

Perancangan Aplikasi Perbaikan Citra Digital pada Hasil Screenshot dengan Menggunakan Metode Multiscale Retinex dan Median Filter

Noga Adi Putra¹, Resti Amalia²

^{1, 2}Department of Technical Information, Universitas Pamulang, Indonesia

Article Info

Article history:

Received July 11, 2022

Revised Sep 28, 2022

Accepted Oct 6, 2022

Keywords:

Image

Image Enhancement

Multiscale Retinex

Median Filter

ABSTRACT

The image is a representation of the light intensity function in a discrete form on a two-dimensional plane. An image is a form of information other than text, sound, and video. However, sometimes the image can also experience a decrease, namely degradation or loss of quality caused by noise, too contrasting colors, and blurring. So this study aims to improve the quality of a digital image by analyzing the use of Multiscale Retinex and Median Filter methods in improving digital images on screenshots, as well as to implement these methods in improving digital images. The data used in this study using library research, while the data used are secondary data from document studies and from library research. The method used in the improvement of digital images is the Multiscale Retinex and Median Filter methods. From the results of the application of the Multiscale Retinex method and the Median Filter on the screenshot image that has been done, it can improve the quality of a digital image. The Multiscale Retinex method can brighten digital images and the Median Filter can reduce noise/noise in digital images which makes digital image quality better. The results of several image data that have been tested are then analyzed with a histogram, the original image analyzed with a histogram shows a graph of the average pixel intensity value approaching the value 0, but after being processed using the Multiscale Retinex and Median Filter method, the average pixel intensity graph value is shown by the histogram close to the value of 250 which shows the brightness and sharpness of the digital image for the better.

Copyright © 2022 Universitas Indraprasta PGRI.
All rights reserved.

Corresponding Author:

Noga Adi Putra,

Universitas Pamulang,

Jl. Surya Kencana No. 1, Pamulang Barat, Pamulang, Tangerang Selatan.

Email: nogaadiputra2@gmail.com

1. PENDAHULUAN

Salah satu contoh perkembangan teknologi saat ini adalah pada bidang citra [1]. Citra merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra merupakan salah satu bentuk informasi selain teks, suara, dan video. Informasi yang terkandung dalam sebuah citra dapat diinterpretasikan berbeda-beda oleh manusia. Artinya, nilai pada sebuah citra bersifat subjektif tergantung keperluan masing-masing manusia [2]. Namun terkadang citra juga dapat mengalami penurunan yaitu

degradasi atau penurunan kualitas yang disebabkan oleh derau/*noise*, warna terlalu kontras, kabur, dan lain-lain. Citra yang memiliki gangguan biasanya terjadi karena banyak faktor, misalnya pada saat pengambilan gambar pencahayaan tidak maksimal, resolusi piksel pada kamera rendah, kemampuan dalam pengambilan gambar buruk, dan juga alat yang digunakan pada pengambilan gambar kurang baik. Salah satunya adalah pada saat pengambilan gambar bergerak oleh cctv, terbatasnya memori dan buffer dapat menjadi penyebab gambar yang dihasilkan memiliki kualitas yang buruk [3]. Maka untuk memberikan solusi pada masalah tersebut diperlukan adanya suatu sistem untuk mengolah sebuah citra agar citra menjadi lebih baik kualitasnya. Pada sistem pengolahan citra diperlukan adanya metode yang digunakan untuk memperbaiki kualitas citra. Metode yang digunakan antara lain *Multiscale Retinex* dan *Median Filter*.

Multiscale Retinex merupakan sebuah algoritma yang mempertahankan orisinalitas warna meskipun dalam keadaan pencahayaan yang berbeda warna objek yang dilihat memiliki warna yang relatif sama. Seperti suatu objek yang dilihat tetap memiliki warna yang sama meskipun dilihat saat siang yang cerah maupun sore yang memiliki warna kemerahan [4]. Algoritma *retinex* dapat mempertajam detail gambar karena memiliki konstansi warna dan juga rentang dinamis yang tinggi. Maka dari itu diharapkan dapat meningkatkan kualitas pada sebuah citra pada bagian pencerahannya.

Selain itu, *Median Filter* adalah salah satu teknik dalam pengolahan citra pada domain spasial yang dapat digunakan untuk peningkatan kualitas citra (*image enhancement*) terutama mengurangi *noise* (distorsi) pada sebuah citra. Median filter bekerja dengan mengevaluasi tingkat *brightness* dari suatu pixel dan menentukan pixel mana yang tingkat *brightness*-nya adalah nilai median (nilai tengah) dari semua pixel [5]. Berdasarkan latar belakang diatas, maka akan dilakukan sebuah penelitian terhadap citra digital pada hasil screenshot agar sebuah citra digital tersebut memiliki kualitas yang lebih baik setelah dilakukan sebuah proses perbaikan citra digital.

