Image Enhancement Dengan Metode *Median Filter* Dan *Multiscale Retinex With Color Restoration* Pada *Dataset* Video

Agus Supriyanto, Galih Hermawan

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : supriyantoagus666@gmail.com, galih.hermawan@email.unikom.ac.id

ABSTRAK

Video sebagai salah satu media penyimpan informasi, dimana video terbentuk dari kumpulan gambar yang bergerak terhadap ketetapan satuan waktu tertentu atau disebut *frame per second* (fps). Informasi yang erkandung dalam video dapat berkurang bila kualitas video kurang baik. Pada penelitian ini dilakukan perbaikan kualitas video menggunakan metode *Median Filte*r dan *Multiscale Retinex with Color Restoraion* (MSRCR). *Median Filter* digunakan untuk mengurangi derau dan MSRCR digunakan untuk mengatur kecerahan video.

Pada penelitian ini dataset yang digunakan yaitu rekaman video CCTV. Proses yang dilakukan yaitu melakukan reduksi derau dengan *Median Filter*, dilanjutkan dengan metode MSRCR. terdapat beberapa tahapan pada metode MSRCR yaitu, *Gaussian Surround*, *Single Scale Retinex*, *Multiscale Retinex*, *Color Restoration* dan *Multiscale Retinex with Color Restoration*.

Pengujian yang dilakukan menggunakan pengujian funsionalitas dan pengujian metode. Pengujian metode menggunakan nilai *Peak Signal to Noise Ratio* (PSNR) dan penghitungan waktu proses. Nilai PSNR dihitung dengan perhitungan rat-rata, dikarenakan nilai PSNR pada setiap *frame* berbeda-beda namun tidak signifikan. Berdasarkan hasil pengujian nilai PSNR tertinggi yang didapatkan adalah 19.6088 dB dengan waktu proses tercepat yaitu ± 4 menit, dengan hasil yang lebih cerah dan warna yang mendekati warna asli. Namun masih terdapat sedikit derau yang terlihat. Adapun yang dapat terlihat dari pengujian adalah Nilai PSNR dan gambar tidak akan berubah jika nilai Beta jauh melebihi nilai Alpha, sehingga jika nilai Alpha lebih besar dari Beta maka akan menghasilkan video yang lebih tajam dan lebih cerah serta warna dapat dikembalikan mendekati warna asli.

Kata Kunci : Perbaikan Kualitas Video, *Median Filter*, *Multiscale Retinex With Color Restoration*, Restorasi Warna, PSNR

1. PENDAHULUAN

Video merupakan teknologi pemrosesan sinyal elektronik yang mewakilkan gambar bergerak. Video dapat digunakan dalam aplikasi teknik, keilmuan, produksi dan keamanan [1]. Video pada dasarnya tersusun dari banyak gambar atau citra, lalu gambar berubah pada satuan waktu maka dari itu video memiliki satuan *frame per second* (fps). Salah satu contoh penggunaan video pada bidang keamanan adalah *closed-circuit television* (CCTV) atau kamera pengintai. Kamera pengintai difungsikan untuk memantau pergerakan objek di suatu tempat tertentu. Kamera pengintai biasanya lebih banyak digunakan untuk mengintai seseorang seperti di bank, toko atau perkantoran. Kamera pengintai tersebut difungsikan untuk memantau pergerakan dari seseorang atau sekelompok orang. Kamera tersebut juga digunakan untuk mengenali objek atau seseorang dengan cara menyeleksi atau atau mengambil gambar (*capture*) dari video tersebut.

Namun citra yang dihasilkan terkadang kurang baik yakni terdapat buram (*blur*) derau (*noise*) atau pada kecerahan citra. Sehingga objek sulit terlihat atau dikenali dengan jelas. Hasil citra yang kurang baik tersebut dapat diturunkan dengan operasi pengolahan citra yaitu perbaikan kualitas citra agar objek dapat terlihat lebih jelas dibandingkan citra asli. Proses perbaikan kualitas citra menggunakan metode yang dapat mereduksi derau dengan citra hasil yang lebih baik dari citra asli, adapun pada proses pencerahan citra digunakan metode yang berbasiskan ketetapan warna (*color constancy*) agar warna pada citra dapat terlihat lebih baik.

Penelitian yang dilakukan oleh Christine Wulandari, Esti Suryani dan Umi Salamah mengatakan bahwa metode *Median Filter* dapat menghilangkan *noise salt and pepper* sehingga penggunaan metode untuk mendeteksi ciri dapat bekerja secara optimal [2]. Proses pencerahan citra menggunakan metode *Multiscale Retinex with Color Restoration*, pada penelitian yang dilakukan oleh Laurentius Kuncoro Probo Saputra mengatakan bahwa metode *Multiscale Retinex with Color Restoration* menghasilkan citra yang lebih baik dibandingkan dengan metode *Adaptive Multiscale Retinex* dan sangat mendukung dalam proses pencarian citra wajah [3].

