i> hujan sedang <i>background</i> diam objek bergerak	46
5.2.3	Uji coba reduksi <i>noise</i> hujan rendah <i>background</i> diam objek bergerak	48
5.2.4	Uji coba reduksi <i>noise</i> hujan deras <i>background</i> diam tanpa objek ..	50
5.2.5	Uji coba reduksi <i>noise</i> hujan sedang <i>background</i> diam tanpa objek .	52
5.2.6	Uji coba reduksi <i>noise</i> hujan rendah <i>background</i> diam tanpa objek .	54
5.2.7	Uji coba reduksi <i>noise</i> hujan deras <i>background</i> diam objek diam	56
5.2.8	Uji coba reduksi <i>noise</i> hujan sedang <i>background</i> diam objek diam .	58
5.2.9	Uji coba reduksi <i>noise</i> hujan rendah <i>background</i> diam objek diam ..	60
5.2.10	Uji coba reduksi <i>noise</i> hujan deras <i>background</i> bergerak objek bergerak ..	62

5.2.11 Uji coba reduksi <i>noise</i> hujan sedang <i>background</i> bergerak objek bergerak	65
5.2.12 Uji coba reduksi <i>noise</i> hujan rendah <i>background</i> bergerak objek bergerak	67
BAB 6 KESIMPULAN DAN SARAN.....	71
6.1 Kesimpulan.....	71
6. 2 Saran.....	72
DAFTAR PUSTAKA	73
BIODATA PENULIS	75

DAFTAR GAMBAR

Gambar 4.1 : Diagram alir perancangan proses	26
Gambar 4.2 : Representasi pixel A yang diambil dari citra Asal B	27
Gambar 4.3 : Matriks filter wavelet <i>Daubechies</i> (db4).....	28
Gambar 4.4 : Matriks <i>frame</i> pada titik tertentu.....	28
Gambar 4.5 : Perkalian konvolusi antara matriks filter <i>Daubechies</i> (db4) dengan bagian kolom matriks A.....	28
Gambar 4.6: Hasil perkalian per kolom yang terdiri dari bagian komponen aproksimasi dan bagian komponen detail	29
Gambar 4.7 : Perkalian konvolusi baris antara matriks filter <i>Daubechies</i> (db4) dengan hasil perkalian kolom.....	29
Gambar 4.8: Hasil perkalian per baris yang terdiri dari bagian komponen aproksimasi dan bagian komponen detail horizontal, detail vertikal dan detail diagonal.....	30
Gambar 4.9 : Representasi citra hasil dekomposisi level 1.....	30
Gambar 4.10 : Skema hasil dekomposisi level 1.	31
Gambar 4.11 : Representasi citra hasil dekomposisi level 2. Kiri:Citra Asli. Kanan: Hasil Dekomposisi.	31
Gambar 4.12: Representasi citra hasil dekomposisi level 2.....	32
Gambar 4.13: Proses fusi antara 2 <i>frame</i> yang saling berdekatan.	32
Gambar 4.14: Representasi proses rekonstruksi wavelet.....	35
Gambar 4.15: Aturan wavelet fusion pada proses reduksi <i>noise</i> hujan.	36
Gambar 4.16: Representasi citra hasil dekomposisi level 1.....	39
Gambar 4.17: Representasi citra hasil fusi. (a) <i>frame</i> ke-3. (b) <i>frame</i> ke-4. (c) <i>frame</i> hasil fusi.....	40
Gambar 5.2 : Hujan deras <i>background</i> diam objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.	44
Gambar 5.3 : Hujan sedang <i>background</i> diam objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.	46
Gambar 5.4 : Hujan rendah <i>background</i> diam objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.	48

Gambar 5.5 : Hujan deras <i>background</i> diam objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.....	50
Gambar 5.6 : Hujan sedang <i>background</i> diam objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.....	52
Gambar 5.7 : Hujan rendah <i>background</i> diam objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.....	54
Gambar 5.8 : Hujan deras <i>background</i> diam objek diam. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.....	56
Gambar 5.9: Hujan sedang <i>background</i> diam objek diam. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.....	58
Gambar 5.10: Hujan rendah <i>background</i> diam objek diam. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.....	60
Gambar 5.11: Hujan deras <i>background</i> bergerak objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.	62
Gambar 5.12: Hujan sedang <i>background</i> bergerak objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.	65
Gambar 5.13: Hujan rendah <i>background</i> bergerak objek bergerak. (a) <i>frame</i> sebelum reduksi. (b) <i>frame</i> sesudah reduksi.	67

DAFTAR TABEL

Tabel 2.1 : Koefisien <i>scaling function</i> db2 – db10 (<i>lowpass filter</i>)	10
Tabel 2.2 : Interpretasi Nilai PSNR	20
Tabel 5.1 : <i>Dataset</i> video uji coba reduksi <i>noise</i>	42
Tabel 5.2 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan deras <i>background</i> diam, objek bergerak.....	45
Tabel 5.3 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan deras <i>background</i> diam, objek bergerak.....	45
Tabel 5.4 : Tabel 5.4: Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan deras <i>background</i> diam, objek bergerak.	46
Tabel 5.5 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi hujan sedang <i>background</i> diam, objek bergerak.....	47
Tabel 5.6 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi hujan sedang <i>background</i> diam, objek bergerak.....	47
Tabel 5.7 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi hujan sedang <i>background</i> diam, objek bergerak.....	48
Tabel 5.8 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi hujan rendah <i>background</i> diam, objek bergerak.	49
Tabel 5.9 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi hujan rendah <i>background</i> diam, objek bergerak.	49
Tabel 5.10 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi hujan rendah <i>background</i> diam, objek bergerak.	50
Tabel 5.11 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi hujan deras <i>background</i> diam tanpa objek.....	51
Tabel 5.12 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi hujan deras <i>background</i> diam tanpa objek.....	51
Tabel 5.13 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi hujan deras <i>background</i> diam tanpa objek.....	52
Tabel 5.14 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi hujan sedang <i>background</i> diam tanpa objek.	53

Tabel 5.15 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi hujan sedang <i>background</i> diam tanpa objek.....	53
Tabel 5.16 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi hujan sedang <i>background</i> diam tanpa objek.....	54
Tabel 5.17 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan rendah <i>background</i> diam tanpa objek.....	55
Tabel 5.18 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan rendah <i>background</i> diam tanpa objek.....	55
Tabel 5.19 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan rendah <i>background</i> diam tanpa objek.....	56
Tabel 5.20 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan deras <i>background</i> diam objek diam.....	57
Tabel 5.21 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan deras <i>background</i> diam objek diam.....	57
Tabel 5.22 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan deras <i>background</i> diam objek diam.....	58
Tabel 5.23 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan sedang <i>background</i> diam objek diam.....	59
Tabel 5.24 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan sedang <i>background</i> diam objek diam.....	59
Tabel 5.25 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan sedang <i>background</i> diam objek diam.....	60
Tabel 5.26 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan rendah <i>background</i> diam objek diam.....	61
Tabel 5. 27 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan rendah <i>background</i> diam objek diam.....	61
Tabel 5.28 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan rendah <i>background</i> diam objek diam.....	62
Tabel 5.29 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan deras <i>background</i> bergerak objek bergerak.....	63

Tabel 5.30 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan deras <i>background</i> bergerak objek bergerak.....	64
Tabel 5.31 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan deras <i>background</i> bergerak objek bergerak.....	64
Tabel 5.32 : Perbandingan nilai PSNR <i>BayeshShrink</i> dan waktu komputasi data hujan sedang <i>background</i> bergerak objek bergerak.....	66
Tabel 5.33 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan sedang <i>background</i> bergerak objek bergerak.....	66
Tabel 5.34 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan sedang <i>background</i> bergerak objek bergerak.....	67
Tabel 5.35 : Perbandingan nilai PSNR <i>BayesShrink</i> dan waktu komputasi data hujan rendah <i>background</i> bergerak objek bergerak.....	68
Tabel 5.36 : Perbandingan nilai PSNR <i>VishuShrink</i> dan waktu komputasi data hujan rendah <i>background</i> bergerak objek bergerak.....	69
Tabel 5.37 : Perbandingan nilai PSNR <i>NormalShrink</i> dan waktu komputasi data hujan rendah <i>background</i> bergerak objek bergerak.....	69

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi selalu berkembang sepanjang sejarah peradaban manusia di segala bidang. Dalam setiap perkembangan teknologi selalu diikuti dengan pengembangan dari teknologi sebelumnya atau dengan penemuan baru. Sejak ditemukannya energi listrik perlahan teknologi analog digantikan oleh teknologi digital. Dalam perkembangannya, penggunaan teknologi digital yang awalnya hanya terbatas untuk pengolahan data sederhana berkembang pada pengolahan data yang lebih kompleks dan luas.

Saat ini penggunaan kamera digital tidak hanya terbatas pada pengambilan gambar saja namun juga pada pengambilan video digital. Selain digunakan di dalam ruangan, video digital juga sering digunakan di luar ruangan seperti pengawasan lalu lintas, estimasi kecepatan kendaraan, klasifikasi jenis kendaraan, perhitungan jumlah kendaraan bergerak, dan sebagainya. Akan tetapi, cuaca buruk mempengaruhi video digital yang dihasilkan, salah satunya adalah pengambilan video digital pada saat hujan. Objek yang diamati seringkali tidak jelas karena terhalang oleh hujan. Hujan dalam video digital dianggap *noise* karena mampu menghalangi objek yang sedang di amati. Tentu hal ini menyebabkan ketidaknyamanan pada pengguna, terlebih jika *noise* hujan ini muncul di saat yang sangat penting. Sehingga dibutuhkan suatu aplikasi pemrosesan citra digital untuk mengurangi atau bahkan menghilangkan keberadaan *noise* hujan pada video tersebut.

Pada penelitian sebelumnya, sudah banyak peneliti yang melakukan penelitian reduksi *noise* hujan pada video. Seperti yang sudah dilakukan oleh Garg dan Nayar (2004) dalam penelitiannya yang berjudul "*Detection and Removal of Rain from Videos*" menggunakan metode *Robust*. Kemudian Zhang (2006) dalam penelitiannya yang berjudul "*Rain Removal in Video by Combining Temporal and Chromatic Properties*" dengan menggunakan metode K-Mean

cluster dan perbedaan *frame*. Nikil Gupta (2011) juga melakukan penelitian yang berjudul "*Video Modelling and NoiseReduction in the Wavelet Domain*" mereduksi *noise* yang berupa *noise* gaussiann dengan menggunakan transformasi wavelet dan hasilnya cukup signifikan. Sedangkan Chen Zhen (2013) dalam penelitiannya yang berjudul "*A New Algorithm of Rain (Snow)Removal in Video*" menggunakan metode dekomposisi wavelet multi-level dan menyatakan bahwa hasil reduksi *noise* lebih baik daripada metode K-Mean Cluster dan metode perbedaan *frame*.

Berdasarkan uraian di atas, pada penelitian ini penulis melakukan reduksi *noise* hujan pada video dengan menerapkan transformasi wavelet dan wavelet *fusion*. Wavelet yang digunakan pada wavelet *fusion* berupa wavelet *Daubechies* yang digabungkan dengan proses *thresholding* untuk mereduksi *noise* hujan. Transformasi wavelet digunakan untuk memperoleh detail hujan pada citra video, sedangkan wavelet *fusion* digunakan untuk memperjelas *noise* hujan yang akan direduksi melalui proses *thresholding*. Proses reduksi *noise* dilakukan dengan menggabungkan wavelet *fusion* dengan *threshold*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dirumuskan suatu masalah:

1. Bagaimana menerapkan transformasi wavelet *Daubechies* untuk mereduksi *noise* hujan pada video?
2. Bagaimana kinerja transformasi wavelet *Daubechies* untuk mereduksi *noise* hujan pada video ?

1.3. Batasan Masalah

Permasalahan yang akan dibahas dalam penelitian ini dibatasi sebagai berikut:

1. Pada penelitian ini akan dilakukan reduksi *noise* hujan pada video *grayscale*.
2. Data video yang digunakan telah tersimpan di komputer dengan kecepatan 25 fps.
3. Keakuratan hasil diukur dengan menggunakan nilai PSNR.

1.4 Tujuan

Berdasarkan rumusan masalah diatas, tujuan dari penelitian ini adalah:

1. Melakukan reduksi *noise* hujan pada video dengan menggunakan transformasi wavelet *Daubechies*.
2. Menganalisis kinerja transformasi wavelet *Daubechies* untuk mereduksi *noise* hujan pada video.

1.5 Manfaat

Diharapkan penelitian ini dapat membantu penelitian di bidang ilmu komputer seperti *tracking object* yang dilakukan pada waktu hujan, penghitungan jumlah kendaraan pada waktu hujan, klasifikasi jenis kendaraan pada waktu hujan, estimasi kecepatan kendaraan pada waktu hujan dan sebagainya yang memungkinkan objek sering terhalang oleh hujan.

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Penelitian Terdahulu

Untuk mereduksi *noise* hujan dalam video telah banyak peneliti yang melakukan penelitian dengan berbagai macam metode. Garg dan Nayar (2004) dalam penelitiannya yang berjudul "*Detection and Removal of Rain from Videos*" melakukan reduksi *noise* hujan dengan menggunakan algoritma *Robust*. Reduksi *noise* hujan dilakukan dengan cara membagi image menjadi bagian hujan dan non-hujan serta menetapkan intensitas rata-rata dalam 2 piksel tetangga yang tidak dipengaruhi oleh hujan, akan tetapi metode ini tidak efektif karena hanya mampu mereduksi *noise* hujan yang terdeteksi saja.

Zhang (2006) dalam penelitiannya yang berjudul "*Rain Removal in Video by Combining Temporal and Chromatic Properties*" mereduksi *noise* hujan dengan menggunakan metode K-Mean cluster dan perbedaan *frame*. Penghapusan *noise* hujan dengan menggunakan algoritma yang menggabungkan sifat temporal chromatic dan menyatakan bahwa Gambar tidak pernah ditutupi oleh *noise* hujan.

Nikil Gupta (2011) juga melakukan penelitian yang berjudul "*Video Modelling and NoiseReduction in the Wavelet Domain*" mereduksi *noise* dengan menggunakan transformasi wavelet dan hasilnya cukup signifikan. Sedangkan Chen Zhen (2013) dalam penelitiannya yang berjudul "*A New Algorithm of Rain (Snow)Removal in Video*" menggunakan transformasi wavelet dan menyatakan bahwa hasil reduksi *noise* lebih baik daripada metode *K-Mean Cluster* dan metode perbedaan *frame*.

Pada penelitian ini, dilakukan reduksi *noise* dengan menggunakan transformasi wavelet *Daubechies* yang digabungkan dengan wavelet *fusion* dan proses *thresholding*. Proses reduksi *noise* ini melibatkan 4 proses yaitu dekomposisi dengan transformasi wavelet *Daubechies*, dari hasil dekomposisi ditentukan koefisien fusi berdasarkan aturan wavelet *fusion*, hasilnya dilewatkan melalui fungsi *thresholding* kemudian dilakukan proses rekonstruksi wavelet *Daubechies*.

2.2 Pengertian Citra Digital

Citra merupakan suatu gambaran, kemiripan atau imitasi suatu objek. Citra merupakan keluaran suatu sistem perekaman data, dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan (Sutoyo dkk, 2009).

Citra terdiri dari citra diam dan citra bergerak. Citra diam merupakan citra tunggal yang tidak bergerak. Sedangkan citra bergerak merupakan rangkaian citra diam yang ditampilkan secara berurutan sehingga memberi kesan pada mata sebagai gambar yang bergerak.

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan nilai dari fungsi f pada sembarang koordinat (x,y) merupakan nilai intensitas cahaya yang merupakan representasi dari warna cahaya yang ada pada citra kontinu (Gonzales dan Woods, 2001).

2.3 Video

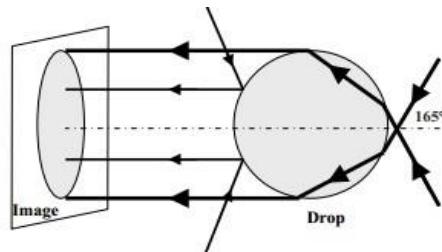
Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film seluloid, sinyal elektronik atau media digital. Alan C. Bovik dalam bukunya *Handbook of Image and Video Processing* menjelaskan bahwa video digital juga merupakan hasil sampling dan kuantisasi dari video analog (Bovik, 2000). Secara mendasar, tidak ada perbedaan proses sampling dan kuantisasi antara citra digital dan video digital.

Video digital pada dasarnya tersusun atas serangkaian citra digital diam (*frame*) berurutan yang direkam dan ditampilkan secara *sequensial* (bersambung) dalam suatu satuan waktu dengan kecepatan tertentu. Jika laju *frame* cukup tinggi maka mata manusia melihatnya sebagai rangkaian gerak yang kontinu sehingga tercipta ilusi gerak yang halus dan pergerakan obyek yang ada dalam citra digital tersebut seolah-olah hidup. Kecepatan sebuah video dalam menampilkan rangkaian gambar tersebut disebut *frame rate* dengan satuan fps (*frame per*

second). Semakin besar nilai *frame rate*, maka akan semakin halus pergerakan yang ditampilkan dan akan terkesan semakin hidup dan nyata (Solomon dan Breckon, 2013). Bagaimanapun juga, video analog yang kita lihat sehari-hari seperti tampilan pada TV analog, sebenarnya bukan sesuatu yang benar-benar kontinyu melainkan terdiri dari beberapa citra analog yang diputar secara cepat sehingga terlihat seperti bergerak.

2.4 Noise Hujan

Noise (derau) merupakan gambar atau piksel yang mengganggu kualitas citra dan bukan merupakan bagian dari citra. Gangguan tersebut umumnya berupa variasi intensitas suatu piksel yang tidak berkorelasi dengan piksel-piksel tetangga. Menurut (Garg and K. Nayar, 2004) dalam jurnalnya mengatakan bahwa gambar tetesan hujan merupakan hasil dari pantulan cahaya pada permukaan dan pantulan dari dalam. Seperti yang terlihat pada Gambar 2.1.



Gambar 2.1 : Ilustrasi gambar tetesan hujan.

Tetesan hujan juga bergerak dengan cepat menyerupai garis-garis tipis dan mampu menghalangi citra yang ada di belakangnya. Sehingga, dengan alasan inilah hujan dalam video ini dianggap sebagai *noise* karena mengganggu kualitas citra video dan mampu menghalangi objek yang ada dibelakangnya. Seperti yang terlihat pada Gambar 2.2



Gambar 2.2 : Gambar *noise* hujan. (a) *frame* tanpa *noise* hujan (b) *frame* dengan *noise* hujan.

2.5 Transformasi Wavelet

Kata wavelet berasal dari bahasa Prancis, ondelette yang berarti gelombang kecil. Onde berarti gelombang, diterjemahkan kedalam bahasa Inggris menjadi wave dan digabung dengan kata asli sehingga menjadi wavelet. Ditemukan oleh Jean Morlet dan Alex Grossmann diawal tahun 1980-an. Transformasi wavelet merupakan metode aproksimasi dari suatu fungsi berbasis wavelet $\psi(t)$ dengan bantuan lokalisasi waktu dan frekuensi.

Transformasi Wavelet dibagi menjadi dua yaitu Transformasi Wavelet Kontinu (TWK) dan Transformasi Wavelet Diskrit (TWD). Semua fungsi yang digunakan dalam transformasi TWK dan TWD diturunkan dari mother wavelet melalui translasi pergeseran dan penskalaan / kompresi. Mother wavelet merupakan fungsi dasar yang digunakan dalam transformasi wavelet. Karena mother wavelet menghasilkan semua fungsi wavelet yang digunakan dalam transformasi melalui translasi dan penskalaan, maka mother wavelet juga menentukan karakteristik dari transformasi wavelet yang dihasilkan.

Dalam transformasi wavelet, penggambaran sebuah skala waktu sinyal digital didapatkan dengan menggunakan teknik filterisasi digital. Secara garis besar, proses dalam teknik ini adalah dengan melewatkannya sinyal yang dianalisis pada filter dengan frekuensi dan skala yang berbeda.

Filterisasi sendiri merupakan sebuah fungsi yang digunakan dalam pemrosesan sinyal. Wavelet dapat direalisasikan menggunakan iterasi filter dengan penskalaan. Resolusi dari sinyal merupakan rata-rata dari jumlah detail

informasi dalam sinyal yang ditentukan melalui filterisasi. Sedangkan skalanya didapatkan dengan *up-sampling* dan *down-sampling* (*sub-sampling*).

2.5.1 Fungsi skala (*Scaling function*)

Wavelet juga merupakan sebuah basis. Basis wavelet berasal dari sebuah fungsi penskalaan atau dikatakan juga sebuah *scaling function*. *Scaling function* memiliki sifat yaitu dapat disusun dari sejumlah salinan dirinya yang telah didilasikan, ditranslasikan dan diskalakan. Fungsi ini diturunkan dari Persamaan dilasi (*dilation equation*), yang dianggap sebagai dasar dari teori wavelet [7]. Persamaan dilasi berbunyi demikian :

$$\phi(x) = \sum C_k \phi(2x - k) \quad (2.1)$$

dari persamaan *scaling function* ini dapat dibentuk Persamaan wavelet (*wavelet function*) dengan bentuk sebagai berikut :

$$\varphi^0 = \sum_k (-1)^k - C_{1-k} \phi(2x - k) \quad (2.2)$$

dengan:

C_k = koefisien penskalaan/koefisien transformasi,

k = indeks koefisien penskalaan.

Dari mother wavelet ini kemudian dapat dibentuk wavelet-wavelet berikutnya ψ_1, ψ_2 , dan seterusnya) dengan cara mendilasikan (memampatkan atau meregangkan) dan menggeser mother wavelet.

Terdapat bermacam-macam *scaling function* yang dapat membentuk wavelet. Berdasarkan *scaling function* inilah basis wavelet memiliki nama yang berbeda-beda diantaranya wavelet Haar, *Daubechies* , Coiflets, Symlet, Meyer, Morlet, dan Mexican Hat.

2.5.2. Wavelet *Daubechies*

Nama wavelet *Daubechies* berasal dari seorang pakar yang bernama Ingrid *Daubechies*. Wavelet *Daubechies* merupakan keluarga dari transformasi wavelet diskrit. Memiliki derajat *vanishing moment* yang maksimal. Derajat *vanishing moment* merupakan parameter untuk menyatakan kemampuan mengaproksimasi suatu sinyal. Semakin besar derajat *vanishing moment*, maka semakin kecil galat/ *eror* aproksimasi yang terjadi [7]. Penamaan Wavelet *Daubechies* ditulis dbN, dengan N menyatakan orde yang menggambarkan jumlah koefisien filternya dan db merupakan singkatan dari *Daubechies*.

Berdasarkan Persamaan (2.1) *scaling function* wavelet *Daubechies* yang selanjutnya disebut dengan *lowpass filter* (h_k) dapat dilihat pada Tabel 2.1.

Tabel 2.1 : Koefisien *scaling function* db2 – db10 (*lowpass filter*)

Db2	Db4	Db6	Db8	Db10
1	-0.1294095226	0.0352262919	0.0352262919	-0.0105974018
1	0.2241438680	-0.0854412739	-0.0854412739	0.0328830117
	0.8365163037	-0.1350110200	-0.1350110200	0.0308413818
	0.4829629131	0.4598775021	0.4598775021	-0.1870348117
		0.8068915093	0.8068915093	-0.0279837694
		0.3326705530	0.3326705530	0.6308807679
			0.0352262919	0.7148465706
			-0.0854412739	0.2303778133
				-0.0105974018
				0.0328830117

Sedangkan untuk *wavelet function* berdasarkan Persamaan (2.2) yang selanjutnya disebut *highpass filter* (g_0) dapat dinyatakan dalam Persamaan :

$$g_k = (-1)^k - h_{L-k} \quad (2.3)$$

dengan :

g = koefisien *highpass filter*,

h = koefisien *lowpass filter*,

$k = 0, \dots, L$.

Scaling function dan *wavelet function* yang digunakan dapat dinyatakan dalam bentuk matriks secara umum [16] yang dapat dinyatakan sebagai berikut:

$$\begin{bmatrix} h_L & h_{L-1} & h_{L-2} & h_{L-3} & \dots & h_1 & h_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & h_L & h_{L-1} & \dots & h_3 & h_2 & h_1 & h_0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & h_5 & h_4 & h_3 & h_2 & \dots & 0 & 0 \\ & & & & & \ddots & & & & & & \\ 0 & 0 & 0 & 0 & \dots & h_L & h_{L-1} & h_{L-2} & h_{L-3} & \dots & 0 & 0 \\ & & & & & \ddots & & & & & & \\ h_{L-2} & h_{L-3} & h_{L-4} & h_{L-5} & \dots & 0 & 0 & 0 & 0 & \dots & h_L & h_{L-1} \\ g_L & g_{L-1} & g_{L-2} & g_{L-3} & \dots & g_1 & g_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & g_L & g_{L-1} & \dots & g_3 & g_2 & g_1 & g_0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & g_5 & g_4 & g_3 & g_2 & \dots & 0 & 0 \\ & & & & & \ddots & & & & & & \\ 0 & 0 & 0 & 0 & \dots & g_L & g_{L-1} & g_{L-2} & g_{L-3} & \dots & 0 & 0 \\ & & & & & \ddots & & & & & & \\ g_{L-2} & g_{L-3} & g_{L-4} & g_{L-5} & \dots & & & & & \dots & g_L & g_{L-1} \end{bmatrix} \quad (2.4)$$

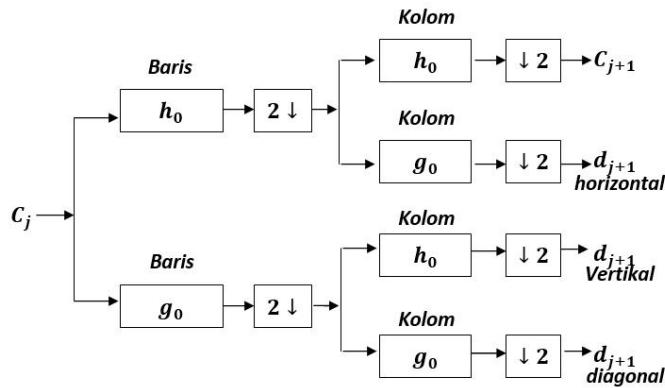
2.5.3 Proses Dekomposisi

Proses dekomposisi merupakan proses memecah suatu citra menjadi beberapa komponen dengan resolusi lebih rendah. Dua operasi dasar pada proses dekomposisi adalah *filtering* dan *down-sampling*. Pada citra 2-D proses dekomposisi dilakukan atas baris dan kolom dalam array 2 dimensi, yang masing-masing bersesuaian dengan arah horizontal dan arah vertikal.

Pada proses dekomposisi dilakukan *down-sampling* setelah proses pemfilteran untuk mengurangi jumlah *output* dari masing-masing filter sehingga sama dengan jumlah *input*. Untuk deret input $x(n)$, koefisien filter $h(n)$, dan deret output $y(n)$ diperoleh dari Persamaan (2.5) .

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (2.5)$$

Proses dekomposisi pada level j menghasilkan satu koefisien *aproksimasi* pada level $j + 1$ dan tiga koefisien *detail* (horizontal, vertikal, dan diagonal) seperti yang diilustrasikan pada Gambar 2.3.



Gambar 2.3 : Diagram dekomposisi wavelet 2D level 1.

dimana :

C_j = data input dekomposisi level ke- j ,

C_{j+1} = koefisien aproksimasi level ke - $j + 1$,

d_{j+1}^h = koefisien detail horizontal level ke - $j + 1$,

d_{j+1}^v = koefisien detail vertikal level ke - $j + 1$,

d_{j+1}^d = koefisien detail diagonal level ke - $j + 1$,

$2 \downarrow$ = *down-sampling* kolom : hilangkan kolom berindeks ganjil,

$\downarrow 2$ _{baris} = *down-sampling* baris : hilangkan baris berindeks ganjil,

\boxed{X} = konvolusi bagian baris dengan filter X,

$\boxed{\begin{matrix} \text{kolom} \\ X \end{matrix}}$ = konvolusi bagian kolom dengan filter X.

Pada Gambar 2.4, C_{j+1} menyatakan bagian koefisien yang diperoleh melalui proses filter *lowpass* pada baris dilanjutkan dengan filter *lowpass* pada kolom dan biasanya ditulis LL. Citra bagian ini mirip dan lebih halus dari citra C_j sehingga disebut dengan komponen aproksimasi. d_{j+1}^h menyatakan bagian koefisien yang diperoleh melalui proses filter *lowpass* pada baris dan dilanjutkan

dengan filter *highpass* pada kolom dan biasanya ditulis LH. d_{j+1}^v menyatakan bagian koefisien yang diperoleh melalui proses filter *highpass* pada baris dan dilanjutkan dengan filter *lowpass* pada kolom dan biasanya ditulis HL. d_{j+1}^d menyatakan bagian koefisien yang diperoleh melalui proses filter *highass* pada baris dan dilanjutkan dengan filter *highpass* pada kolom dan biasanya ditulis HH.

Representasi koefisien aproksimasi dan koefisien detail yang dihasilkan oleh dekomposisi 2-D level 2 pada citra di ilustrasikan oleh Gambar 2.4.

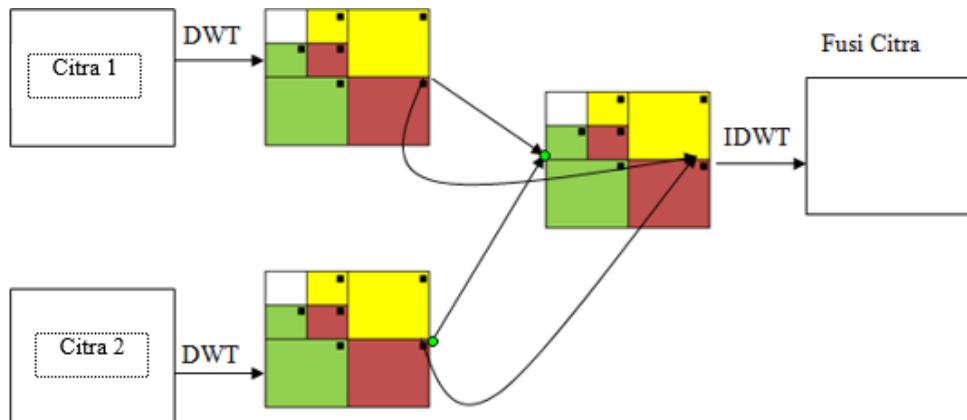
Level 2 C_2	Level 2 $d_2^{(h)}$	Detail horizontal Level 1 $d_1^{(h)}$ Subband LH
Level 2 $d_2^{(v)}$	Level 2 $d_2^{(d)}$	Detail diagonal Level 1 $d_1^{(d)}$ Subband HH
Detail vertikal Level 1 $d_1^{(v)}$ Subband HL		

Gambar 2.4 : Representasi koefisien pada dekomposisi wavelet 2D level 2.

2.5 Wavelet *fusion*

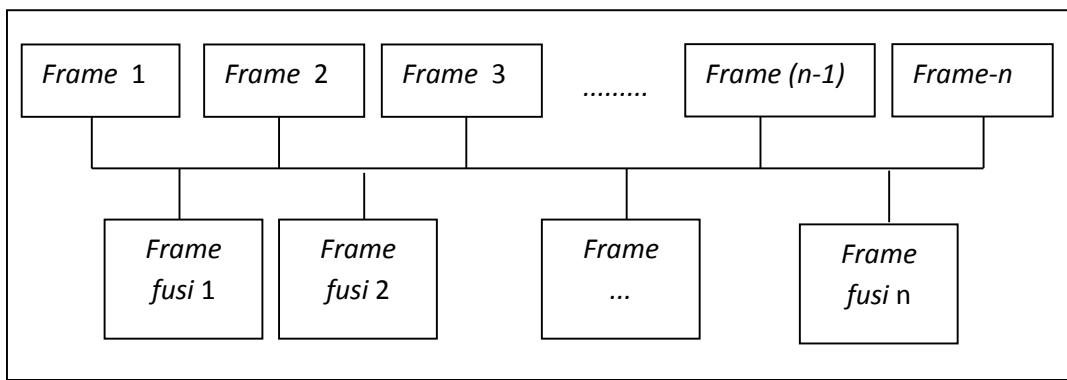
Fusi citra merupakan proses penggabungan informasi dari 2 citra pada level resolusi yang sama. Wavelet *fusion* merupakan penggabungan 2 citra melalui transformasi wavelet. Transformasi wavelet digunakan untuk memisahkan detail-detail komponen suatu citra, sehingga di peroleh informasi dari 2 citra pada tingkat level resolusi yang sama.

Berdasarkan [8], metode transformasi wavelet sangat baik digunakan untuk metode fusi karena transformasi wavelet mampu membagi citra kedalam frekuensi tinggi dan frekuensi rendah dengan resolusi yang sama seperti ditunjukkan pada ditunjukkan pada Gambar 2.5.

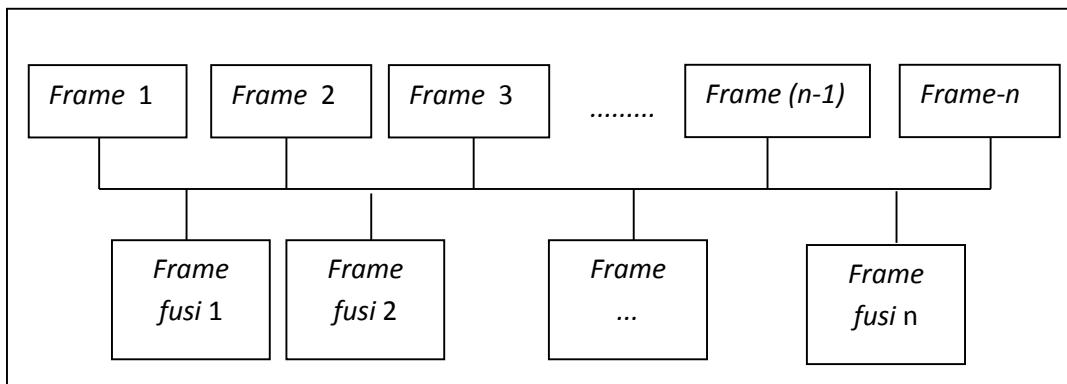


Gambar 2.5 : Diagram proses fusi citra menggunakan transformasi wavelet.

Pada Gambar 2.5 menunjukkan proses fusi dekomposisi wavelet level 2 dari 2 citra. Kotak warna menunjukkan bagian frekuensi yang sama dan digabungkan sesuai resolusi yang sama berdasarkan aturan fusi. Sedangkan kotak hitam menunjukkan bagian frekuensi yang berbeda sesuai dengan grup pixel yang sama. Sedangkan wavelet *fusion* pada video dapat dilakukan antara *frame* ke-*i* dan *frame* ke- (i+1), *frame* ke-*i* dan *frame* ke- (i+2), *frame* ke-*i* dan *frame* ke- (i+3) dan seterusnya seperti yang ditunjukkan pada Gambar 2.6.



(a)



(b)

Gambar 2.6 : Proses fusi pada video. (a) *frame* ke-i dan *frame* ke-(i+1). (b) *frame* ke-i dan *frame* ke-(i+2)

Wavelet *fusion* dapat diterapkan dengan menggunakan aturan wavelet *fusion*. Berdasarkan [9], aturan wavelet *fusion* dibuat dengan mengukur *Activity-level* menggunakan metode *window-based* yang berupa metode *spatial-frequency* [9]. Metode *spatial-frequency* dihitung berdasarkan nilai gradien lokal dan energi lokal dari masing-masing subband [8]. Aturan fusi dibuat berdasarkan parameter yang dihasilkan dari perkalian antara gradien lokal dan energi lokal dari masing-masing komponen hasil dekomposisi untuk menghitung nilai gradien lokal menggunakan Persamaan 2.6 .

$$G = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \sqrt{\Delta x f(i,j)^2 - \Delta y f(i,j)^2} \quad (2.6)$$

dan untuk menghitung energi lokal menggunakan Persamaan 2.7.

$$E = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N f(i,j)^2 \quad (2.7)$$

dengan :

- G = gradien lokal,
- $\Delta x f(i,j)$ = gradien titik (i,j) pada arah horizontal,
- $\Delta y f(i,j)$ = gradien titik (i,j) pada arah Vertikal,
- M dan N = dimensi citra.