Berdasarkan permasalahan tersebut perbaikan citra yang akan dilakukan akan difokuskan pada pengurangan intensitas derau dengan menggunakan metode *Median Filter*. Metode *Median Filter* dapat mereduksi derau dengan tingkat keburaman sedikit.

Proses pencerahan citra menggunakan metode *Multi Scale Retinex with Color Restoration* (MSRCR) yang dapat mencerahkan citra dengan mengembalikan warna asli namun memiliki ketetapan warna yang baik. Pengujian akan dilakukan dengan pengujian fungsionalitas dan pengujian metode menggunakan nilai PSNR dan menghitung wktu proses. Implementasi dari metode-metode tersebut untuk mengetahui kinerja dan video yang dihasilkan setelah proses perbaikan kualitas video.

1.1 Video

Video merupakan sinyal elektronik yang mewakilkan gambar bergerak. Penggunaan umum video adalah televisi namunvideo juga digunakan dalam aplikasi teknik, keilmuan, produksi, dan keamanan.

Video digital dapat disebut array tiga dimensi dari piksel berwarna. Array tersebut terdiri dari dua dimensi melayani arah spasial dari gambar bergerak (horizontal dan vertikal) dan satu dimensi lainnya akan merepresentasikan domain waktu.

Video digital terdiri atas serangkaian gambar digital yang ditampilkan pada kecepatan konstan, gambar tersebut dinamakan *frame* dan satuan ukuran untuk menghitung *frame* rata-rata yang ditampilkan disebut *frame per second* (fps). Setiap *frame* merupakan gambar digital yang terdiri dari raster piksel dan akan mempunyai lebar sebanyak W piksel dan tinggi sebanyak H piksel, maka dapat dikatakan bahwa frame size adalah W x H [1].

1.2 Pengolahan Citra

Pengolahan citra adalah cabang ilmu informatika untuk memperbaiki kualitas citra agar kualitasnya lebih baik atau lebih mudah diinterpretasi oleh manusia atau komputer. *Input* dan *ouput* dari program pengolahan citra adalah citra [4].

Terdapat dua prinsip daerah aplikasi pengolahan cira digital: peningkatan informasi piktorial untuk interpretasi manusia dan pengolahan citra untuk penyampaian, transmisi dan representasi bagi peralatan persepsi (*perception*) [5].

1.3 Perbaikan Kualitas Citra

Perbaikan kualitas citra bisa memiliki beberapa tujuan diantaranya :

1. Citra memiliki tampilan yang lebih baik menurut selera manusia.
2. Citra menjadi lebih mudah untuk dianalisis oleh proses otomatisasi berbasis citra.
3. Menghilangkan artifak-artifak pengganggu yang tidak diinginkan atau lebih dikenal dengan istilah derau.

Proses-proses yang termasuk kedalam perbaikan citra (*image enhancement*) diantaranya :

1. Pengubahan kecerahan citra (*image brightness*).
2. Perbaikan kontras.
3. Pengurangan derau.
4. Pelembutan citra (*image smoothing*).
5. Penajaman citra (*image sharpening*) [4].

1.4 MATLAB

MATLAB (Matrix Laboratory) adalah sebuah pemrograman untuk analisis dan komputasi numerik, merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks.

Matlab merupakan *software* yang dikembangkan oleh Mathworks, Inc. dan merupakan software yang paling efisien untuk perhitungan numerik berbasis matriks. Dengan demikian jika didalam perhitungan kita dapat memformulasikan masalah kedalam format matriks, maka matlab merupakan software terbaik untuk penyelesaian numeriknya.

Matlab yang merupakan bahasa pemrograman tingkat tinggi berbasis pada matriks sering digunakan untuk teknik komputasi numerik, digunakan untuk menyelesaikan masalah-masalah yang melibatkan operasi matematika elemen, matriks, optimasi, aproksimasi dan lain-lain. Matlab banyak digunakan pada :

1. Matematika dan Komputasi
2. Pengembangan dan Algoritma
3. Pemrograman *Modelling*, simulasi, dan pembuatan prototipe
4. Analisis Data, eksplorasi, dan visualisasi
5. Analisis numerik dan statistik
6. Pengembangna aplikasi teknik [6]

2. ISI PENELITIAN

Permasalahan yang diangkat pada penelitian ini adalah kualitas video dari CCTV yang kurang baik yaitu, masih memiliki *noise* dan tingkat kecerahan pada video yang kurang baik. Sehingga informasi pada video kurang dapat tersampaikan dengan baik, untuk memperbaiki kualitas video maka diterapkan metode *Median Filter* untuk mengurangi *noise* dan *Multiscale Retinex with Color Restoration* (MSRCR) untuk memperbaiki tingkat kecerahan video. Analisis tersebut dilakukan dengan menggunakan *dataset* video CCTV, dengan tujuan penelitian untuk mengetahui kinerja kedua metode tersebut dalam memperbaiki kualitas citra.