2.6. Thresholding

Thresholding merupakan salah satu metode reduksi *noise*. Pada proses *threshold* ini, koefisien detail citra hasil transformasi pada tiap level dekomposisi yang sudah dilewatkan aturan fusi dibandingkan dengan suatu nilai *threshold-t* dan akan dilakukan proses *thresholding* dengan fungsi *soft-thresholding* yaitu:

$$\hat{d}^\lambda = T_s(d^\lambda, t) = \begin{cases} d^\lambda - t, & d^\lambda \geq t \\ 0, & |d^\lambda| < t \\ d^\lambda + t, & d^\lambda < -t \end{cases} \quad (2.8)$$

atau *hard-thresholding*

$$\hat{d}^\lambda = T_s(d^\lambda, t) = \begin{cases} d^\lambda, & |d^\lambda| \geq t \\ 0, & |d^\lambda| < t \end{cases} \quad (2.9)$$

dengan:

- \hat{d}^λ = koefisien detail *threshold*,
- λ = orientasi bidang horizontal, vertikal, dan diagonal,
- t = nilai *threshold*.

Nilai *threshold-t* dapat ditentukan dengan menggunakan *global threshold* dan *level adaptive threshold*. Pada *global threshold*, nilai *threshold* dihitung sekali dan berlaku secara global untuk seluruh level dekomposisi. Sedangkan pada *level adaptive threshold*, nilai *threshold* akan dihitung dari tiap nilai koefisien detail pada tiap level dekomposisi, sehingga akan diperoleh tiga nilai *threshold* pada setiap level dekomposisi (Chang, 2000).

Nilai *threshold* ditentukan dengan menggunakan 3 macam metode yaitu :

a. VisuShrink

VishuShrink digunakan dengan penerapan fungsi *hard-thresholding* yang dikenalkan oleh Donoho dan Johnstone pada tahun 1995 [7]. *VishuShrink* termasuk jenis *global threshold* yang dikenal cepat secara komputasi, karena penghitungan nilai *threshold* hanya dilakukan sekali. Nilai *threshold* *VishuShrink* dihitung dengan menggunakan persamaan:

$$t = \sigma \sqrt{2 \log M} \quad (2.10)$$

dengan :

σ = standar deviasi,

M = ukuran citra.

Standar deviasi σ diperoleh dari estimasi median berikut:

$$\hat{\sigma} = \frac{\text{median}(Y_{ij})}{0.6745}, Y_{ij} \in \text{subband } d_1^{(\text{diagonal})} \quad (2.11)$$

b. BayesShrink

BayesShrink termasuk dalam *level adaptive threshold* dan merupakan penerapan dari fungsi *soft-thresholding* yang ditemukan oleh Martin Vetterli dan Grace Chang [8]. Nilai *threshold* *BayesShrink* dihitung dari tiap detail pada tiap level dekomposisi dengan menggunakan persamaan:

$$t_B = \frac{\sigma^2}{\hat{\sigma}_X} \quad (2.12)$$

dengan σ^2 adalah variasi dari *noise*, sedangkan $\hat{\sigma}_X$ adalah standar deviasi yang dapat diestimasi dengan persamaan:

$$\hat{\sigma}_X = \sqrt{\max(\sigma_Y^2 - \sigma^2, 0)} \quad (2.13)$$

σ_Y^2 dapat dihitung dari tiap koefisien detail dengan menggunakan persamaan :

$$\sigma_Y^2 = \frac{1}{n^2} \sum_{i,j=1}^N Y_{ij}^2 \quad (2.14)$$

c. NormalShrink

Nilai *threshold* *NormalShrink* digunakan dengan menerapkan fungsi *soft-thresholding* dan termasuk dalam *level adaptive threshold* [13]. Nilai *threshold* dihitung menggunakan persamaan , yaitu :

$$t_B = \frac{\beta \sigma^2}{\sigma_Y} \quad (2.15)$$

dengan β adalah parameter skala yang dapat dihitung menggunakan persamaan :

$$\beta = \sqrt{\log\left(\frac{L_k}{J}\right)} \quad (2.16)$$

L_k adalah panjang koefisien pada level ke k , dan j adalah jumlah level dekomposisi.

2.7 Mean Filter

Selain menggunakan *thresholding*, reduksi *noise* juga bisa dilakukan dengan menggunakan *mean filter*. *Mean* adalah nilai rata-rata dari kumpulan data (Usman,2005). *Mean filter* dapat diperoleh dengan menggunakan Persamaan 2.17.

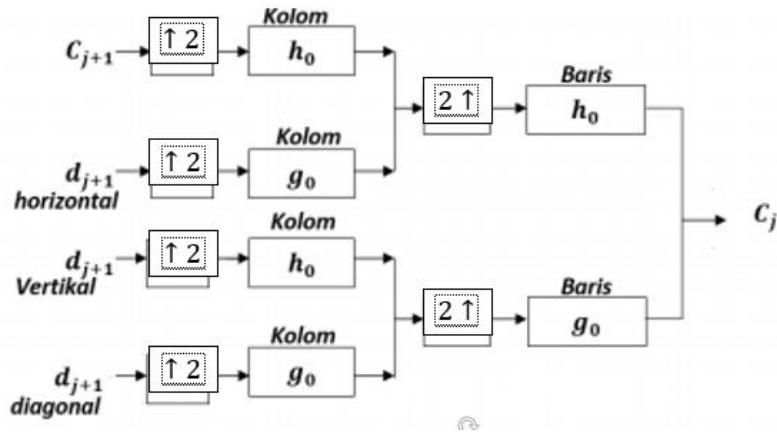
$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.17)$$

\bar{X} adalah nilai rata-rata (*mean*), n jumlah data, x_i nilai ke- i dan i adalah nilai awal.

2.8 Rekonstruksi Wavelet

Rekonstruksi wavelet merupakan kebalikan dari proses dekomposisi. Pada proses rekonstruksi dilakukan proses *up-sampling* yang dilanjutkan dengan pemfilteran. Proses *up-sampling* membentangkan sinyal input 2 kali panjang awal dan menyisipkan nilai 0 pada indeks ganjil.

Rekonstruksi dilakukan dengan cara mengkombinasikan satu koefisien aproksimasi dan tiga koefisien detail yang dihasilkan oleh proses dekomposisi. Langkah-langkah proses rekonstruksi diilustrasikan pada Gambar 2.7.



Gambar 2.7 : Diagram Rekonstruksi Wavelet 2D level 1

dimana:

C_j = data input dekomposisi level ke- j ,

C_{j+1} = koefisien aproksimasi level ke - $j + 1$,

d_{j+1}^h = koefisien detail horizontal level ke - $j + 1$,

d_{j+1}^v = koefisien detail vertikal level ke - $j + 1$,

d_{j+1}^d = koefisien detail diagonal level ke - $j + 1$,

$\uparrow 2$ = *up-sampling* kolom : sisipkan 0 pada baris berindeks ganjil,

$2 \uparrow$ = *up-sampling* baris : sisipkan 0 kolom berindeks ganjil.

2.7 Peak Signal to Noise Ratio(PSNR)

Peak Signal to Noise Ratio merupakan metode yang dapat digunakan untuk mengukur kualitas video. Metode ini membandingkan sinyal setiap *frame* video hasil reduksi dengan setiap *frame* pada video sebelum terkena *noise* hujan.

Berdasarkan [10], PSNR dirumuskan oleh persamaan :

$$PSNR = 10 \log_{10} \frac{(2^M - 1)^2}{MSE} \quad (2.16)$$

dengan $(2^M - 1)$ merupakan maksimum nilai pixel untuk M = bit *frame* video.

MSE adalah *Mean Square Error* yang dapat dihitung menggunakan persamaan:

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x, y) - I'(x, y)]^2 \quad (2.17)$$

dimana M, N adalah dimensi citra. $I(x, y)$ merupakan nilai piksel *frame* video sebelum hujan. Sedangkan $I'(x, y)$ adalah nilai piksel *frame* video setelah proses reduksi *noise*. Kualitas video yang baik ditandai dengan tingginya nilai PSNR.

Berdasarkan [14] Interpretasi nilai PSNR terhadap kualitas citra ditampilkan pada Tabel 2.2.

Tabel 2.2 : Interpretasi Nilai PSNR

Nilai PSNR	Kualitas
$\text{PSNR} > 33 \text{ dB}$	Baik Sekali
$33 \text{ dB} > \text{PSNR} > 30 \text{ dB}$	Cukup
$\text{PSNR} < 30 \text{ dB}$	Buruk

BAB 3

METODE PENELITIAN

Pada bagian ini, membahas tentang uraian mengenai penelitian yang akan dilakukan beserta block diagram reduksi *noise* hujan pada video.

3.1 Peralatan Penelitian

Pada tahap ini, peralatan yang digunakan adalah kamera untuk merekam video dan *software* Matlab untuk implementasi.

3.2 Tahapan Penelitian

Tahapan penelitian yang akan dilakukan dalam penelitian ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini, dilakukan kajian teori mengenai reduksi *noise* hujan dengan membaca jurnal-jurnal sebelumnya serta pengumpulan informasi yang berkaitan dengan transformasi wavelet dan wavelet *fusion* yang diperoleh dari berbagai sumber pustaka.

2. Pengumpulan Data

Pada tahap ini, akan dilakukan pengumpulan data video. Video diperoleh langsung di lingkungan ITS dan *youtube*.

3. Penerapan Transformasi Wavelet *Daubechies* untuk Reduksi *Noise* Hujan Pada video

Pada tahap ini, diawali dengan proses dekomposisi *frame* menggunakan transformasi wavelet *Daubechies* untuk memperoleh komponen frekuensi tinggi dan komponen frekuensi rendah. Komponen frekuensi tinggi dan komponen frekuensi rendah hasil dekomposisi wavelet akan memudahkan dalam proses fusi citra. Fusi *frame* dilakukan dengan menggabungkan komponen 2 *frame* hasil dekomposisi untuk memperjelas *noise* hujan pada *frame*. Selanjutnya dilakukan proses *thresholding* untuk mereduksi *noise*

hujan. Tahap terakhir dilanjutkan dengan proses rekonstruksi menggunakan transformasi wavelet *Daubechies*.

4. Implementasi Pada Matlab

Untuk mempermudah percobaan dan analisis hasil, implementasi dilakukan pada Matlab yang diharapkan proses penggerjaan dapat dilakukan dengan tepat.

5. Percobaan dan Analisis Hasil

Dengan menggunakan nilai PSNR, akan dianalisis hasil dari proses reduksi *noise* hujan pada video.

6. Publikasi

7. Penyusunan Tesis

3.3 Block Diagram Reduksi Noise Hujan pada Video

Block diagram dari metode penelitian dimulai dari *input* / masukan hingga hasil reduksi yang berupa video. Bolock diagram dari metode ini ditunjukkan pada Gambar 3.1.

1. *Input*

Langkah pertama proses reduksi *noise* hujan pada video yaitu *input* video yang telah tersimpan di komputer yang diperoleh dari pengambilan di lingkungan ITS dan dari *youtube*.

2. Konversi video RGB ke *grayscale*

Video hasil input di konversi ke bentuk *grayscale* untuk mempermudah dalam proses proses berikutnya dan untuk mengestimasi waktu komputasi.

3. Ekstrak video ke dalam urutan *frame*.

Video *grayscale* dipecah menjadi beberapa rangkaian *frame*.

4. Dekomposisi wavelet

Dekomposisi wavelet dilakukan pada *frame* hasil ekstrak video dengan tujuan untuk memperoleh komponen frekuensi tinggi dan komponen frekuensi rendah dengan resolusi yang sama.

5. Menerapkan aturan wavelet fusi

Aturan fusi diterapkan setelah diperoleh komponen frekuensi tinggi dan komponen frekuensi rendah dari tiap-tiap *frame* untuk memperjelas *noise*

hujan dengan menggabungkan 2 *frame* dengan resolusi yang sama menggunakan metode wavelet fusi sehingga diperoleh *frame* hasil fusi.

6. Proses *Thresholding*

Frame hasil fusi dilewatkan pada fungsi *thresholding* dengan tujuan untuk memperoleh *frame* yang *noise* nya sudah direduksi.

7. Menghitung nilai PSNR

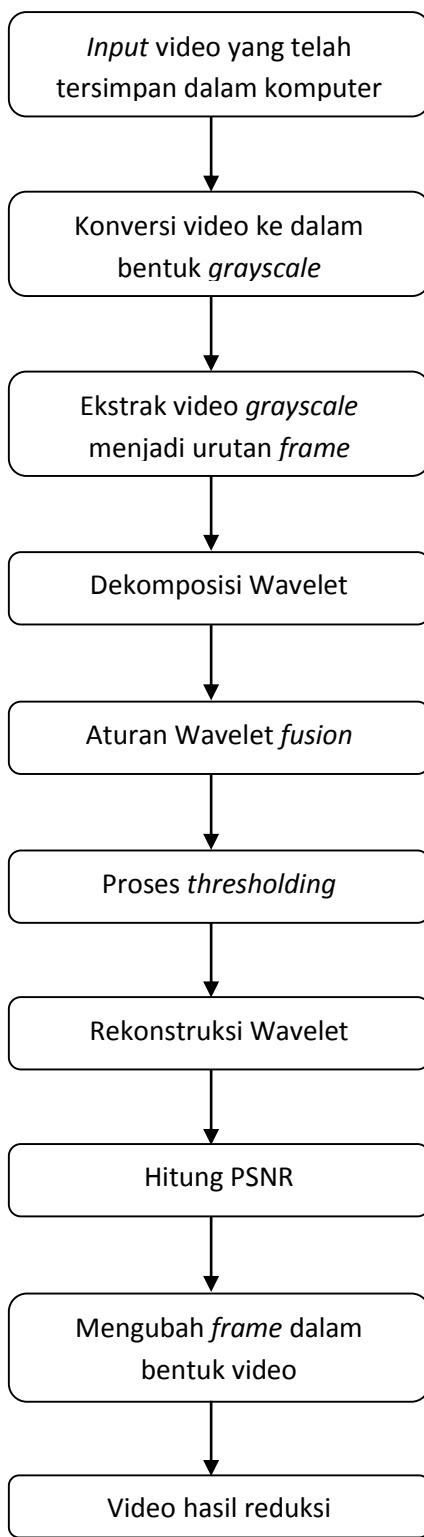
Nilai PSNR dihitung untuk mengetahui hasil dari proses reduksi *noise* hujan pada tiap-tiap *frame*.

8. Mengubah *frame* ke dalam bentuk video

Frame-frame berurutan hasil fusi di ubah kembali ke dalam bentuk video.

9. Video hasil reduksi

Hasil reduksi *noise* hujan berupa video yang *noise* hujannya sudah direduksi.



Gambar 3.1 : Block diagram proses reduksi *noise* hujan pada video

BAB 4

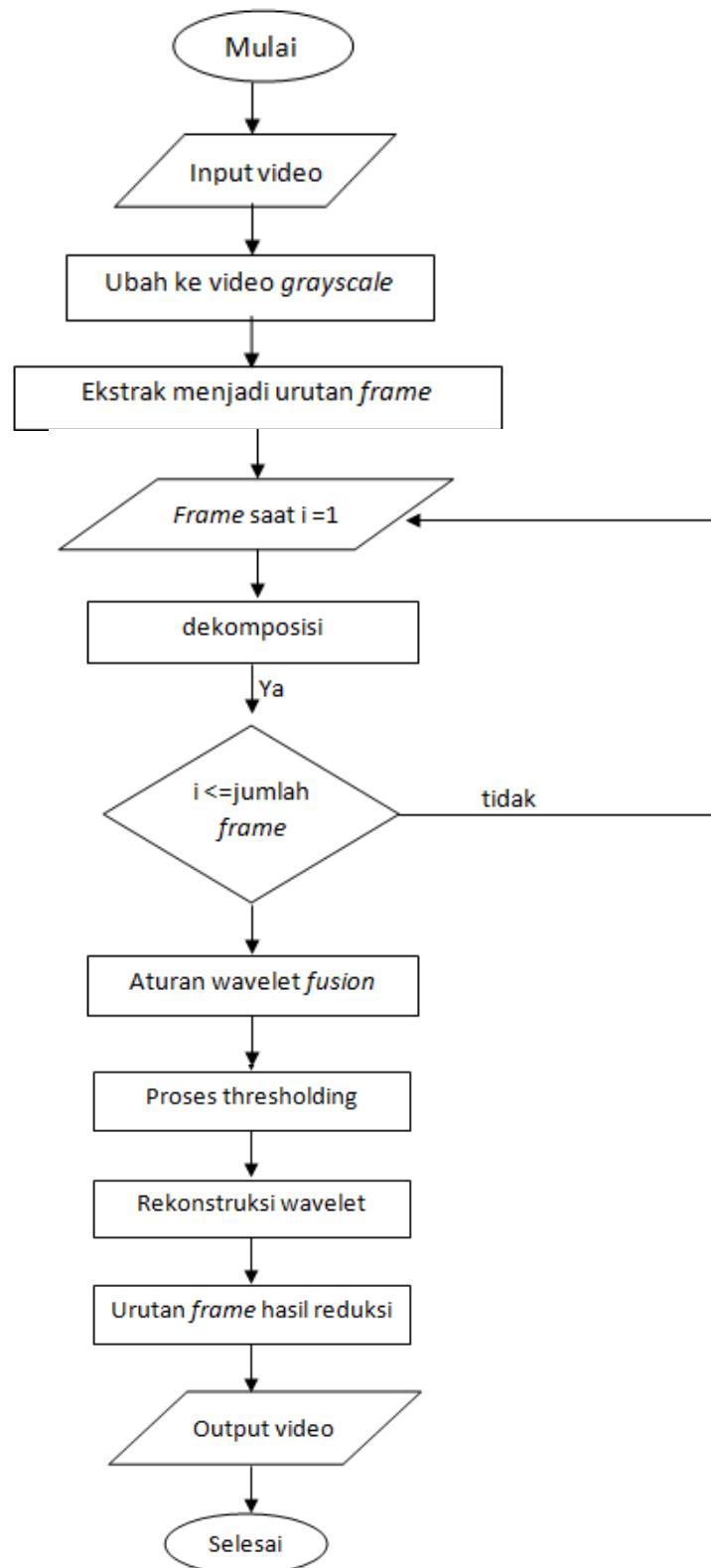
PENERAPAN TRANSFORMASI WAVELET DAUBECHIES UNTUK REDUKSI NOISE HUJAN PADA VIDEO

Pada bab ini dijelaskan mengenai perancangan proses penerapan transformasi wavelet *Daubechies* untuk reduksi *noise* hujan pada video. Perancangan proses diawali dengan proses dekomposisi dengan menggunakan transformasi wavelet *Daubechies*. Selanjutnya aturan fusi diterapkan untuk memperjelas *noise* hujan pada video dilanjutkan dengan proses *thresholding*. Tahapan terakhir adalah implementasi perangkat lunak yang tujuannya untuk mempermudah analisis dan uji coba hasil.

4.1 Perancangan Proses

Pada Gambar 4.1 menjelaskan tentang perancangan proses yang disusun dengan tujuan menjaga agar proses reduksi *noise* hujan pada video berjalan dengan lancar dan teratur, sehingga bisa memperoleh informasi dari setiap proses.

Proses awal yaitu berupa *input* video yang diambil dari penyimpanan komputer yang selanjutnya diubah dalam bentuk video *grayscale*. Penggunaan video *grayscale* bertujuan untuk mempercepat waktu komputasi dalam proses reduksi *noise* hujan. Selanjutnya video *grayscale* diekstrak ke dalam urutan *frame* yang nantinya akan didekomposisi melalui transformasi wavelet *Daubechies*. Bagian-bagian komponen *frame* hasil dekomposisi akan dilibatkan aturan wavelet *fusion* dengan tujuan agar *noise* hujan semakin jelas. Langkah berikutnya adalah komponen-komponen *frame* yang sudah di libatkan aturan wavelet *fusion* di lewatkan melalui proses *thresholding*. Selanjutnya direkonstruksi menggunakan invers transformasi wavelet *Daubechies* dan terakhir *frame-frame* diubah kembali ke dalam bentuk video.

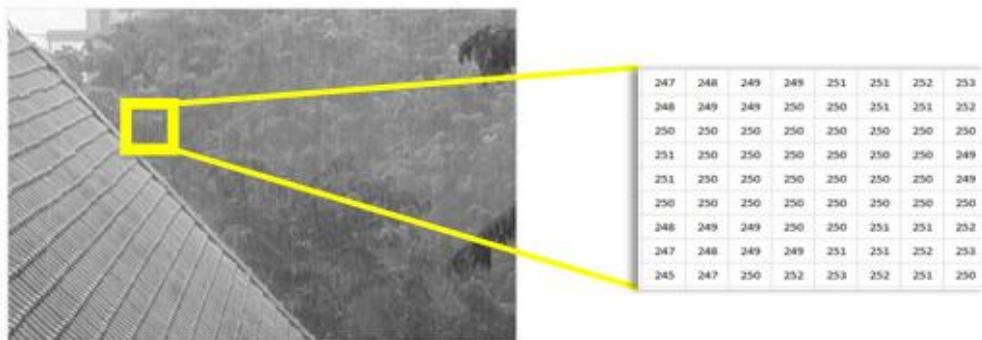


Gambar 4.1 : Diagram alir perancangan proses

4.1.1 Dekomposisi Wavelet

Proses dekomposisi wavelet diperoleh melalui operasi perkalian konvolusi antara matriks koefisien *lowpass filter* dan *highpass filter Daubechies* dengan matriks *frame* yang berupa *grayscale*. Berdasarkan diagram proses dekomposisi pada Gambar 2.4, ilustrasi proses dekomposisi wavelet *Daubechies* adalah sebagai berikut:

Misalkan diberikan citra A berukuran 8×8 dari citra asal B.



Gambar 4.2 : Representasi pixel A yang diambil dari citra Asal B

1. Langkah pertama adalah menentukan koefisien *lowpass filter* dan koefisien *highpass filter*. Untuk filter *Daubechies* db4, berdasarkan Tabel 2.1 *lowpass filter* db4 adalah :

$$h_0 = 0.4830$$

$$h_1 = 0.8365$$

$$h_2 = 0.2241$$

$$h_3 = -0.1294$$

dan 4 koefisien *highpass filter* yaitu :

$$g_0 = h_3$$

$$g_1 = -h_2$$

$$g_2 = h_1$$

$$g_3 = -h_0.$$

Yang ditunjukkan pada matriks berikut:

$$\begin{bmatrix} 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 \\ 0.2241 & -0.1294 & 0 & 0 & 0 & 0 & 0.4830 & 0.8365 \\ -0.1294 & -0.2241 & 0.8365 & 0.4830 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1294 & -0.2241 & 0.8365 & 0.4830 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1294 & -0.2241 & 0.8365 & 0.4830 \\ 0.8365 & 0.4830 & 0 & 0 & 0 & 0 & -0.1294 & -0.2241 \end{bmatrix}$$

Gambar 4.3 : Matriks filter wavelet *Daubechies* (db4)

dan bentuk matrik pixel A adalah :

$$\begin{bmatrix} 247 & 248 & 249 & 249 & 251 & 251 & 252 & 253 \\ 248 & 249 & 249 & 250 & 250 & 251 & 251 & 252 \\ 250 & 250 & 250 & 250 & 250 & 250 & 250 & 250 \\ 251 & 250 & 250 & 250 & 250 & 250 & 250 & 249 \\ 251 & 250 & 250 & 250 & 250 & 250 & 250 & 249 \\ 250 & 250 & 250 & 250 & 250 & 250 & 250 & 250 \\ 248 & 249 & 249 & 250 & 250 & 251 & 251 & 252 \\ 247 & 248 & 249 & 249 & 251 & 251 & 252 & 253 \end{bmatrix}$$

Gambar 4.4 : Matriks *frame* pada titik tertentu

2. Langkah kedua adalah mengalikan matriks *filter* wavelet *Daubechies* dengan A dengan cara mengalikan per kolom secara konvolusi.

Contoh untuk kolom pertama:

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ g_2 & g_3 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{bmatrix} * \begin{bmatrix} 247 \\ 248 \\ 250 \\ 251 \\ 251 \\ 250 \\ 248 \\ 247 \end{bmatrix} = \begin{bmatrix} 350.2986 \\ 354.6106 \\ 353.9730 \\ 349.6610 \\ 0.3534 \\ 0.6124 \\ -0.3534 \\ -0.6124 \end{bmatrix}$$

Gambar 4.5 : Perkalian konvolusi antara matriks filter wavelet *Daubechies* (db4) dengan bagian kolom matriks A.

Hasil perkalian per kolom akan menghasilkan matriks sebagai berikut:

350.2986	351.7475	352.2305	353.0670	354.0330	354.8695	355.3525	356.8014
354.6106	353.5500	353.5500	353.5500	353.5500	353.5500	353.5500	352.4894
353.9730	353.5847	353.4553	353.6794	353.4206	353.6447	353.5153	353.1270
349.6610	351.0752	352.1358	352.4894	354.6106	354.9642	356.0248	357.4390
0.3534	0.4829	0.3535	0.1294	-0.1294	-0.3535	-0.4829	-0.3534
0.6124	0	0	0	0	0	0	-0.6124
-0.3534	0.1295	-0.3535	0.4830	-0.4830	0.3535	-0.1295	0.3534
-0.6124	-0.6124	0.0000	-0.6124	0.6124	0.0000	0.6124	0.6124

Gambar 4.6: Hasil perkalian per kolom yang terdiri dari bagian komponen aproksimasi dan bagian komponen detail

- Langkah ketiga adalah Mengalikan matriks *filter wavelet Daubechies* dengan matriks hasil perkalian kolom dengan cara mengalikan perbaris secara konvolusi.

Contoh untuk baris pertama:

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ g_2 & g_3 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{bmatrix} * \begin{bmatrix} 350.2986 \\ 351.7475 \\ 352.2305 \\ 353.0670 \\ 354.0330 \\ 354.8695 \\ 355.3525 \\ 356.8014 \end{bmatrix} = \begin{bmatrix} 350.2986 \\ 354.6106 \\ 353.9730 \\ 349.6610 \\ 0.3534 \\ 0.6124 \\ -0.3534 \\ -0.6124 \end{bmatrix}$$

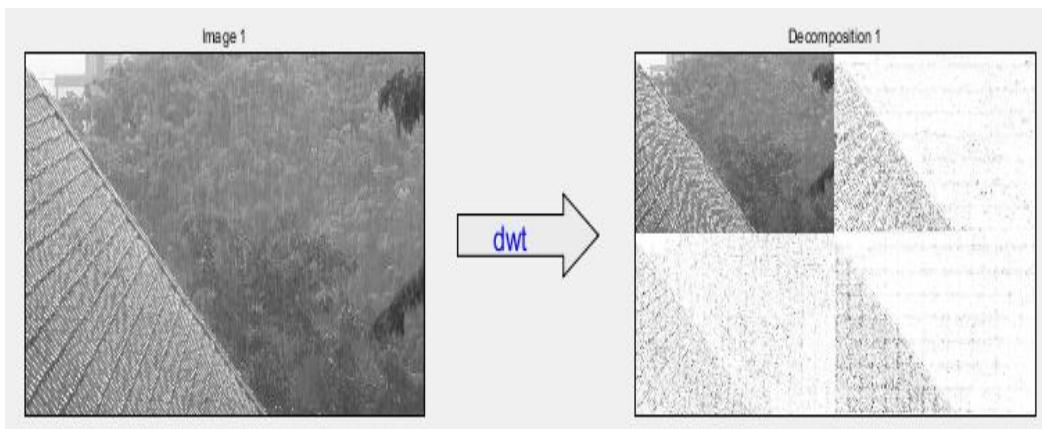
Gambar 4.7 : Perkalian konvolusi baris antara matriks filter *Daubechies* (db4) dengan hasil perkalian kolom.

Hasil perkalian perbaris akan menghasilkan matriks sebagai berikut:

496.6790	498.8866	501.3107	503.0854	-0.0458	0.0457	-0.4208	-2.8111
500.5027	499.9904	500.1277	499.3409	-0.1372	0.0000	0.5123	1.1249
500.1858	500.0117	500.0541	499.7101	-0.2042	-0.1707	0.1708	0.4364
495.8622	498.4748	501.7370	503.8876	0.3871	0.6248	-0.2624	-3.2496
0.6371	0.2957	-0.4207	-0.5122	0.0793	-0.0122	-0.1373	0.2041
0.2958	0	0.0792	-0.3750	-0.0792	0	0.2958	0.6495
-0.2041	0.0793	-0.0123	0.1371	-0.5123	-0.6373	-0.2957	-0.4206
-0.7288	-0.3750	0.3538	0.7501	0.5123	0.6495	0.1372	-0.4330

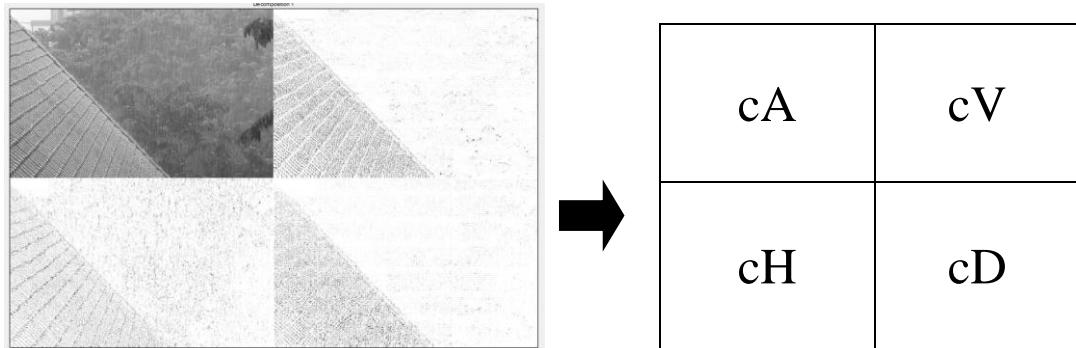
Gambar 4.8: Hasil perkalian per baris yang terdiri dari bagian komponen aproksimasi dan bagian komponen detail horizontal, detail vertikal dan detail diagonal.

Hasil perkalian diatas jika ditampilkan dalam bentuk citra, akan menghasilkan Gambar 4.9



Gambar 4.9 : Representasi citra hasil dekomposisi level 1.

Hasil dekomposisi level 1 pada Gambar 4.10 sering ditampilkan dengan skema sebagai berikut:

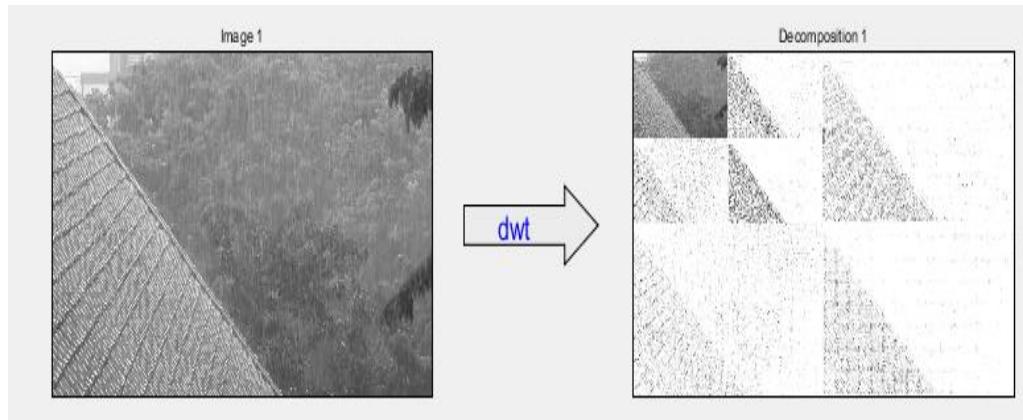


Gambar 4.10: Skema hasil dekomposisi level 1.

dimana:

- cA : Komponen Aproksimasi,
- cH : Komponen Detail Horizontal,
- cV : Komponen Detail Vertikal,
- cD : Komponen Detail Diagonal.

Untuk mendapatkan hasil dekomposisi level ke-2 , langkah-langkah diatas diulang kembali dengan mengalikan hasil komponen aproksimasi dengan matriks filter wavelet.



Gambar 4.11 : Representasi citra hasil dekomposisi level 2. Kiri:Citra Asli.
Kanan: Hasil Dekomposisi.

Dan untuk hasil dekomposisi level 2 pada Gambar 4.12 sering ditampilkan dengan skema sebagai berikut:

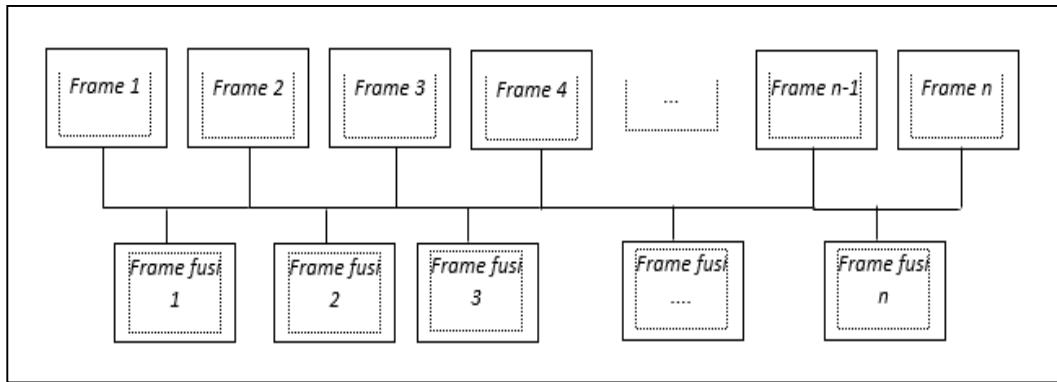


Gambar 4.12: Representasi citra hasil dekomposisi level 2.

Dekomposisi dilakukan untuk setiap *frame* secara berurutan sampai *frame ke-n* dari video sehingga masing-masing *frame* mempunyai komponen hasil dekomposisi sesuai dengan tingkat level dekomposisi yang dilakukan.

4.1.2 Aturan Wavelet Fusion

Wavelet fusion dilakukan untuk menggabungkan 2 *frame* dengan tujuan agar *noise* hujan semakin jelas. Pada penelitian ini, 2 *frame* yang digunakan berupa 2 *frame* yang saling berdekatan yaitu antara *frame-1* dan *frame-2*, *frame-3* dan *frame-4*, sampai *frame-n* seperti yang diGambarkan pada Gambar 4.13



Gambar 4.13: Proses fusi antara 2 *frame* yang saling berdekatan.

Sedangkan Transformasi wavelet yang digunakan berupa transformasi wavelet *Daubechies* sampai level dekomposisi ke-3.

Langkah-langkah wavelet *fusion* adalah sebagai berikut :

1. Citra di dekomposisi sehingga diperoleh empat bagian frekuensi pada masing-masing level dekomposisi, seperti yang dilakukan pada bagian 4.1.1.

2. Mengukur *level activity* komponen hasil dekomposisi pada bagian 4.1.1, yaitu dengan cara menghitung gradien lokal dari masing-masing komponen hasil dekomposisi menggunakan Persamaan (2.5) dan Persamaan (2.6) .

Sebagai ilustrasi, misalkan koefisien hasil dekomposisi pada *frame* 1 adalah:

$$CA = \begin{bmatrix} 496.6790 & 498.8866 & 501.3107 & 503.0854 \\ 500.5027 & 499.9904 & 500.1277 & 499.3409 \\ 500.1858 & 500.0117 & 500.0541 & 499.7101 \\ 495.8622 & 498.4748 & 501.7370 & 503.8876 \end{bmatrix}$$

$$CV = \begin{bmatrix} -0.0458 & 0.0457 & -0.4208 & -2.8111 \\ -0.1372 & 0 & 0.5123 & 1.1249 \\ -0.2042 & -0.1707 & 0.1708 & 0.4364 \\ 0.3871 & 0.6248 & -0.2624 & -3.2496 \end{bmatrix}$$

$$CH = \begin{bmatrix} 0.6371 & 0.2957 & -0.4207 & -0.5122 \\ 0.2958 & 0 & 0.0792 & -0.3750 \\ -0.2042 & 0.0793 & -0.0123 & 0.1371 \\ -0.2041 & -0.3750 & 0.3538 & 0.7501 \end{bmatrix}$$

$$CD = \begin{bmatrix} 0.0793 & -0.0122 & -0.1373 & 0.2041 \\ -0.0792 & 0 & 0.2958 & 0.6495 \\ -0.5123 & -0.6373 & -0.2957 & -0.4206 \\ 0.5123 & 0.6495 & 0.1372 & -0.4330 \end{bmatrix}$$

Dengan menggunakan rumus Gradien lokal :

$$G = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \sqrt{\Delta x f(i,j)^2 - \Delta y f(i,j)^2}$$

Dan rumus energi lokal :

$$E = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N f(i,j)^2$$

Untuk *frame* 1 dekomposisi level 1 diperoleh gradien lokal dan energi lokal sebagai berikut:

- a) Gradien lokal dari CA (komponen aproksimasi) adalah 327.4927
- b) Gradien lokal dari CV (komponen detail vertikal) adalah 0.1118
- c) Gradien lokal dari CH (komponen detail Horizontal) adalah 0.0587
- d) Gradien lokal dari CD (komponen detail Diagonal) adalah 0.1125
- e) Energi lokal dari CA (komponen aproksimasi) adalah 1.4049×10^5
- f) Energi lokal dari CV (komponen detail vertikal) adalah 0.0352
- g) Energi lokal CH (komponen detail Horizontal) adalah 0.0508
- h) Energi lokal dari CD (komponen detail Diagonal) adalah 0.0547

Untuk *frame* kedua sampai *frame* ke-n dilakukan perhitungan gradien lokal dan energi lokal seperti perhitungan diatas.