2.1 Video Masukan

Video masukan berupa video yang dapat berformat .AVI, .MP4, .MPG. Ukuran atau resolusi video yang digunakan dapat bervariasi seperti, 512 x 288, 640 x 480, 854 x 480. Adapun *frame rate* yang digunakan pada dataset seperti, 25fps dan 30fps.



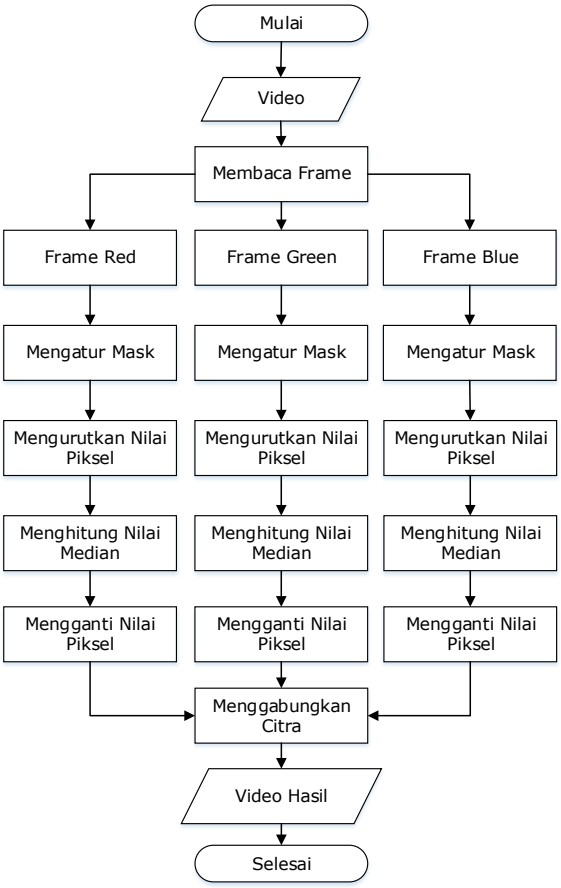
Gambar 1 Video Masukan

2.2 Reduksi Derau *Median Filter*

Filter median sangat populer dalam pengolahan citra. Filter ini dapat dipakai untuk menghilangkan derau bintik-bintik. Nilai yang lebih baik digunakan untuk suatu piksel ditentukan oleh nilai median dari setiap piksel dan kedelapan piksel tetangga pada 8-ketetanggaan [8]. Berikut adalah notasi *Median Filter* secara matematis :

$$g(y,x)=median\left(\begin{array}{l} f(y-1,x-1),\ f(y-1,x),\\ f(y-1,x+1),\ f(y,x-1),\\ f(y,x),\ f(y,x+1),\\ f(y+1,x-1),\ f(y+1,x),\\ f(y+1,x+1)\end{array}\right) \quad (1)$$

Pada proses ini dilakukan pengurangan intensitas derau (*noise*) dengan menggunakan metode *Median Filter*. Proses ini menghasilkan citra yang halus (*smooth*) dan cenderung menghilangkan noise, namun metode ini bekerja dengan baik pada *noise salt and pepper*. Adapun langkah yang dilakukan metode *Median Filter* adalah sebagai berikut:



Gambar 2 *Flowchart Median Filter* Citra Berwarna

Penjelasan cara kerja *Median Filter* adalah sebagai berikut:

1. Membaca *frame* pada video masukan.
2. Memisahkan *frame* berdasarkan kompoen warna yaitu *Red*, *Green* dan *Blue*. Kemudian dilanjutkan dengan pembacaan nilai pikselnya.

Berikut contoh dari matriks nilai piksel pada *frame* berdasarkan komponen warna RGB dan memiliki derau:

R : 177 G : 183 B : 197	R : 171 G : 179 B : 192	R : 161 G : 171 B : 183	R : 162 G : 174 B : 186
R : 173 G : 179 B : 195	**R : 45** **G : 53** **B : 66**	R : 165 G : 180 B : 187	R : 157 G : 174 B : 181
R : 157 G : 164 B : 183	R : 162 G : 175 B : 184	R : 208 G : 234 B : 231	R : 144 G : 174 B : 166
R : 167 G : 180 B : 197	R : 168 G : 185 B : 193	R : 140 G : 169 B : 164	R : 139 G : 171 B : 158

Gambar 3 Matriks nilai Piksel RGB

177	171	161	162
173	**45**	165	157
157	162	208	144
167	168	140	139

Frame Red

183	179	171	174
179	**53**	180	174
164	175	234	174
197	193	164	158

Frame Green

197	192	183	186
195	**66**	187	181
183	184	231	166
197	193	164	158

Frame Blue

Gambar 4 Matriks Nilai Piksel

Pada matriks tersebut terdapat *noise*, yaitu adanya nilai RGB yang signifikan berbeda. *Frame Red* terdapat nilai piksel 45, *Frame Green* terdapat nilai piksel 53 dan *Frame Blue* terdapat nilai piksel 66.