3. Langkah selanjutnya adalah membandingkan gradien lokal dari masing-masing komponen yang searah dan mengambil nilai maksimum antara dua *frame* yang berurutan.

4.1.3 Proses *thresholding*

Pada proses *thresholding* ini, koefisien detail citra hasil transformasi pada tiap level dekomposisi yang sudah dilewatkan aturan fusi dibandingkan dengan suatu nilai *threshold* - t. Langkah-langkah proses *thresholding* adalah sebagai berikut:

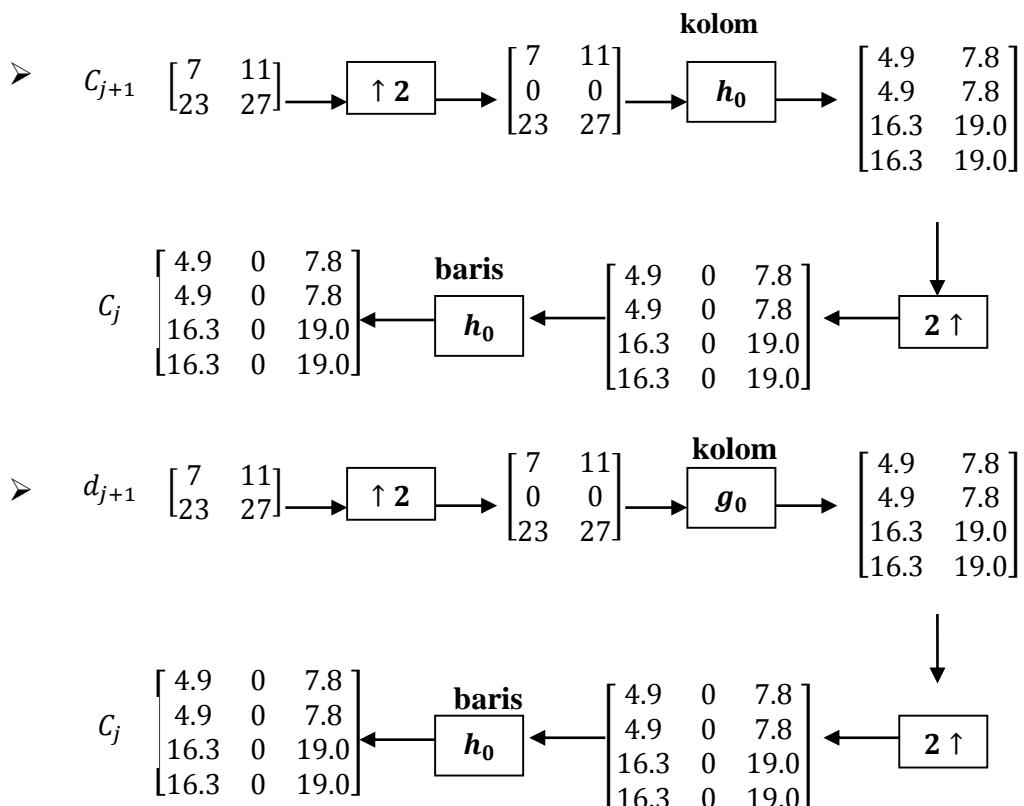
1. Tentukan nilai *threshold*-t masing-masing koefisien detail hasil transformasi pada tiap level dekomposisi yang sudah dilewatkan aturan fusi dengan menggunakan rumus *VishuShrink*, *Bayesshrink*, dan *NormalShrink* pada Persamaan 2.9 sampai 2.15.
2. Melakukan penskalaan pada masing-masing *threshold* – t dengan menghitung $\max(|Y_{i,j}|)/0.6745$.
3. Lakukan *thresholding* pada koefisien-koefisien detail hasil transformasi pada tiap level dekomposisi dengan nilai *threshold*-t yang bersesuaian.
4. Melakukan proses rekonstruksi wavelet.

4.1.4 Rekonstruksi wavelet

Rekonstruksi wavelet dilakukan terhadap koefisien-koefisien yang sudah melalui proses aturan *fusi* dan proses *thresholding* pada masing-masing *frame*..

Berdasarkan Persamaan illustrasi Gambar 2.8, langkah-langkah rekonstruksi wavelet adalah sebagai berikut:

1. Langkah pertama adalah menentukan matriks *lowpass filter* dan *highpass filter* wavelet *Daubechies*.
2. Melakukan *up-sampling* baris yaitu menyisipkan 0 pada masing-masing baris berindeks ganjil pada koefisien aproksimasi C_{j+1} , beserta tiga orientasi koefisien *detail* (d_{j+1} horizontal, d_{j+1} vertikal, dan d_{j+1} diagonal).
3. Mengalikan matriks kolom koefisien aproksimasi C_{j+1} dengan matriks *lowpass filter* dan mengalikan matriks kolom koefisien detail horizontal d_{j+1} dengan matriks *highpass filter*. Seperti yang ditunjukkan pada Gambar 4.14.



Gambar 4.14: Representasi proses rekonstruksi wavelet.

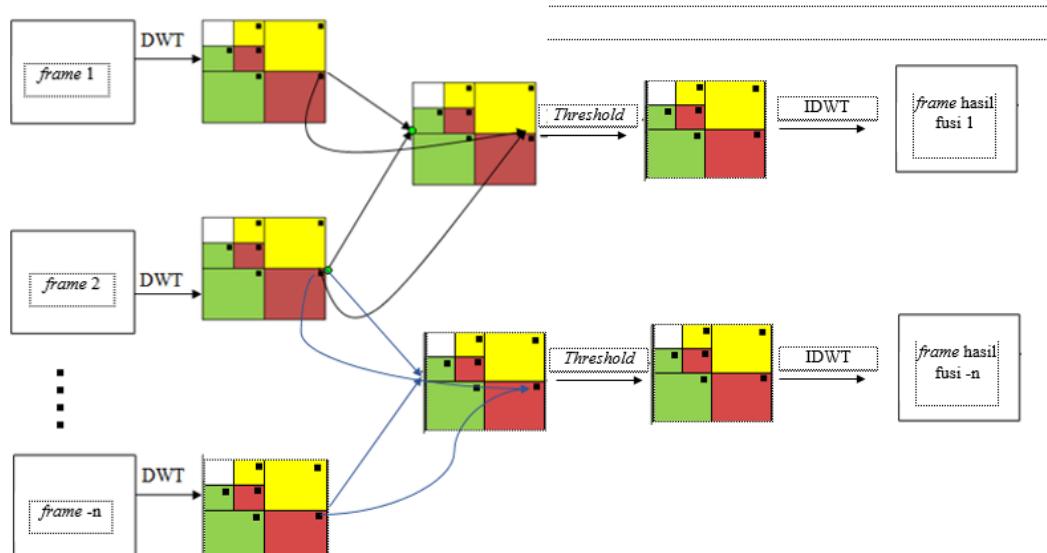
4. Uangi langkah 3 pada koefisien detail vertikal d_{j+1} dan koefisien detail diagonal d_{j+1} .

5. Lakukan *up-sampling* kolom yaitu menyisipkan 0 pada masing-masing kolom pada hasil langkah ke-3 dan langkah ke-4.

6. Mengalikan matriks baris hasil yang diperoleh pada langkah ke-5 dengan matriks *lowpass filter* untuk komponen aproksimasi dan matriks *highpass filter* untuk komponen detail.

7. Lakukan 1-6 pada tiap-tiap *frame* sehingga diperoleh *frame* hasil fusi yang *noise* hujannya sudah direduksi.

Langkah-langkah aturan wavelet *fusion* pada proses reduksi *noise* hujan dari 4.1.1 sampai 4.1.4 ditunjukkan pada Gambar 4.13



Gambar 4.15: Aturan wavelet *fusion* pada proses reduksi *noise* hujan.

4.1.5. Mengukur Kualitas Video dengan Membandingkan Nilai PSNR.

Untuk mengukur kualitas video hasil reduksi, yaitu dengan cara membandingkan nilai PSNR setiap *frame* video hasil reduksi dengan setiap *frame* pada video sebelum terkena *noise* hujan dengan menggunakan Persamaan 2.16 dan 2.17. Pada penelitian ini, proses membandingkan nilai PSNR dilakukan sebanyak 60 kali yang kemudian hasilnya dirata-rata sehingga diperoleh nilai PSNR tertinggi sesuai dengan yang diharapkan.

4.2 Implementasi Proses Reduksi *noise* Hujan Pada Video

Implementasi proses Reduksi *noise* hujan pada video terdiri dari proses *input* video, konversi video ke dalam *grayscale*, ekstraksi video menjadi urutan *frame*, proses dekomposisi dan aturan wavelet *fusion*, proses *thresholding* dan render *frame* ke dalam bentuk video.

4.2.1 Proses input Video

Proses *input* video menggunakan GUI Matlab menggunakan *sourcecode* program sebagai berikut :

```
[filename, pathname] = uigetfile({'*.avi'; '*.mp4'; '*.mpeg'}, 'Pilih  
Video'); % akses directory file video  
videoPath=strcat(pathname, filename);  
  
video=VideoReader(videoPath);  
numFrames = video.NumberOfFrames;  
vidHeight = video.Height;  
vidWidth = video.Width;  
  
movAsli = struct('cdata',zeros(vidHeight, vidWidth, 1, 'uint8'),  
'colormap', []);
```

4.2.2 Konversi video ke *grayscale*

Proses awal yang dilakukan sebelum melakukan reduksi *noise* hujan pada video yaitu dengan mengubah video berwarna ke bentuk video *grayscale* dengan tujuan untuk menyederhanakan *frame* video RGB yang terdiri dari 3 layer matrik R-layer, G-Layer dan B-Layer menjadi 1 layer matrik *grayscale* yang sudah tidak mengandung unsur warna lagi dan hanya ada derajat keabuan. Untuk melakukan konversi video ke *grayscale* menggunakan fungsi sebagai berikut:

```
numFrames = video.NumberOfFrames;  
vidHeight = video.Height;  
vidWidth = video.Width;  
movAsli = struct('cdata',zeros(vidHeight, vidWidth, 1, 'uint8'),  
'colormap', []);  
  
for i = 1 : (numFrames-1)  
    xAsli = read(video, i);  
    movAsli(i).cdata = rgb2gray(xAsli);  
end
```

4.2.3 Ekstraksi video menjadi urutan *frame*

Video diekstraksi ke dalam beberapa *frame* untuk mempermudah proses dekomposisi yang selanjutnya akan dilakukan reduksi *noise* hujan pada video. Untuk melakukan ekstraksi video ke dalam beberapa *frame* menggunakan fungsi sebagai berikut:

```
for i = 1 : (numFrames-1)
    X1 = read(video, i);
    X1 = rgb2gray(X1);

    X2 = read(video, (i+1));
    X2 = rgb2gray(X2);
end
```

4.2.4 Dekomposisi wavelet *Daubechies* dari tiap-tiap *frame*

Setelah diekstraksi menjadi urutan *frame*, maka tiap-tiap *frame* didekomposisi menggunakan transformasi wavelet *Daubechies*. Pada Gambar 5.2 ditampilkan hasil dekomposisi wavelet orde 2 (db2) level 1 yang menghasilkan 4 sub image dari 1 *frame* asli yang terdiri dari 1 bagian aproksimasi dan 3 bagian detail.

Untuk melakukan proses dekomposisi wavelet menggunakan fungsi sebagai berikut:

```
close all; clear all; clear all;

X=imread('rain_1.jpg');

[c,s]=wavedec2(X,2,'db2');
[H1,V1,D1] = detcoef2('all',c,s,1);
A1 = appcoef2(c,s,'db2',1);

V1img = wcodemat(V1,255,'mat',1);
H1img = wcodemat(H1,255,'mat',1);
D1img = wcodemat(D1,255,'mat',1);
A1img = wcodemat(A1,255,'mat',1);
```



Gambar 4.16: Representasi citra hasil dekomposisi level 1.

4.2.5 Wavelet *fusion*

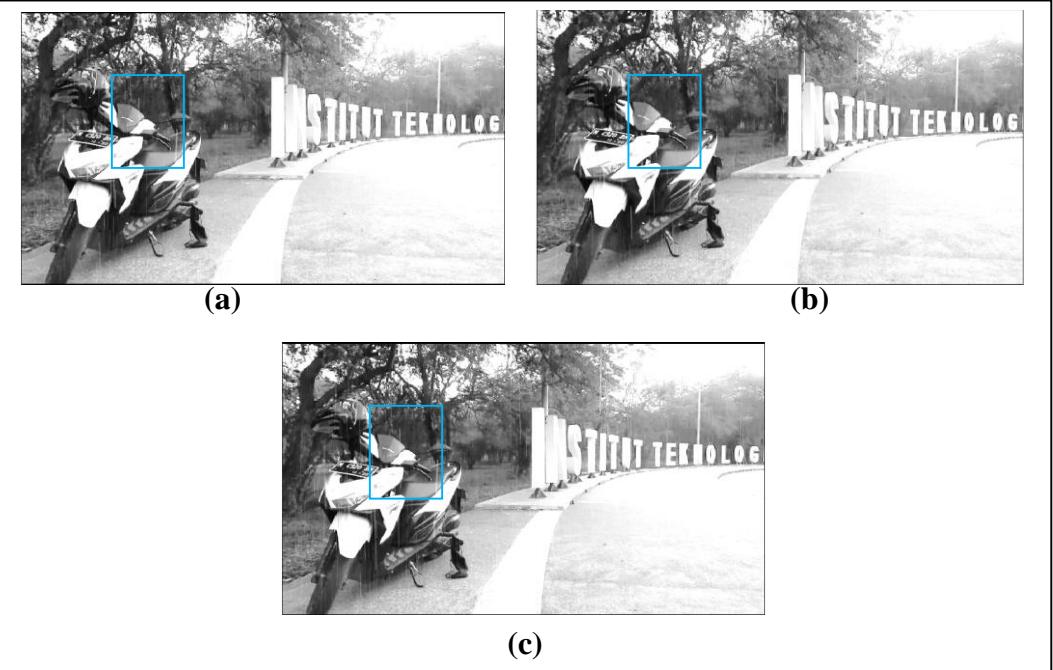
Setelah melalui proses dekomposisi, maka masing-masing koefisien aproksimasi dan koefisien detail di libatkan aturan fusi. Untuk melakukan wavelet *fusion* menggunakan fungsi sebagai berikut:

```
close all; clear all; clear all;

X1 = imread('rain_1.jpg');
X2 = imread('rain_2.jpg');

XFusi = wfusimg(X1, X2, 'db4', 2, 'max', 'max');

figure; imshow(uint8(XFusi))
```



Gambar 4.17: Representasi citra hasil fusi. (a) *frame* ke-3. (b) *frame* ke-4. (c) *frame* hasil fusi.

BAB 5

UJI COBA DAN ANALISIS HASIL

Pada bab ini menjelaskan tentang uji coba perancangan proses reduksi *noise* hujan yang sudah dibuat pada bab sebelumnya dan menjelaskan analisis hasil dari setiap uji coba video yang dilakukan. Uji coba ini dilakukan dengan menerapkan transformasi wavelet *Daubechies* db2, db4 db6 dan db8 dengan level dekomposisi 1- level 3. Sedangkan jenis *threshold* yang digunakan berupa *BayesShrink*, *VishuShrink*, dan *NormalShrink*. Tujuannya yaitu untuk mengetahui tingkat pengurangan *noise* hujan dari masing-masing filter *Daubechies* dan jenis *threshold* pada jenis *input* video dengan kondisi yang berbeda. Selanjutnya membandingkan nilai PSNR sebelum hujan dan setelah proses reduksi *noise* hujan untuk mengetahui kualitas hasil video yang sudah direduksi.

5.1 Data Uji Coba

Uji Coba program dilakukan pada video berekstensi.avi yang telah tersimpan dalam penyimpanan komputer yang diperoleh dari pengambilan di lingkungan sekitar ITS. Data yang digunakan berupa:

1. Hujan deras, sedang, dan rendah *background* diam objek bergerak
2. Hujan deras, sedang, dan rendah *background* diam tanpa objek
3. Hujan deras, sedang, dan rendah *background* diam objek diam
4. Hujan deras, sedang, dan rendah *background* bergerak objek bergerak

Parameter hujan deras, sedang, dan rendah berdasarkan parameter dari tool *after effect* adalah sebagai berikut :

- a. Hujan Deras : Tetesan hujan 20.000/detik, *size* hujan 0,00006 mm, *dept* 5000, *speed* 4000.
- b. Hujan Sedang : Tetesan hujan 10.000/detik, *size* hujan 0,00004 mm, *dept* 5000, *speed* 3000.
- c. Hujan Rendah : Tetesan hujan 5.000/detik, *size* hujan 0,00003 mm, *dept* 5000, *speed* 2000.

Tabel 5.1 : Dataset video uji coba reduksi noise

No.	Nama	Screenshot Video	Ket.
1.	Hujan1_deras.avi		<i>background</i> diam objek bergerak
2.	Hujan1_sedang.avi		<i>background</i> diam objek bergerak
3.	Hujan1_rendah.avi		<i>background</i> diam objek bergerak
4.	Hujan2_deras.avi		<i>background</i> diam tanpa objek
5.	Hujan2_sedang.avi		<i>background</i> diam tanpa objek
6.	Hujan2_rendah.avi		<i>background</i> diam tanpa objek

No.	Nama	Screenshot Video	Ket.
7.	Hujan3_deras.avi		<i>background</i> diam objek diam
8.	Hujan3_sedang.avi		<i>background</i> diam objek diam
9.	Hujan3_rendah.avi		<i>background</i> diam objek diam
10.	Hujan4_deras.avi		<i>background</i> bergerak objek bergerak
11.	Hujan4_sedang.avi		<i>background</i> bergerak objek bergerak
12.	Hujan4_rendah.avi		<i>background</i> bergerak objek bergerak

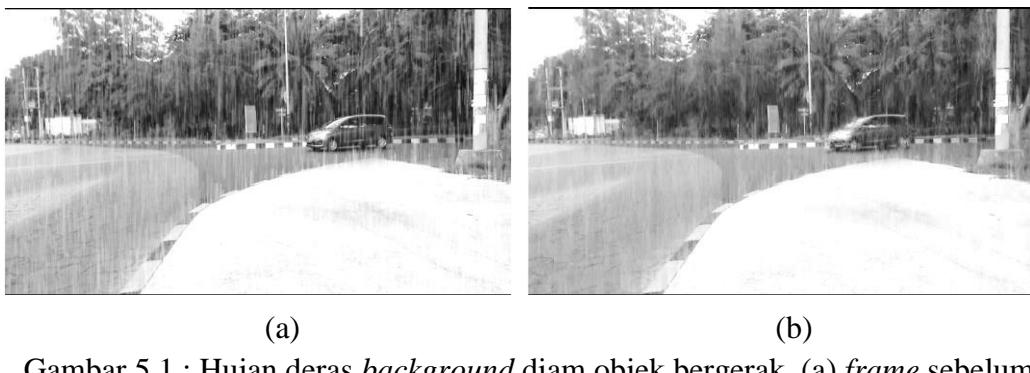
5.2 Uji Coba reduksi *noise* hujan pada video dan Analisis Hasil

Langkah pertama yang dilakukan untuk melakukan uji coba reduksi *noise* hujan pada video yaitu memilih video yang akan di uji coba. Selanjutnya memilih jenis wavelet *Daubechies* untuk mendekomposisi *frame*, tujuannya untuk memisahkan antara detail hujan dan detail komponen yang lain sesuai dengan tingkatan level dekomposisi. Langkah selanjutnya dilakukan proses wavelet *fusion* agar detail hujan semakin jelas. Pada penelitian ini, wavelet *fusion* menggunakan 2 *frame* saling berdekatan yaitu *frame 1* dan *frame 2*, *frame 2* dan *frame dst*, sampai *frame* ke-n. Langkah terakhir adalah memilih jenis *threshold* untuk mereduksi *noise* hujan dan hasilnya akan direkonstruksi dan dirender kembali ke dalam bentuk video yang hujannya sudah direduksi.

Pada uji coba ini, filter wavelet *Daubechies* yang digunakan adalah db2, db4, db6 dan db8 level 1 sampai level 3. Untuk jenis *threshold* menggunakan *BayesShrink*, *VishuShrink* dan *NormalShrink*.

5.2.1 Uji coba reduksi *noise* hujan deras *background* diam objek bergerak.

Hasil uji coba hujan deras pada file Hujan1_deras.avi ditunjukkan pada Gambar 5.1.



Gambar 5.1 : Hujan deras *background* diam objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan deras *background* diam objek bergerak, secara visual *noise* hujan di sekitar objek dan sekitar *background* kelihatan sedikit berkurang, akan tetapi objek semakin buram karena pada waktu proses dekomposisi, objek juga dianggap sebagai *noise* sehingga pada waktu proses *thresholding* objek juga ikut tereduksi. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan deras *background* diam objek bergerak ditunjukkan pada Tabel 5.2.

Tabel 5.2 Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan deras *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	25.7225	20.9154	0	0
		2	26.0726	25.1207	0.3501	4.2053
		3	26.4584	29.2422	0.3858	4.1215
2	Db4	1	25.7201	20.8227	0	0
		2	26.0641	25.8624	0.3440	5.0397
		3	26.4585	30.3195	0.3944	4.4571
3	Db6	1	25.7113	21.2053	0	0
		2	26.0586	26.4847	0.3473	5.2794
		3	26.4525	30.4326	0.3939	3.9479
4	Db8	1	25.7063	21.3697	0	0
		2	26.0500	27.4448	0.3437	6.0751
		3	26.4420	31.6642	0.3920	4.2194

Tabel 5.3 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan deras *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	25.7488	20.7328	0	0
		2	26.1006	26.5166	0.3518	5.7838
		3	26.489	30.165	0.3884	3.6484
2	Db4	1	25.7393	21.5283	0	0
		2	26.0844	26.8642	0.3451	5.3359
		3	26.4808	30.9326	0.3964	4.0684
3	Db6	1	25.7277	22.0781	0	0
		2	25.7488	27.997	0.0211	5.9189
		3	26.1006	32.5973	0.3518	4.6003
4	Db8	1	26.489	22.5274	0	0
		2	25.7393	28.8876	-0.7497	6.3602
		3	26.0844	33.3371	0.3451	4.4495

Tabel 5.4 : Tabel 5.4: Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan deras *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	25.7230	20.2979	0	0
		2	26.0730	25.7800	0.3500	5.4821
		3	26.4589	30.0182	0.3859	4.2382
2	Db4	1	25.7206	21.1885	0	0
		2	26.0646	26.2834	0.3440	5.0949
		3	26.4590	30.5763	0.3944	4.2929
3	Db6	1	25.7118	21.3902	0	0
		2	26.0591	27.5313	0.3473	6.1411
		3	26.4530	31.7146	0.3939	4.1833
4	Db8	1	25.7068	21.8075	0	0
		2	26.0504	28.5378	0.3436	6.7303
		3	26.4425	32.5754	0.3921	4.0376

Berdasarkan Tabel 5.2 sampai 5.4 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan deras pada *background* diam objek bergerak wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 26,4890 dB dan waktu komputasi 30,1650 detik.

5.2.2 Uji coba reduksi *noise* hujan sedang *background* diam objek bergerak

Hasil uji coba hujan sedang pada file Hujan1_sedang.avi ditunjukkan pada Gambar 5.2.



Gambar 5.2 : Hujan sedang *background* diam objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan sedang *background* diam objek bergerak, secara visual *noise* hujan di sekitar objek dan sekitar *background* kelihatan sedikit berkurang, akan tetapi objek semakin buram karena pada waktu proses dekomposisi objek

juga dianggap sebagai *noise* sehingga pada waktu proses *thresholding* objek juga ikut tereduksi. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan sedang *background* diam objek bergerak ditunjukkan pada Tabel 5.5.

Tabel 5.5 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi hujan sedang *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.5387	20.5311	0	0
		2	30.9481	24.676	0.4094	4.1449
		3	31.2747	28.6496	0.3266	3.9736
2	Db4	1	30.561	20.5061	0	0
		2	30.9669	24.7482	0.4059	4.2421
		3	31.2999	29.1902	0.3330	4.4420
3	Db6	1	30.558	20.9107	0	0
		2	30.9675	25.4845	0.4095	4.5738
		3	31.2955	29.4668	0.3280	3.9823
4	Db8	1	30.5557	20.5417	0	0
		2	30.9508	26.3415	0.3951	5.7998
		3	31.2738	30.2495	0.3230	3.9080

Tabel 5.6 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi hujan sedang *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.6379	19.8322	0	0
		2	31.0543	25.2142	0.4164	5.3820
		3	31.3888	28.98	0.3345	3.7658
2	Db4	1	30.6375	20.2883	0	0
		2	31.048	25.564	0.4105	5.2757
		3	31.3871	26.6049	0.3391	1.0409
3	Db6	1	30.6255	21.3775	0	0
		2	31.0391	26.3906	0.4136	5.0131
		3	31.3725	30.5506	0.3334	4.1600
4	Db8	1	30.6241	21.685	0	0
		2	31.0231	27.36138	0.3990	5.6764
		3	31.3515	31.6468	0.3284	4.2854

Tabel 5.7 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi hujan sedang *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.5404	19.4967	0	0
		2	30.9499	24.2007	0.4095	4.7040
		3	31.2763	28.2014	0.3264	4.0007
2	Db4	1	30.5628	19.9634	0	0
		2	30.9688	25.0249	0.4060	5.0615
		3	31.3018	29.1532	0.3330	4.1283
3	Db6	1	30.5599	20.2534	0	0
		2	30.9694	25.5674	0.4095	5.3140
		3	31.2974	29.7115	0.3280	4.1476
4	Db8	1	30.5575	20.6132	0	0
		2	30.9528	26.7118	0.3953	6.1048
		3	31.2756	30.4806	0.3228	3.7626

Berdasarkan Tabel 5.5 sampai 5.7 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan sedang pada *background* diam objek bergerak wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 31,3888 dB dan waktu komputasi 28,98 detik.

5.2.3 Uji coba reduksi noise hujan rendah *background* diam objek bergerak

Hasil uji coba hujan sedang pada file Hujan1_rendah.avi ditunjukkan pada Gambar 5.3.



Gambar 5.3 : Hujan rendah *background* diam objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan rendah *background* diam objek bergerak, secara visual *noise* hujan di sekitar objek dan sekitar *background* kelihatan sedikit berkurang, akan tetapi objek semakin buram karena pada waktu proses dekomposisi objek

juga dianggap sebagai *noise* sehingga pada waktu proses *thresholding* objek juga ikut tereduksi. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan sedang *background* diam objek bergerak ditunjukkan pada Tabel 5.8.

Tabel 5.8 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi hujan rendah *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.5387	20.5311	0	0
		2	30.9481	24.676	0.4094	4.1449
		3	31.2747	28.6496	0.3266	3.9736
2	Db4	1	30.561	20.5061	0	0
		2	30.9669	24.7482	0.4059	4.2421
		3	31.2999	29.1902	0.3330	4.4420
3	Db6	1	30.558	20.9107	0	0
		2	30.9675	25.4845	0.4095	4.5738
		3	31.2955	29.4668	0.3280	3.9823
4	Db8	1	30.5557	20.5417	0	0
		2	30.9508	26.3415	0.3951	5.7998
		3	31.2738	30.2495	0.3230	3.9080

Tabel 5.9 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi hujan rendah *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.6379	19.8322	0	0
		2	31.0543	25.2142	0.4164	5.3820
		3	31.3888	28.98	0.3345	3.7658
2	Db4	1	30.6375	20.2883	0	0
		2	31.048	25.564	0.4105	5.2757
		3	31.3871	26.6049	0.3391	1.0409
3	Db6	1	30.6255	21.3775	0	0
		2	31.0391	26.3906	0.4136	5.0131
		3	31.3725	30.5506	0.3334	4.1600
4	Db8	1	30.6241	21.685	0	0
		2	31.0231	27.36138	0.3990	5.6764
		3	31.3515	31.6468	0.3284	4.2854

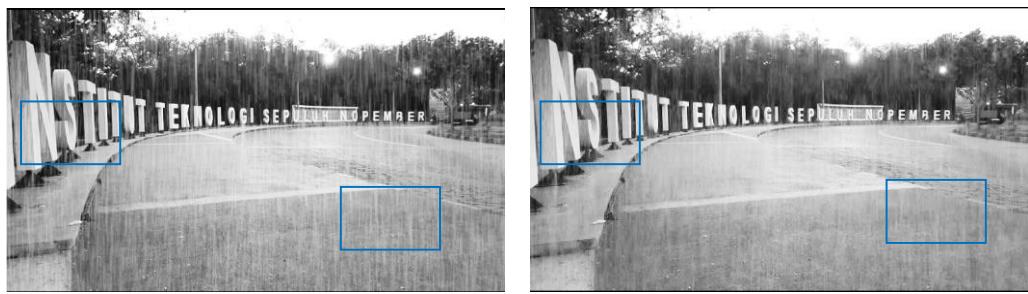
Tabel 5.10 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi hujan rendah *background* diam, objek bergerak.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.5404	19.4967	0	0
		2	30.9499	24.2007	0.4095	4.7040
		3	31.2763	28.2014	0.3264	4.0007
2	Db4	1	30.5628	19.9634	0	0
		2	30.9688	25.0249	0.4060	5.0615
		3	31.3018	29.1532	0.3330	4.1283
3	Db6	1	30.5599	20.2534	0	0
		2	30.9694	25.5674	0.4095	5.3140
		3	31.2974	29.715	0.3280	4.1476
4	Db8	1	30.5575	20.6132	0	0
		2	30.9528	26.718	0.3953	6.1048
		3	31.2756	30.4806	0.3228	3.7626

Berdasarkan Tabel 5.8 sampai 5.10 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan rendah pada *background* diam objek bergerak wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 31,3888 dB dan waktu komputasi 28,98 detik.

5.2.4 Uji coba reduksi noise hujan deras *background* diam tanpa objek .

Hasil uji coba hujan deras pada file Hujan2_deras.avi ditunjukkan pada Gambar 5.4.



Gambar 5.4 : Hujan deras *background* diam objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan deras *background* diam tanpa objek secara visual *noise* hujan pada *background* kelihatan berkurang, *background* nampak terlihat terang

karena proses dekomposisi, wavelet *fusion* dan proses *thresholding* tidak berpengaruh pada *background*. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya relatif besar tanpa adanya objek bergerak. Sehingga pada proses *thresholding*, tingkat pengurangan *noise* semakin banyak. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan deras *background* diam tanpa objek ditunjukkan pada Tabel 5.11.

Tabel 5.11 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi hujan deras *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	26.577	20.4132	0	0
		2	26.8909	25.4787	0.3139	5.0655
		3	27.2342	29.1311	0.3433	3.6524
2	Db4	1	26.5824	20.1324	0	0
		2	26.8876	24.3961	0.3052	4.2637
		3	27.2445	29.3066	0.3569	4.9105
3	Db6	1	26.5778	20.3404	0	0
		2	26.8915	25.9156	0.3137	5.5752
		3	27.2458	29.7946	0.3543	3.8790
4	Db8	1	26.5771	20.8971	0	0
		2	26.8836	26.2882	0.3065	5.3911
		3	27.2445	30.8473	0.3609	4.5591

Tabel 5.12 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi hujan deras *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	26.6456	20.316	0	0
		2	26.9644	25.4639	0.3188	5.1479
		3	27.3138	29.4475	0.3494	3.9836
2	Db4	1	26.6383	20.6488	0	0
		2	26.9472	26.7857	0.3089	6.1369
		3	27.3093	31.2796	0.3621	4.4939
3	Db6	1	26.6286	22.0219	0	0
		2	26.9458	27.4917	0.3172	5.4698
		3	27.3049	32.207	0.3591	4.7153
4	Db8	1	26.6249	22.6241	0	0
		2	26.9347	28.5265	0.3098	5.9024
		3	27.3001	32.9918	0.3654	4.4653

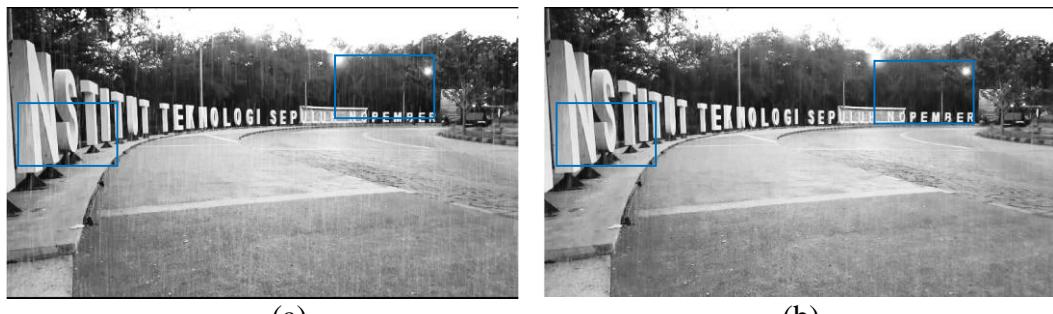
Tabel 5.13 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi hujan deras *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan waktu Waktu Komputasi (detik)
1	Db2	1	26.5772	20.5144	0	0
		2	26.8909	25.5118	0.3137	4.9974
		3	27.2341	29.6122	0.3432	4.1004
2	Db4	1	26.5826	20.7333	0	0
		2	26.8876	26.0053	0.305	5.2720
		3	27.2444	30.2605	0.3568	4.2552
3	Db6	1	26.578	21.1786	0	0
		2	26.8914	26.6148	0.3134	5.4362
		3	27.2457	30.941	0.3543	4.3262
4	Db8	1	26.5773	21.825	0	0
		2	26.8836	27.5685	0.3063	5.7435
		3	27.2444	31.9266	0.3608	4.3581

Berdasarkan Tabel 5.11 sampai 5.13 serta grafik perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan deras pada *background* diam tanpa objek, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 27,3138 dB dan waktu komputasi 29,4475 detik.

5.2.5 Uji coba reduksi *noise* hujan sedang *background* diam tanpa objek .

Hasil uji coba hujan sedang pada file Hujan2_sedang.avi ditunjukkan pada Gambar 5.5.



Gambar 5.5 : Hujan sedang *background* diam objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan sedang *background* diam tanpa objek secara visual *noise* hujan pada *background* kelihatan berkurang, *background* nampak terlihat terang

karena proses dekomposisi, wavelet *fusion* dan proses *thresholding* tidak berpengaruh pada *background*. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya relatif besar tanpa adanya objek bergerak. Pada proses *thresholding*, tingkat pengurangan *noise* semakin banyak disebabkan kerapatan dan *size* hujan yang semakin berkurang dibandingkan dengan hujan deras. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan sedang *background* diam tanpa objek ditunjukkan pada Tabel 5.14.