3. Mengurutkan nilai piksel yaitu memasukan nilai piksel yang dibaca oleh *mask* yang dilakukan berdasarkan persamaan (1) untuk mengambil nilai piksel dan dimasukan kedalam matrik 1x9 yang bertujuan untuk memudahkan proses *sorting* atau pengurutan, maka hasil dari proses tersebut dapat digambarkan seperti berikut:

177	171	161	162
173	**45**	165	157
157	162	208	144
167	168	140	139

Frame Red

183	179	171	174
179	**53**	180	174
164	175	234	174
197	193	164	158

Frame Green

197	192	183	186
195	**66**	187	181
183	184	231	166
197	193	164	158

Frame Blue

Gambar 5 Penempatan *Mask* Pada Matriks Piksel

R :	45	157	161	162	165	171	173	177	208
G :	53	164	171	175	179	179	180	183	234
B :	66	183	183	184	187	192	195	197	231

Gambar 6 Pengurutan Pada Matriks 1x9

4. Menghitung nilai median menggunakan persamaan sebagai berikut:

$$Median = \frac{n+1}{2} \quad (2)$$

Maka perhitungannya menjadi sebagai berikut:

$$Median = \frac{9+1}{2} = \frac{10}{2} = 5$$

Berdasarkan perhitungan tersebut nilai median adalah data ke-5 pada matriks.

R :	45	157	161	162	165	171	173	177	208
G :	53	164	171	175	179	179	180	183	234
B :	66	183	183	184	187	192	195	197	231

Gambar 7 Nilai Median Pada Matriks 1x9

5. Mengganti nilai pusat *mask* dengan nilai median.

177	171	161	162
173	**165**	**45**	157
157	162	208	144
167	168	140	139

Frame Red

183	179	171	174
53	**179**	180	174
164	175	234	174
197	193	164	158

Frame Green

197	192	183	186
195	**187**	**66**	181
183	184	231	166
197	193	164	158

Frame Blue

Gambar 8 Mengganti Nilai Pusat *Mask*

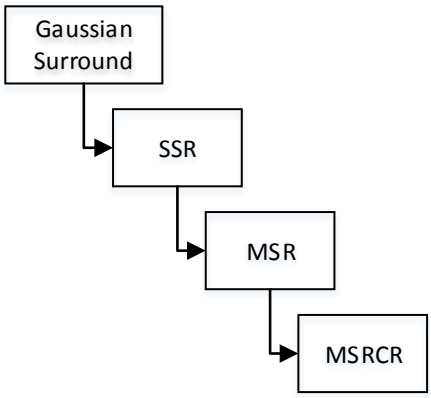
R : 177 G : 183 B : 197	R : 171 **G : 53** B : 192	R : 161 G : 171 B : 183	R : 162 G : 174 B : 186
R : 173 G : 179 B : 195	**R : 165** G : 179 **B : 187**	**R : 45** G : 180 **B : 66**	R : 157 G : 174 B : 181
R : 157 G : 164 B : 183	R : 162 G : 175 B : 184	R : 208 G : 234 B : 231	R : 144 G : 174 B : 166
R : 167 G : 180 B : 197	R : 168 G : 185 B : 193	R : 140 G : 169 B : 164	R : 139 G : 171 B : 158

Gambar 9 *Frame* RGB Setelah Perbaikan Derau

2.3 Multiscalse Retinex With Color Restoration

Retinex teori dijelaskan oleh Edwin H. Land pada tahun 1971. Kata "*Retinex*" adalah penggabungan kata yang terbentuk dari kata "*Retina*" dan "*Cortex*", yang menunjukan bahwa baik mata dan otak terlibat dalam pemrosesan. Teori Retinex yaitu teori yang berdasarkan ketetapan warna "*Color Constancy*" yang didasarkan bahwa persepsi mata manusia pada warna tergantung pada karakteristik refleksi permukaan objek, dan tidak berhubungan dengan kejadian tersebut.

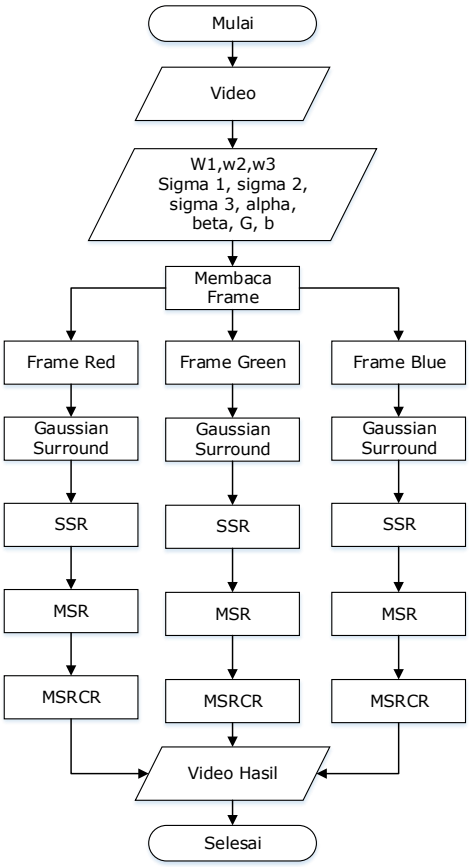
Terdapat beberapa perkembangan pada teori *Retinex* yaitu, *Single-Scale Retinex* (SSR), *Multi-Scale Retinex* (MSR), dan *Multi-Scale Retinex with Color Restoration* (MSRCR). MSRCR memiliki hasil lebih baik dari tipe *Retinex* yang lain, namun tidak dapat mempertahankan ketetapan corak sehingga masih ada sedikit distorsi warna [7]. Adapun tahapan yang dilakukan pada MSRCR adalah sebagai berikut :