Tabel 5.14 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi hujan sedang *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	33.4676	20.7934	0	0
		2	33.8881	25.3365	0.4205	4.5431
		3	34.2074	29.5	0.3193	4.1635
2	Db4	1	33.5448	20.8436	0	0
		2	33.9651	25.8474	0.4203	5.0038
		3	34.2932	21.5942	0.3281	-4.2532
3	Db6	1	33.555	21.9169	0	0
		2	33.9879	27.7451	0.4329	5.8282
		3	34.3053	31.7334	0.3174	3.9883
4	Db8	1	33.5663	22.6994	0	0
		2	33.9873	28.567	0.421	5.8676
		3	34.3091	32.7227	0.3218	4.1557

Tabel 5.15 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi hujan sedang *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi
1	Db2	1	33.4685	21.3452	0	0
		2	33.8884	26.7394	0.4199	5.3942
		3	34.2074	31.2456	0.319	4.5062
2	Db4	1	33.5458	21.7516	0	0
		2	33.9651	27.4903	0.4193	5.7387
		3	34.293	38.3172	0.3279	10.8269
3	Db6	1	33.5559	22.6417	0	0
		2	33.9877	32.2966	0.4318	9.6549
		3	34.3048	35.1377	0.3171	2.8411
4	Db8	1	33.5672	24.8863	0	0
		2	33.9871	31.1325	0.4199	6.2462
		3	34.3084	36.1364	0.3213	5.0039

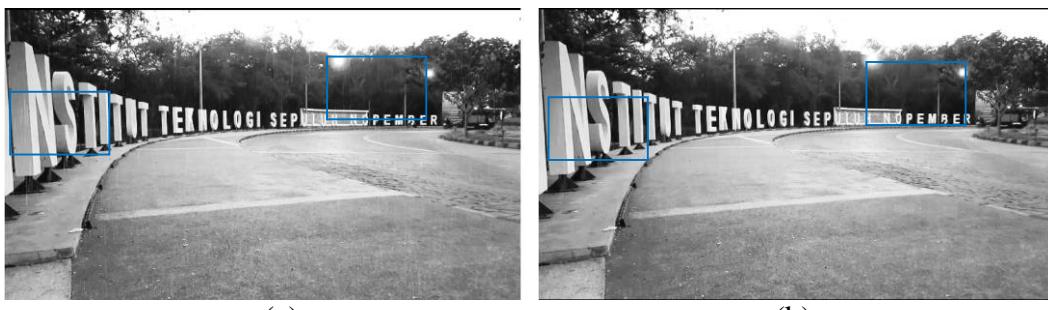
Tabel 5.16 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi hujan sedang *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	26.5772	22.3839	0	0
		2	26.8909	27.8482	0.3137	5.4643
		3	27.2341	32.1961	0.3432	4.3479
2	Db4	1	26.5826	22.8532	0	0
		2	26.8876	33.8283	0.305	10.9751
		3	27.2444	34.2344	0.3568	0.4061
3	Db6	1	26.578	23.6718	0	0
		2	26.8914	29.4074	0.3134	5.7356
		3	27.2457	35.3567	0.3543	5.9493
4	Db8	1	26.5773	24.8472	0	0
		2	26.8836	30.0528	0.3063	5.2056
		3	27.2444	34.8306	0.3608	4.7778

Berdasarkan Tabel 5.13 sampai Tabel 5.15 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan sedang pada *background* diam tanpa objek, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db4 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 34,6985 dB dan waktu komputasi 38,3172 detik.

5.2.6 Uji coba reduksi *noise* hujan rendah *background* diam tanpa objek .

Hasil uji coba hujan rendah pada file Hujan2_rendah.avi ditunjukkan pada Gambar 5.6.



Gambar 5.6 : Hujan rendah *background* diam objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan rendah *background* diam tanpa objek secara visual *noise* hujan pada *background* kelihatan berkurang, *background* nampak terlihat terang karena proses dekomposisi, wavelet *fusion* dan proses *thresholding* tidak

berpengaruh pada *background*. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya relatif besar tanpa adanya objek bergerak. Pada proses *thresholding*, tingkat pengurangan *noise* semakin banyak disebabkan kerapatan dan *size* hujan yang semakin berkurang dibandingkan dengan hujan sedang. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan rendah *background* diam tanpa objek ditunjukkan pada Tabel 5.17.

Tabel 5.17 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan rendah *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	37.2473	23.2006	0	0
		2	37.4536	28.0421	0.2063	4.8415
		3	37.5681	32.1511	0.1145	4.1090
2	Db4	1	37.4411	22.4616	0	0
		2	37.6462	28.468	0.2051	6.0064
		3	37.7694	33.4183	0.1232	4.9503
3	Db6	1	37.4968	23.0995	0	0
		2	37.7169	29.2884	0.2200	6.1889
		3	37.8314	33.8687	0.1145	4.5803
4	Db8	1	37.5311	23.539	0	0
		2	37.74	30.3638	0.2089	6.8248
		3	37.8589	35.139	0.1189	4.7752

Tabel 5.18 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan rendah *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	38.3919	32.2255	0	0
		2	38.6323	28.5325	0.2404	-3.6930
		3	38.777	33.2408	0.1447	4.7083
2	Db4	1	38.4144	23.578	0	0
		2	38.6425	29.497	0.2281	5.9190
		3	38.7916	34.5306	0.1491	5.0336
3	Db6	1	38.4031	24.241	0	0
		2	38.647	29.9208	0.2439	5.6798
		3	38.7842	35.0702	0.1372	5.1494
4	Db8	1	38.3982	24.8161	0	0
		2	38.6303	31.4699	0.2321	6.6538
		3	38.7728	36.3757	0.1425	4.9058

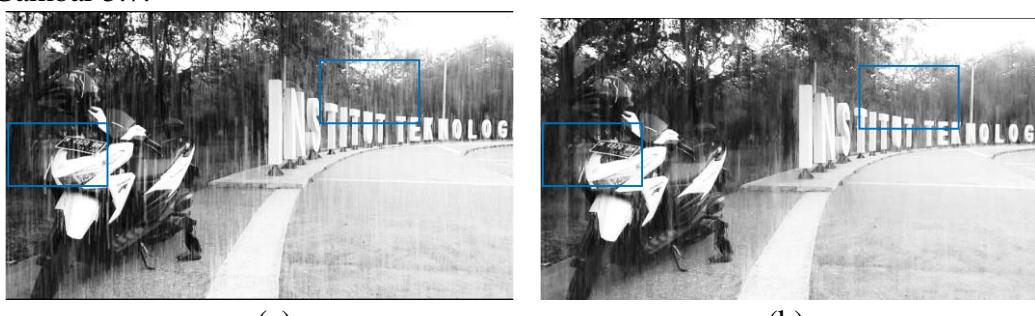
Tabel 5.19 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan rendah *background* diam tanpa objek.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	37.2492	22.5678	0	0
		2	37.4545	28.3718	0.2053	5.8040
		3	37.5686	32.7222	0.1141	4.3504
2	Db4	1	37.4428	23.1533	0	0
		2	37.6464	28.6099	0.2036	5.4566
		3	37.769	33.8449	0.1226	5.2350
3	Db6	1	37.5	23.5264	0	0
		2	37.717	29.8418	0.217	6.3154
		3	37.8309	34.6459	0.1139	4.8041
4	Db8	1	37.5325	24.0517	0	0
		2	37.7397	31.0151	0.2072	6.9634
		3	37.858	35.6249	0.1183	4.6098

Berdasarkan Tabel 5.17 dan 5.19 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan rendah pada *background* diam tanpa objek, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db4 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 38,7196 dB dan waktu komputasi 34,5306 detik.

5.2.7 Uji coba reduksi *noise* hujan deras *background* diam objek diam .

Hasil uji coba hujan deras pada file Hujan3_deras.avi ditunjukkan pada Gambar 5.7.



Gambar 5.7 : Hujan deras *background* diam objek diam. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan deras *background* diam objek diam secara visual *noise* hujan pada *background* dan objek kelihatan berkurang, *background* dan plat nomor pada motor tidak terlalu nampak terang karena masih tertutup oleh *noise*

hujan. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya relatif besar. Pada proses *thresholding*, tingkat pengurangan *noise* tidak terlalu banyak disebabkan kerapatan dan *size* hujan yang besar. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan deras *background* diam objek diam ditunjukkan pada Tabel 5.20.

Tabel 5.20 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan deras *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	25.1126	23.3616	0	0
		2	25.4479	28.4391	0.3353	5.0775
		3	25.8071	30.5821	0.3592	2.1430
2	Db4	1	25.1293	21.9814	0	0
		2	25.4538	26.9739	0.3245	4.9925
		3	25.8269	31.6023	0.3731	4.6284
3	Db6	1	25.1264	21.9115	0	0
		2	25.4609	27.6227	0.3345	5.7112
		3	25.8281	32.731	0.3672	5.1083
4	Db8	1	25.1249	22.4483	0	0
		2	25.4547	28.8408	0.3298	6.3925
		3	25.8224	33.5773	0.3677	4.7365

Tabel 5.21 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan deras *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	25.2593	21.7744	0	0
		2	25.6062	27.5056	0.3469	5.7312
		3	25.9791	31.6267	0.3729	4.1211
2	Db4	1	25.2516	27.9393	0	0
		2	25.5851	28.3097	0.3335	0.3704
		3	25.9701	32.1867	0.385	3.8770
3	Db6	1	25.2383	22.9998	0	0
		2	25.5822	29.4559	0.3439	6.4561
		3	25.9602	33.7595	0.378	4.3036
4	Db8	1	25.2341	23.5483	0	0
		2	25.5735	29.7177	0.3394	6.1694
		3	25.9521	34.5428	0.3786	4.8251

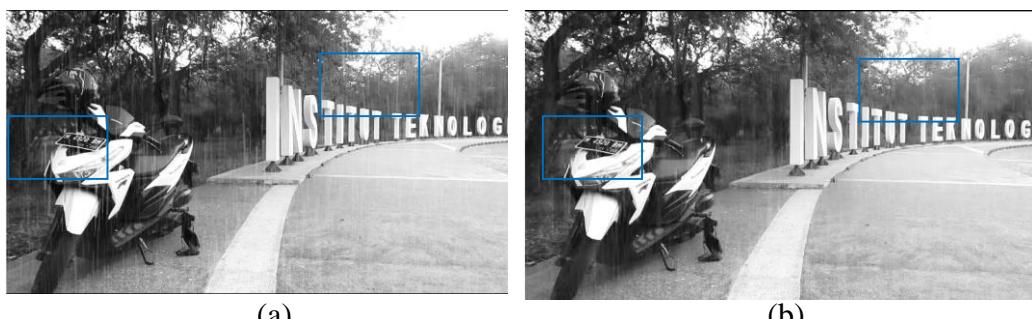
Tabel 5.22 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan deras *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	25.1126	21.3497	0	0
		2	25.4477	26.6758	0.3351	5.3261
		3	25.8066	30.8846	0.3589	4.2088
2	Db4	1	25.1293	21.8132	0	0
		2	25.4534	27.1011	0.3241	5.2879
		3	25.8263	31.5034	0.3729	4.4023
3	Db6	1	25.1264	22.0812	0	0
		2	25.4605	28.1642	0.3341	6.0830
		3	25.8275	32.5232	0.367	4.3590
4	Db8	1	25.1249	22.5917	0	0
		2	25.4543	28.9158	0.3294	6.3241
		3	25.8217	33.5417	0.3674	4.6259

Berdasarkan Tabel 5.20 dan 5.22 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan deras pada *background* diam objek diam, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 25,9791 dB dan waktu komputasi 31,6267 detik.

5.2.8 Uji coba reduksi *noise* hujan sedang *background* diam objek diam .

Hasil uji coba hujan sedang pada file Hujan3_sedang.avi ditunjukkan pada Gambar 5.8.



Gambar 5.8: Hujan sedang *background* diam objek diam. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan sedang *background* diam objek diam secara visual *noise* hujan pada *background* dan objek kelihatan berkurang, *background* dan Plat

nomor motor tidak terlalu nampak terang karena masih tertutup oleh *noise* hujan. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya relatif besar. Pada proses *thresholding*, tingkat pengurangan *noise* semakin besar dibandingkan dengan hujan deras disebabkan kerapatan dan *size* hujan yang semakin kecil. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan sedang *background* diam objek diam ditunjukkan pada Tabel 5.23.

Tabel 5.23 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan sedang *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.5508	21.4246	0	0
		2	31.0197	26.5174	0.4689	5.0928
		3	31.3753	30.8372	0.3556	4.3198
2	Db4	1	30.6406	21.634	0	0
		2	31.1038	27.1085	0.4632	5.4745
		3	31.4754	33.7299	0.3716	6.6214
3	Db6	1	30.667	23.2527	0	0
		2	31.14	27.9771	0.473	4.7244
		3	31.4953	32.332	0.3553	4.3549
4	Db8	1	30.6744	22.5959	0	0
		2	31.1342	28.6292	0.4598	6.0333
		3	31.4869	33.3969	0.3527	4.7677

Tabel 5.24 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan sedang *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	31.1415	21.9082	0	0
		2	31.6746	27.5886	0.5331	5.6804
		3	32.0888	32.039	0.4142	4.4504
2	Db4	1	31.142	23.2612	0	0
		2	31.6574	28.7383	0.5154	5.4771
		3	32.0801	32.1355	0.4227	3.3972
3	Db6	1	31.13	22.8609	0	0
		2	31.6539	28.6417	0.5239	5.7808
		3	32.0552	33.1531	0.4013	4.5114
4	Db8	1	31.1264	23.378	0	0
		2	31.6374	29.4811	0.511	6.1031
		3	32.0354	33.9042	0.398	4.4231

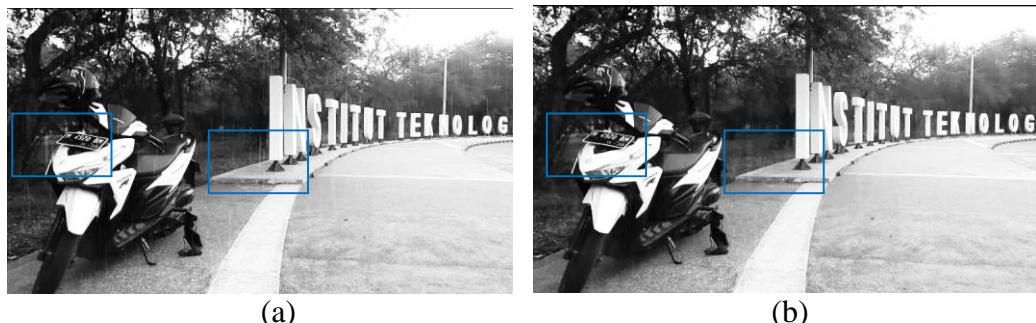
Tabel 5.25 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan sedang *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	30.5515	21.11	0	0
		2	31.0192	26.6279	0.4677	5.5179
		3	31.3743	30.9678	0.3551	4.3399
2	Db4	1	30.641	21.6687	0	0
		2	31.103	27.1454	0.462	5.4767
		3	31.4714	31.4622	0.3684	4.3168
3	Db6	1	30.6674	22.1931	0	0
		2	31.139	27.6346	0.4716	5.4415
		3	31.4937	32.4054	0.3547	4.7708
4	Db8	1	30.6748	22.4686	0	0
		2	31.1331	28.5022	0.4583	6.0336
		3	31.4851	33.0122	0.352	4.5100

Berdasarkan Tabel 5.23 sampai 5.25 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan sedang pada *background* diam objek diam, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 32,0888 dB dan waktu komputasi 32,0390 detik.

5.2.9 Uji coba reduksi *noise* hujan rendah *background* diam objek diam .

Hasil uji coba hujan rendah pada file Hujan3_rendah.avi ditunjukkan pada Gambar 5.9.



Gambar 5.9: Hujan rendah *background* diam objek diam. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan rendah *background* diam objek diam secara visual *noise* hujan pada *background* dan objek kelihatan berkurang, *background* dan Plat nomor motor sudah semakin nampak terang dibandingkan dengan hujan sedang.

Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya relatif besar. Pada proses *thresholding*, tingkat pengurangan *noise* semakin besar dibandingkan dengan hujan sedang disebabkan kerapatan dan *size* hujan yang kecil. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan rendah *background* diam objek diam ditunjukkan pada Tabel 5.26.

Tabel 5.26 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan rendah *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	34.6276	20.9363	0	0
		2	34.9163	26.1768	0.2887	5.2405
		3	35.0843	30.3465	0.168	4.1697
2	Db4	1	34.8694	21.5055	0	0
		2	35.1616	27.0358	0.2922	5.5303
		3	35.3435	31.2619	0.1819	4.2261
3	Db6	1	34.9653	22.0111	0	0
		2	35.2644	27.4357	0.2991	5.4246
		3	35.4324	31.8662	0.168	4.4305
4	Db8	1	34.9921	22.521	0	0
		2	35.2693	29.1006	0.2772	6.5796
		3	36.973	35.2812	1.7037	6.1806

Tabel 5.27: Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan rendah *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	36.394	22.2715	0	0
		2	36.7747	26.921	0.2887	5.2405
		3	37.021	30.8997	0.168	4.1697
2	Db4	1	36.3968	21.9572	0	0
		2	36.7598	27.733	0.2922	5.5303
		3	37.0119	32.1705	0.1819	4.2261
3	Db6	1	36.3949	22.7157	0	0
		2	36.7659	28.4537	0.2991	5.4246
		3	36.9956	32.7798	0.168	4.4305
4	Db8	1	36.3944	23.8036	0	0
		2	36.7435	29.8474	0.2772	6.5796
		3	38.7728	34.271	1.7037	6.1806

Tabel 5.28 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan rendah *background* diam objek diam.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR	kenaikan Waktu Komputasi (detik)
1	Db2	1	34.6291	21.6584	0	0
		2	34.9158	27.1063	0.2867	5.4479
		3	35.0832	31.681	0.1674	4.5747
2	Db4	1	34.8701	22.4611	0	0
		2	35.16	27.9746	0.2899	5.5135
		3	35.3409	32.1698	0.1809	4.1952
3	Db6	1	34.966	22.802	0	0
		2	35.2623	28.8301	0.2963	6.0281
		3	35.4291	33.3095	0.1668	4.4794
4	Db8	1	34.9927	23.4576	0	0
		2	35.2668	29.367	0.2741	5.9094
		3	35.4329	33.9302	0.1661	4.5632

Berdasarkan Tabel 5.26 dan 5.28 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan rendah pada *background* diam objek diam, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db2 level 3, menggunakan *threshold VisuShrink* dengan nilai PSNR sebesar 37,021 dB dan waktu komputasi 30,8997 detik.

5.2.10 Uji coba reduksi *noise* hujan deras *background* bergerak objek bergerak .

Hasil uji coba hujan deras pada file Hujan4_deras.avi ditunjukkan pada Gambar 5.10.



Gambar 5.10: Hujan deras *background* bergerak objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan deras *background* bergerak objek bergerak secara visual *noise* hujan pada *background* dan objek tidak terlihat berkurang. Plat nomor

motor semakin nampak buram. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya kecil disebabkan *background* dan objek bergerak sehingga terdapat perbedaan antara 2 *frame* yang digabungkan. Pada proses *thresholding*, tingkat pengurangan *noise* kecil karena kerapatan dan *size* hujan yang besar sehingga *noise* hujan tidak berkurang dan hanya terlihat buram dan menutupi objek di belakangnya. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan deras *background* bergerak objek bergerak ditunjukkan pada Tabel 5.29.

Tabel 5.29 : Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan deras *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>BayesShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	21.2755	22.6309	0	0
		2	21.4142	27.3301	0.1387	4.6992
		3	21.58	32.1429	0.1658	4.8128
2	Db4	1	22.2707	22.6411	0	0
		2	21.4071	28.5716	-0.8636	5.9305
		3	21.5888	32.7849	0.1817	4.2133
3	Db6	1	21.2649	23.2959	0	0
		2	21.4039	28.4294	0.139	5.1335
		3	21.5918	33.319	0.1879	4.8896
4	Db8	1	21.2595	23.2578	0	0
		2	21.3968	29.7325	0.1373	6.4747
		3	21.5937	34.3801	0.1969	4.6476

Tabel 5.30 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan deras *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>VishuShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	21.2745	22.7749	0	0
		2	21.413	29.3785	0.1385	6.6036
		3	21.5786	34.7329	0.1656	5.3544
2	Db4	1	21.2696	23.1564	0	0
		2	21.4058	28.7301	0.1362	5.5737
		3	21.5873	33.6023	0.1815	4.8722
3	Db6	1	21.2638	23.456	0	0
		2	21.4026	29.6576	0.1388	6.2016
		3	21.5903	34.0260	0.1877	4.3684
4	Db8	1	21.2585	24.1197	0	0
		2	21.3955	30.5572	0.137	6.4375
		3	21.5924	36.3381	0.1969	5.7809

Tabel 5.31 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan deras *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>NormalShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	21.2773	21.5393	0	0
		2	21.4166	26.8956	0.1393	5.3563
		3	21.5825	31.6836	0.1659	4.7880
2	Db4	1	21.2724	21.9349	0	0
		2	21.4095	27.6446	0.1371	5.7097
		3	21.5913	32.2854	0.1818	4.6408
3	Db6	1	21.2665	22.5996	0	0
		2	21.406	28.7436	0.1395	6.1440
		3	21.5941	33.3917	0.1881	4.6481
4	Db8	1	21.2612	23.2422	0	0
		2	21.3991	29.4817	0.1379	6.2395
		3	21.5962	33.9725	0.1971	4.4908

Berdasarkan Tabel 5.29 sampai 5.31 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan deras pada *background* bergerak objek bergerak, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db4 level 1, menggunakan *threshold BayesShrink* dengan nilai PSNR sebesar 22,2707 dB dan waktu komputasi 22,6411 detik.

5.2.11 Uji coba reduksi *noise* hujan sedang *background* bergerak objek bergerak .

Hasil uji coba hujan sedang pada file Hujan4_sedang.avi ditunjukkan pada Gambar 5.11.



Gambar 5.11: Hujan sedang *background* bergerak objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan sedang *background* bergerak objek bergerak secara visual *noise* hujan pada *background* dan objek sedikit berkurang. Akan tetapi, plat nomor motor dan *background* semakin nampak buram. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya kecil disebabkan *background* dan objek bergerak sehingga terdapat perbedaan antara 2 *frame* yang digabungkan. Pada proses *thresholding*, tingkat pengurangan *noise* semakin bertambah dibandingkan dengan *noise* hujan deras karena kerapatan dan *size* hujan yang berkurang sehingga *noise* hujan semakin berkurang. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan sedang *background* bergerak objek bergerak ditunjukkan pada Tabel 5.32.

Tabel 5.32 : Perbandingan nilai PSNR *BayeshShrink* dan waktu komputasi data hujan sedang *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>BayesShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	22.4112	22.6448	0	0
		2	22.4949	27.7641	0.0837	5.1193
		3	22.5662	32.29	0.0713	4.5259
2	Db4	1	22.4087	22.496	0	0
		2	22.491	27.687	0.0823	5.1910
		3	22.5781	33.1853	0.0871	5.4983
3	Db6	1	22.4077	22.4889	0	0
		2	22.4924	28.3181	0.0847	5.8292
		3	22.5814	34.3511	0.089	6.0330
4	Db8	1	22.4041	22.4956	0	0
		2	22.4845	29.544	0.0804	7.0484
		3	22.5818	34.4394	0.0973	4.8954

Tabel 5.33 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan sedang *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>VishuShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	22.4093	22.7649	0	0
		2	22.4927	27.8292	0.0834	5.0643
		3	22.5637	32.5166	0.071	4.6874
2	Db4	1	22.4067	22.8644	0	0
		2	22.4887	29.1436	0.082	6.2792
		3	22.5757	33.9662	0.087	4.8226
3	Db6	1	22.4059	23.6731	0	0
		2	22.4902	29.1198	0.0843	5.4467
		3	22.579	34.2277	0.0888	5.1079
4	Db8	1	22.4025	24.1377	0	0
		2	22.4824	30.6391	0.0799	6.5014
		3	22.5795	35.5231	0.0971	4.8840

Tabel 5.34 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan sedang *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>NormalShrink</i>	kenaikan waktu komputasi (detik)
1	Db2	1	22.4143	21.6415	0	0
		2	22.4992	27.0223	0.0849	5.3808
		3	22.5706	31.9846	0.0714	4.9623
2	Db4	1	22.4117	21.9883	0	0
		2	22.4951	28.0404	0.0834	6.0521
		3	22.5825	32.5481	0.0874	4.5077
3	Db6	1	22.4105	22.7161	0	0
		2	22.4962	28.2479	0.0857	5.5318
		3	22.5855	33.1035	0.0893	4.8556
4	Db8	1	22.407	23.1737	0	0
		2	22.4885	29.3425	0.0815	6.1688
		3	22.586	34.3482	0.0975	5.0057

Berdasarkan Tabel 5.32 sampai 5.34 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan sedang pada *background* bergerak objek bergerak, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db8 level 3, menggunakan *threshold NormalShrink* dengan nilai PSNR sebesar 22,5860 dB dan waktu komputasi 34,3482 detik.

5.2.12 Uji coba reduksi *noise* hujan rendah *background* bergerak objek bergerak .

Hasil uji coba hujan rendah pada file Hujan4_rendah.avi ditunjukkan pada Gambar 5.12.



Gambar 5.12: Hujan rendah *background* bergerak objek bergerak. (a) *frame* sebelum reduksi. (b) *frame* sesudah reduksi.

Untuk data hujan rendah *background* bergerak objek bergerak secara visual *noise* hujan pada *background* dan objek sudah hilang. Akan tetapi, plat

nomor motor dan *background* semakin nampak buram. Pada waktu proses fusi, 2 *frame* yang digabungkan tingkat kesamaan *noise* nya kecil disebabkan *background* dan objek bergerak sehingga terdapat perbedaan antara 2 *frame* yang digabungkan. Pada proses *thresholding*, tingkat pengurangan *noise* besar dibandingkan dengan *noise* hujan sedang karena kerapatan dan *size* hujan yang relatif kecil sehingga *noise* hujan semakin berkurang. Hasil perbandingan PSNR yang dilakukan sebanyak 60 kali, untuk data hujan rendah *background* bergerak objek bergerak ditunjukkan pada Tabel 5.35.

Tabel 5.35: Perbandingan nilai PSNR *BayesShrink* dan waktu komputasi data hujan rendah *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>BayesShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>BayesShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	22.7757	21.8069	0	0
		2	22.82	27.6565	0.0443	5.8496
		3	22.8376	31.9577	0.0176	4.3012
2	Db4	1	22.7754	21.7628	0	0
		2	22.8193	27.6865	0.0439	5.9237
		3	22.8504	32.1792	0.0311	4.4927
3	Db6	1	22.7771	21.5127	0	0
		2	22.8219	28.4564	0.0448	6.9437
		3	22.8553	33.0493	0.0334	4.5929
4	Db8	1	22.775	21.5127	0	0
		2	22.8169	29.2555	0.0419	7.7428
		3	22.8556	34.1421	0.0387	4.8866

Tabel 5.36 : Perbandingan nilai PSNR *VishuShrink* dan waktu komputasi data hujan rendah *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>VishuShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>VishuShrink</i> (dB)	kenaikan waktu komputasi (detik)
1	Db2	1	22.7728	22.3072	0	0
		2	22.8167	27.8021	0.0439	5.4949
		3	22.8342	31.5401	0.0175	3.7380
2	Db4	1	22.7728	22.9184	0	0
		2	22.8162	28.4838	0.0434	5.5654
		3	22.847	33.2783	0.0308	4.7945
3	Db6	1	22.7746	23.2273	0	0
		2	22.8189	29.3476	0.0443	6.1203
		3	22.852	33.8525	0.0331	4.5049
4	Db8	1	22.7725	25.0034	0	0
		2	22.8139	32.7236	0.0414	7.7202
		3	22.8534	34.7724	0.0395	2.0488

Tabel 5.37 : Perbandingan nilai PSNR *NormalShrink* dan waktu komputasi data hujan rendah *background* bergerak objek bergerak.

No.	Wavelet	Level	PSNR <i>NormalShrink</i> (dB)	Waktu Komputasi (detik)	kenaikan PSNR <i>NormalShrink</i>	kenaikan waktu komputasi (detik)
1	Db2	1	22.7794	21.9874	0	0
		2	22.8252	26.9769	0.0458	4.9895
		3	22.843	31.4908	0.0178	4.5139
2	Db4	1	22.7789	22.0248	0	0
		2	22.8242	27.7845	0.0453	5.7597
		3	22.8555	32.3496	0.0313	4.5651
3	Db6	1	22.7805	22.7317	0	0
		2	22.8266	28.3166	0.0461	5.5849
		3	22.8601	33.1138	0.0335	4.7972
4	Db8	1	22.7783	25.1171	0	0
		2	22.8218	29.0912	0.0435	3.9741
		3	22.8617	33.9489	0.0399	4.8577

Berdasarkan Tabel 5.35 dan 5.37 perbandingan nilai PSNR dan waktu komputasi dari masing-masing level dekomposisi, untuk data hujan rendah pada *background* bergerak objek bergerak, wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies* db8 level 3, menggunakan *threshold NormalShrink* dengan nilai PSNR sebesar 22,8617 dB dan waktu komputasi 33,9489 detik.

BAB 6

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil penelitian yang sudah dilakukan. Selain itu, pada bab ini juga terdapat saran jika penelitian ini akan dikembangkan.

6.1 Kesimpulan

Berdasarkan hasil analisis terhadap uji coba yang sudah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Untuk jenis data hujan deras, berdasarkan uji coba perbandingan nilai PSNR yang dilakukan sebanyak 60 kali, pada video dengan kecepatan 25 fps diperoleh hasil bahwa wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies db2* dekomposisi level 3 dengan jenis *threshold VishuShrink* terhadap video dengan *background* diam tanpa objek. Perbandingan nilai PSNR yang diperoleh adalah 27,3138 dB dengan waktu komputasi 29,4475 detik.
2. Untuk jenis data hujan sedang, berdasarkan uji coba perbandingan nilai PSNR yang dilakukan sebanyak 60 kali, pada video dengan kecepatan 25 fps diperoleh hasil bahwa wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies db4* dekomposisi level 3 dengan jenis *threshold VishuShrink* terhadap video dengan *background* diam tanpa objek. Perbandingan nilai PSNR yang diperoleh adalah 34,6985 dB dengan waktu komputasi 38,3172 detik.
- 3.Untuk jenis data hujan rendah berdasarkan uji coba perbandingan nilai PSNR yang dilakukan sebanyak 60 kali, pada video dengan kecepatan 25 fps diperoleh hasil bahwa wavelet *Daubechies* yang baik digunakan adalah wavelet *Daubechies db4* dekomposisi level 3 dengan jenis *threshold VishuShrink* terhadap video dengan *background* diam tanpa objek. Perbandingan nilai PSNR yang diperoleh adalah 38,7916 dB dengan waktu komputasi 34,5306 detik.
4. Faktor-faktor yang mempengaruhi proses reduksi *noise* hujan adalah *frame* video yang digunakan pada saat proses fusi. Jika pada video dengan objek bergerak dan *background* diam yang digunakan adalah *frame* ke-*i* dan *frame* ke-

($i+1$), maka reduksi *noise* hujan relatif kecil, tetapi tingkat keburaman objek dan *background* juga kecil. Akan tetapi, jika *frame* yang digunakan adalah *frame* ke- i dan *frame* ke- $(i+2)$, *frame* ke- i dan *frame* ke- $(i+3)$ dan seterusnya, maka reduksi *noise* hujan relatif besar dengan tingkat keburaman objek dan *background* juga besar. Sedangkan untuk video hujan tanpa objek, semakin besar jarak antara *frame* awal dengan *frame* kedua dalam proses fusi, maka reduksi *noise* hujan semakin besar.

6.2 Saran

Saran yang dapat disampaikan terkait dengan hasil penelitian yang dilakukan, antara lain sebagai berikut:

1. Diperlukan suatu penelitian lebih lanjut tentang reduksi *noise* hujan pada video *grayscale* dengan menggunakan keluarga wavelet yang lain selain wavelet *Daubechies*.
2. Perlu adanya penelitian lebih lanjut untuk reduksi *noise* hujan pada video RGB.
3. Video masukan yang digunakan pada penelitian ini berupa video yang diperoleh pada waktu siang hari sehingga perlu dilakukan penelitian lebih lanjut untuk video pada waktu malam hari.
4. Pada penelitian ini hanya mereduksi *noise* hujan tanpa memperbaiki *background* atau objek yang ada di belakangnya, sehingga perlu penelitian kembali untuk memperjelas objek atau *background* di belakang *noise* hujan.

DAFTAR PUSTAKA

- [1] Garg and S.K Nayar. (2004), “Detection and Removal of Rain from Videos”, Proc. CVR, Vol.1, hal 528-535.
- [2] Al Bovik, (2000), *Handbook of Image Video Processing* , Academic Press Publisher, Sand Diego.
- [3] Xiapeng Zhang, (2006), “Rain Removal in Video by Combining Temporal and Chromatic Propertis”, IEEE Transaction On Pattern Analysisand Machine Intelligence hal.461-464.
- [4] Gupta Nikil, (2011), “Video Modelling and Noise Reduction in The Wavelet Domain”, A Thesis In the Departement of Electrical and Computer Engineering.
- [5] Chen Zhen. (2013), “ A New Algorithm of Rain (Snow) Removal in Video ”, Journal of Multimedia, Vol.8, No.2, hal 168-174.
- [6] Solomon, C., Breckon, T., (2013). *Fundamental of Digital Image Processing*, Jhon Wiley and Sons, Ltd., Chichester.
- [7] I. Daubechies , (1992), *Ten Lectures on Wavelets*, SIAM, Philadelphia.
- [8] Pajares.G dan Manuel, J., (2004) “ A Wavelet-based image Fusion Tutorial ”, The Journal of the Pattern Recognition Society, hal. 1855-1872.
- [9] Zhang, Z., S. Blum, R., (1999) “A Categorization of Multiscale- Decomposition- Based Image Fusion Schemes with a Performance Study for a Digital Camera Application”, IEEE Vol.87 No.8 hal. 1315 – 1326.
- [10] Sutoyo. T., Mulyanto, E., Suhartono, V., Nurhayati, O.D dan Wijanarto, (2009), *Teori Pengolahan Citra Digital*, Penerbit ANDI, Yogyakarta.
- [11] Donoho, D.L. and Jhonstone, I.M. (1995), “Adapting to smoothness via Wavelet Shrinkage”, Biometrika Vol.90, No.432 hal 613-627.

- [12] Chang, S.G., Yu, B., Vetterli, M. (2000), "Adaptive Wavelet *Thresholding* for Image Denoising and Compression", IEEE Transaction Image Processing.
- [13] Kaur,L. Gupta, S. dan Chauchan, R.C. (2002), "Image Denoising Using Wavelet *Thresholding*", IEEE Transaction Image Processing.
- [14] M. Vranjes, S. Rimac-Drlje dan K. Grgic , (2008), "Locally Average Psnr As A Simple Objective Video Quality Metric", IEEE Transaction Image Processing, hal 17-20.
- [15] Huynh-Thu, Q., dan Ghanbari, M., (2008), "Scope of Validity of PSNR in Image/Video Quality Assesment", Electronic Letters, hal 800-810.
- [16] Van Fleet, Ptrick J., (2008), *Discrete Wavelet Transformations*, United States of America.
- [17] Y. Ennaji, M. Boulmalf dan C. Alaoui, (2009), "Experimental Analysis of Video Performance Over Wireless Local Area Networks", International Conference on Multimedia Computing and Systems, hal 488-494.
- [18] Donoho, D.L. (1995), "De-noising By Soft-*Thresholding* ", IEEE Transaction on Information Theory Vol.41, hal 613-627.

BIODATA PENULIS



Penulis memiliki nama lengkap Siti Khotijah yang biasa dipanggil Ijah, dilahirkan di Pamekasan tanggal 29 September 1990 dan merupakan anak ketiga dari tiga bersaudara pasangan Bapak Aliwafa dan Ibu Mutmainnah. Pendidikan formal yang pernah ditempuh yaitu TK Tarbiyatul Mubtadi'in Bandungan Pakong Pamekasan, SDN Pakong VII Pamekasan Madura, MTsN Model Sumber Bungur Pakong Pamekasan Madura, MAN Pamekasan Madura, penulis melanjutkan S1 Matematika di Universitas Islam Madura (UIM) yang masuk melalui jalur PMDK reguler tahun 2009 dan lulus pada bulan Maret tahun 2014. Setelah lulus S1, penulis bekerja sebagai guru SMP dan SMK Darul Falah Pegantenan Pamekasan hingga akhirnya penulis melanjutkan studi kembali di Jurusan Matematika ITS pada bulan Desember tahun 2014. Jika ingin memberikan saran, kritik dan pertanyaan mengenai tesis ini, dapat dikirimkan melalui e-mail ija.st29@gmail.com.