Gambar 10 Tahapan MSRCR

2.4 Pencerahan Citra (MSRCR)

Proses ini dilakukan pencerahan gambar dengan menggunakan metode *Multiscale Retinex with Color Restoration* (MSRCR). Gambar yang dihasilkan memiliki tingkat kecerahan yang baik namun tetap mempertahankan ketetapan warna. Adapun tahapan yang dilakukan metode MSRCR adalah sebagai berikut:



Gambar 11 *Flowchart* MSRCR

2.4.1 Perhitungan *Gaussian Surround*

Gaussian Surround digunakan sebagai *filter* untuk memperhalus citra asli. Tahap ini akan digunakan untuk proses SSR [7]. Adapun rumus yang digunakan adalah sebagai berikut:

$$F(x,y)=Ke^{-(x^2+y^2)/(2\times\sigma^2)} \quad (3)$$

Berikut penjelasan dari persamaan tersebut:
F(x,y) = Gaussian pada piksel (x,y)
e = eksponensial
(x,y) = koordinat piksel
σ = nilai sigma

Tahap awal yang dilakukan untuk perbaikan kecerahan video adalah *Gaussian Surround*. Tahap ini dimaksudkan untuk penghalusan video terlebih dahulu. Nilai *Gaussian* didapat dari persamaan (3) dan menggunakan tiga nilai sigma. Secara *default* tiga nilai sigma yang digunakan adalah sebagai berikut :

$\sigma 1 = 15$
$\sigma 2 = 80$
$\sigma 3 = 250$

Namun nilai tersebut dapat diubah sesuai kebutuhan. Berikut contoh perhitungan nilai Gaussian menggunakan nilai sigama ke-1 yaitu 15:

$$\begin{bmatrix} e^{-\frac{1^2+-1^2}{2(15)^2}} & e^{-\frac{1^2+0^2}{2(15)^2}} & e^{-\frac{1^2+1^2}{2(15)^2}} \\ e^{-\frac{0^2+-1^2}{2(15)^2}} & e^{-\frac{0^2+0^2}{2(15)^2}} & e^{-\frac{0^2+1^2}{2(15)^2}} \\ e^{-\frac{-1^2+-1^2}{2(15)^2}} & e^{-\frac{-1^2+0^2}{2(15)^2}} & e^{-\frac{-1^2+1^2}{2(15)^2}} \end{bmatrix}$$

Didapatkan hasil sebagai berikut:

$$\begin{bmatrix} 0{,}996565417 & 0{,}997780245 & 0{,}995565417 \\ 0{,}997780245 & 1 & 0{,}997780245 \\ 0{,}995565417 & 0{,}997780245 & 0{,}995565417 \end{bmatrix}$$

Total = 8,97338265

Total keseluruhan nilai matriks harus sama dengan 1, jika tidak memenuhi maka harus dilakukan normalisasi dengan rumus :

$$a_{(x,y)} = \frac{a}{sum(a)} \quad (4)$$

Contoh:

$$a_{(1,1)} = \frac{0.995565417}{8.97338265} = 0{,}110946502$$

$$a_{(1,2)} = \frac{0{,}997780245}{8.97338265} = 0{,}111193324$$

$$a_{(1,3)} = \frac{0.995565417}{8.97338265} = 0{,}110946502$$

Perhitungan dilakukan pada seluruh elemen matrik, sehingga didapatkan hasil sebagai berikut:

$$\begin{bmatrix} 0{,}110946502 & 0{,}111193324 & 0{,}110946502 \\ 0{,}111193324 & 0{,}111440695 & 0{,}111193324 \\ 0{,}110946502 & 0{,}111193324 & 0{,}110946502 \end{bmatrix}$$

Gambar 12 Kernel Gaussian σ1=15

Perhitungan dilakukan dengan nilai σ2=80 dan σ3=250.

2.4.2 Perhitungan SSR

Multiscale Retinex (MSR) pada dasarnya menggunakan metode Singlescale Retinex (SSR) namun dengan skala yang berbeda [7]. Adapun persamaan dari SSR adalah sebagai berikut:

$$R_i(x,y) = \log I_i(x,y) - \log[F(x,y) * I_i(x,y)] \quad (5)$$

Berikut penjelasan dari persamaan tersebut :

$R_i(x,y)$ = *output Retinex*

$I_i(x,y)$ = distribusi citra pada piksel ke i(x,y)

$*$ = operator konvolusi

i = kanal warna (R,G,B)

Tahap awal yang dilakukan pada metode SSR adalah melakukan konvolusi matriks citra asli dengan kernel gausian yang terdapat pada persamaan (5). Matriks citra asli yang digunakan adalah matriks citra setiap kanal warna. Adapun rumus yang digunakan untuk perhitungan konvolusi adalah sebagai berikut:

$$F(x,y) * I_i(x,y) \quad (6)$$

Berikut penjelasan dari rumus konvolusi:

F(x,y) = kernel *Gaussian*

"*" = konvolusi

$I_i(x,y)$= matriks citra

i = kanal warna (R,G,B)

Perhitungan konvolusi pada matriks citra *frame red* hasil *Median Filter* adalah sebagai berikut:

0	0	0	0	0	0
0	177	171	161	162	0
0	173	165	45	157	0
0	157	162	208	144	0
0	167	168	140	139	0
0	0	0	0	0	0

$$\begin{bmatrix} 0{,}110946502 & 0{,}111193324 & 0{,}110946502 \\ 0{,}111193324 & 0{,}111440695 & 0{,}111193324 \\ 0{,}110946502 & 0{,}111193324 & 0{,}110946502 \end{bmatrix}$$

Berikut perhitungan konvolusi *frame red* dengan nilai $\sigma 1 = 15$

(0 x 0,110946502) + (0 x 0,111193324) +
(0 x 0,110946502) + (0 x 0,111193324) +
(177 x 0,111440695) + (171 x 0,111193324) +
(0 x 0,110946502) + (173 x 0,111193324) +
(165 x 0,110946502) = 76,2816793

Perhitungan dilakukan dengan bergeser kekanan satu piksel.

Seteleah piksel paling kanan habis, maka berpindah turun satu piksel dari piksel awal.

Sehingga didapatkan hasil sebagai berikut:

76,2816793	99,17293825	95,69780201	58,40546221
111,7098941	157,6506271	152,7442329	97,46430508
110,2604229	153,9132463	147,5609364	92,61399825
72,72175969	111,3671789	106,8158888	70,14603304

Gambar 13 Hasil Konvolusi *Frame Red* σ1=15

Perhitungan juga dilakukan dengan menggunakan σ2=80 dan σ3=250. Begitu juga pada *Frame Green* dan *Frame Blue*.

2.4.3 Perhitungan MSR

Ide dari dasar *Multi-Scale Retinex* untuk masalah SSR terdapat pada skala yang berbeda yang digunakan dan diberi bobot yang berbeda untuk menggabungkan kelebihan dan menghilangkan kelemahan yang dimiliki dari skala rendah dan skala besar [7]. Adapun persamaan dari MSR adalah sebagai berikut:

$$R_{MSR_i}(x,y) = \sum_{n=1}^{N} W_n \, R_{n_i} \quad (7)$$

Berikut penjelasan dari persamaan tersebut :

R_{MSR_i} = output dari MSR

N = jumlah skala

Wn = bobot yang diasosiasikan dengan skala ke-n

R_{n_i} = output dari SSR yang diasosiasikan dengan skala ke-n

i = kanal warna (R,G,B)

Tahap awal dari metode MSR adalah menentukan bobot yang akan digunakan. Terdapat 3 pembobotan yang akan digunakan. Setiap bobot akan dikalikan dengan hasil setiap SSR lalu dijumlahkan.

Contoh pembobotan:

W1 = 0,33

W2 = 0,34

W3 = 0,33

Nilai bobot dapat disesuaikan dengan kebutuhan.

Berdasarkan persamaan (7) N=3. Maka bila persamaan tersebut diuraikan akan menjadi sebagai berikut:

$$R_{MSR_i}(x,y) = W_1 \times SSR1 + W_2 \times SSR2 + W_3 \times SSR3$$

Perhitungan MSR *Frame Red*

0,365553021	0,23660293	0,225923913	0,443061549	
0,189954463	0,019788242	-0,530752308	0,207054062	X 0,33
0,153479999	0,022239016	0,149091932	0,191685859	
0,361052092	0,178552063	0,117492177	0,297011685	

SSR1

0,36587976	0,23686424	0,226060268	0,44357935	
0,19011664	0,01974561	-0,53084433	0,20713946	X 0,34
0,1536252	0,02230533	0,149107215	0,19195036	
0,36136966	0,17867943	0,117641707	0,29722035	

SSR2

76,22243604	99,11133348	95,66677862	58,33359273	
111,6668221	157,666609	152,7776572	97,44451584	X 0,33
110,2223596	153,8889765	147,5555749	92,55576577	
72,66686479	111,333455	106,7779149	70,11123668	

SSR3

Setelah dilakukan perhitungan maska didapatkan hasil dari MSR *Frame Red* adalah sebagai berikut:

25,39843551	32,86535286	31,72145233	19,54711289
36,97737592	52,0432246	50,06099154	32,29544548
36,47625964	50,79828493	48,79323651	30,67192216
24,22207826	36,85971334	35,31548252	23,33577688

Gambar 14 MSR *Frame Red*

Setelah mendapatkan hasil, selanjutnya dilakukan normalisasi dengan rumus sebagai berikut:

$$MSR_i = 255 \times \frac{(x,y) - min}{max - min} \quad (8)$$

(x,y) = nilai piksel
min = nilai piksel terkecil
max = nilai piksel terbesar
Berikut hasil setelah proses normalisasi:

45,91587084	104,5094632	95,53316978	0
136,7768893	255	239,4452335	100,0373472
132,8445834	245,230843	229,4970423	87,29740928
36,68488649	135,8535801	123,7358577	29,72999742

Gambar 15 Normalisasi MSR *Frame Red*

Lakukan perhitungan pada *Frame Green* dan *Frame Blue.*

2.4.4 Perhitungan MSRCR

MSRCR merupakan pengembangan dari MSR, dimana metode ini mampu memperbaiki kualitas citra yaitu pada pencerahan citra dengan mempertahankan *color constancy*. *Color constancy* atau ketetapan diambil dari sistem penglihatan manusia yang mengusahakan warna dari suatu benda teteap terlihat sama walaupun pada kondisi pencahayaan yang berbeda-beda [7].

Sebelum melakukan perhitungan MSRCR, terlebih dahulu harus melakukan perhitungan untuk *Color Restoration* (CR). Adapun persamaan dari CR adalah sebagai berikut:

$$C_i(x,y) = \beta \left\{ \log[\alpha I_i(x,y)] - \log[\sum_{i=1}^{S} I_i(x,y)] \right\} \quad (9)$$

Atau dapat diuraikan menjadi berikut:

$$C_i(x,y) = \beta(\log(\alpha I_i(x,y)) - \log(I_R(x,y) + I_G(x,y) + I_B(x,y)))$$

Berikut penjelasan dari persamaan tersebut:

β = konstanta *gain*

α = konstanta kontrol ketidaklinieran

S = 3, kanal warna (R,G,B)

Tahap awal pada perhitungan MSRCR adalah melakukan proses *color restoration* terlebih dahulu. Sebelumnya terdapat parameter yanga harus ditetapkan terlebih dahulu yaitu:

β = 46

α = 125

Nilai alpha dan beta dapat disesuaikan dengan kebutuhan.

Tahap selanjutnya adalah perhitungan MSRCR. adapun persamaan dari MSRCR adalah sebagai berikut:

$$R_{MSRCR_i}(x,y) = G[C_i(x,y)R_{MSR_i}(x,y) + b] \quad (10)$$

G dan b adalah nilai akhir *gain* dan *offset.*
Berikut nilai dari parameter G dan b:
G = 192
b = -30
Namun nilai tersebut dapat disesuaikan sesuai kebutuhan.
Berikut adalah hasil setelah perbaikan kualitas:

R : 177 G : 183 B : 197	R : 171 G : 179 B : 192	R : 161 G : 171 B : 183	R : 162 G : 174 B : 186
R : 173 G : 179 B : 195	R : 45 G : 53 B : 66	R : 165 G : 180 B : 187	R : 157 G : 174 B : 181
R : 157 G : 164 B : 183	R : 162 G : 175 B : 184	R : 208 G : 234 B : 231	R : 144 G : 174 B : 166
R : 167 G : 180 B : 197	R : 168 G : 185 B : 193	R : 140 G : 169 B : 164	R : 139 G : 171 B : 158

Citra Asli

R : 177 G : 183 B : 197	R : 171 G : 53 B : 192	R : 161 G : 171 B : 183	R : 162 G : 180 B : 186
R : 173 G : 179 B : 195	R : 157 G : 175 B : 66	R : 162 G : 174 B : 184	R : 45 G : 174 B : 231
R : 157 G : 164 B : 183	R : 165 G : 179 B : 187	R : 208 G : 174 B : 181	R : 144 G : 234 B : 166
R : 167 G : 180 B : 197	R : 168 G : 185 B : 193	R : 140 G : 169 B : 164	R : 139 G : 171 B : 158

Median Filter

R : 46 G : 255 B : 233	R : 105 G : 182 B : 194	R : 96 G : 158 B : 191	R : 0 G : 224 B : 255
R : 147 G : 0 B : 194	R : 255 G : 111 B : 131	R : 194 G : 116 B : 0	R : 100 G : 155 B : 187
R : 176 G : 165 B : 174	R : 245 G : 105 B : 130	R : 229 G : 142 B : 164	R : 86 G : 160 B : 180
R : 36 G : 239 B : 235	R : 135 G : 169 B : 178	R : 122 G : 148 B : 164	R : 30 G : 205 B : 213

MSRCR

Gambar 16 Perbandingan Matriks Piksel

2.5 Pengujian

Pengujian dilakukan untuk mengetahui apakah metode yang diterapkan berjalan sesuai dengan tujuan atau tidak. Proses pengujian metode dilakukan dengan menghitung waktu proses dan menggunakan *Peak Signal to Noise Ratio* (PSNR) dimana untuk mendapatkan nilai PSNR maka harus menghitung terlebih dahulu nilai *Mean Squared*

Error (MSE). Adapun rumus dari MSE adalah sebagai berikut :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \quad (11)$$

Dan rumus PSNR adalah sebagai berikut :

$$PSNR = 10\log_{10}\left(\frac{C_{max}^2}{MSE}\right) \quad (12)$$

Pengujian dilakukan pada 3 data dengan masing-masing 5 kali pengujian.

Tabel 1 Pengujian Data Ke-1

No	Alpha	Beta	PSNR	Waktu Proses
1	50	100	19.0464 dB	308 Detik
2	30	30	19.0743 dB	417 Detik
3	115	100	18.7869 dB	292 Detik
4	34	33	19.09 dB	424 Detik
5	255	255	18.5328 dB	310 Detik

Tabel 2 Pengujian Data Ke-2

No	Alpha	Beta	PSNR	Waktu Proses
1	100	50	14.8 dB	373 Detik
2	225	225	15.6297 dB	339 Detik
3	100	100	15.251 dB	345 Detik
4	255	100	15.6297 dB	361 Detik
5	55	105	14.8743 dB	359 Detik

Tabel 3 Pengujian Data Ke-2

No	Alpha	Beta	PSNR	Waktu Proses
1	100	50	19.1444 dB	411 Detik
2	30	30	19.5561 dB	290 Detik
3	28	100	19.6088 dB	281 Detik
4	255	255	18.2844 dB	398 Detik
5	29	255	19.5826 dB	398 Detik

Berdasarkan hasil pengujian yang telah dilakukan, dari ketiga video nilai PSNR terbesar dimiliki oleh video ketiga pada pengujian ketiga sebesar 19.6088 dB dan waktu proses tercepat yaitu ± 4 menit

3. PENUTUP

Berdasarkan hasil pengujian didapatkan nilai PSNR terbesar adalah 19.6088 dB dengan waktu proses tercepat yaitu ± 4 menit, namun dapat disimpulkan bahwa nilai PSNR lebih menilai dari kemiripan video hasil dengan video asli.

Nilai PSNR tersebut adalah hasil dari perhitungan rata-rata PSNR pada seluruh *frame* video, perhitungan dilakukan dikarenakan nilai PSNR pada setiap *frame* berbeda namun tidak signifikan. Jika tidak dilakukan perhitungan rata-rata maka nilai PSNR akan sangat besar jumlahnya.

Adapun yang dapat terlihat dari pengujian tersebut adalah, jika nilai Alpha lebih besar dari Beta maka akan menghasilkan video yang lebih tajam, lebih cerah dan detil gambar akan terlihat, namun masih terdapat sedikit derau. Nilai PSNR tidak akan bertambah bila nilai Beta jauh melebihi nilai Alpha, gambar yang dihasilkanpun tidak akan berubah.

Aplikasi yang dibangun masih terdapat kekurangan dan dapat dilakukan pengembangan lebih lanjut untuk mendapatkan hasil yang lebih baik. Saran untuk pengembangan selanjutnya yang dapat dijadikan pertimbangan yaitu:

1. Lebih banyak variabel yang dapat dibuat dinamis sesuai kebutuhan pengguna.
2. Menggunakan kombinasi metode lain untuk mereduksi derau ataupun untuk memperbaiki kecerahan video.
3. Menggunakan metode pengujian lain yang lebih optimal dalam tahap pengujian hasil perbaikan video.
4. Aplikasi dapat dikembangkan dengan deteksi objek seperti manusia atau objek wajah.

DAFTAR PUSTAKA

[1] I. Binanto, Multimedia Digital - Dasar Teori dan Pengembangannya, Yogyakarta: Andi, 2010.

[2] C. Wulandari, E. Suryani dan U. Salamah, "Segmentasi Citra Menggunakan HAAR Wavelet Untuk Deteksi Penyakit TBC Dari Citra Bernoise," *Jurnal ITSMART,* vol. 3, pp. 9-15, 2014.

[3] L. K. Probo S, "Perbandingan Varian Metode Multiscale Retinex untuk Peningkatan Akurasi Wajah Adaboost HAAR-like," *Jurnal JUTISI,* vol. 2, pp. 89-98, 2016.

[4] P. Hidayatullah, Pengolahan Citra Digital Teori Dan Aplikasi Nyata, Bandung: Informatika, 2017.

[5] E. Prasetyo, Pengolahan Citra Digital Dan Aplikasinya Menggunakan Matlab, Yogyakarta: Andi, 2011.

[6] M. Arhami dan A. Desiani, Pemrograman Matlab, Yogyakarta: Andi, 2005.

[7] G. Yin, S. Iijun, C. Feiyan dan L. Liansha, "Enhancement MSRCR Algorithm of Color Fog Image Based on the Adaptive Scale," dalam *Sixth International Conference on Digital Image Processing*, Athens, Greece, 2014.