2. METODE

Peralatan yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Software: Matlab 2015a

Perangkat keras yang digunakan yaitu berupa laptop dengan spesifikasi sebagai berikut.

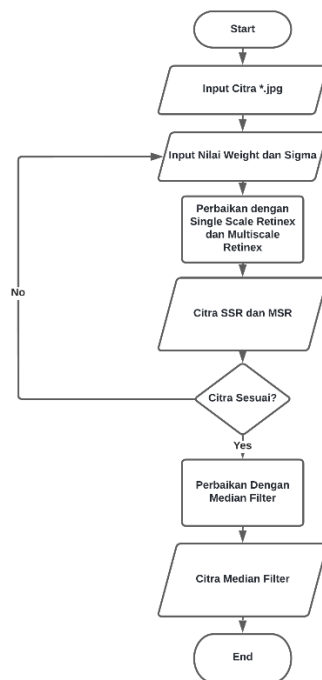
1. AMD A9-9425 Dual-Core @3.1 GHz
2. RAM 8 GB
3. HDD 1 TB

Penelitian dimulai dengan menginputkan citra dengan format file *.jpg, lalu mengisi nilai *weight* dan *sigma* yang masing-masing terdiri dari tiga yaitu $W1 + W2 + W3$ dan $S1 + S2 + S3$. Setelah itu melakukan proses perbaikan citra dengan *Multiscale Retinex* dengan menekan tombol proses MSR.

Proses *Multiscale Retinex* dimulai dari input dengan keterangan $N = 3$ yang merupakan jumlah skala yang dipakai, $n = 1$ merupakan langkah SSR pertama dimana SSR memiliki tiga langkah, W adalah nilai *weight* dimana $W1 + W2 + W3$ yang jika dijumlahkan nilainya adalah 1. *Sigma* 1,2 dan 3 merupakan sigma yang terdiri dari nilai rendah, sedang, dan tinggi. Selanjutnya adalah mencari nilai *Gaussian* dengan rumus $Gn = [F(x, y) * (x, y)]$, kemudian SSR dengan rumus $SSR_n = \log In(x, y) - \log Gn(x, y)$ yang nantinya akan mendapatkan nilai SSR_n. Jika $N = (n + 1)$ maka akan mengulangi proses mencari nilai Gaussian dengan n ditambah 1. Proses pencarian nilai Gaussian diulangi sebanyak tiga kali sehingga mendapatkan tiga *Single Scale Retinex*, setelah itu masuk ke langkah *Multiscale Retinex* dengan menjumlahkan hasil dari SSR₁, 2, dan 3 yang akan menghasilkan citra *Multiscale Retinex*.

Langkah selanjutnya adalah dengan menentukan sendiri apakah citra yang dihasilkan sudah sesuai atau belum, jika belum maka ulangi langkah yang sebelumnya. Selanjutnya adalah dengan memproses citra menggunakan metode *Median Filter* sehingga citra menjadi lebih jelas. Proses *Median Filter* dimulai dari citra hasil MSR yang kemudian diproses dengan formula $f(x, y) = \text{median}\{g(s, t)\}$ lalu dilakukan perhitungan untuk mencari nilai tengah dari piksel citra dari terendah hingga tertinggi, sehingga akan ditemukan hasil dari nilai median. Setelah perhitungan selesai maka akan didapat nilai piksel baru yang menghasilkan citra *Median Filter*.

Tahapan penelitian digambarkan pada *flowchart*. *Flowchart* merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem [6]. *Flowchart* metodologi penelitian ditunjukkan pada gambar 1.



Gambar 1. Flowchart proses kerja sistem

3. HASIL DAN PEMBAHASAN

3.1 Implementasi *Single Scale Retinex*

Dalam perbaikan citra digital yang menggunakan metode *Multiscale Retinex* (MSR), tahap awal sebelum proses *Multiscale Retinex* (MSR) adalah dengan melakukan proses *Single Scale Retinex* (SSR), karena pada dasarnya *Multiscale Retinex* (MSR) menggunakan metode *Single Scale Retinex* (SSR) namun dengan skala yang berbeda, yaitu dengan melalui tiga tahap SSR [7], [8]. Rumus SSR adalah sebagai berikut:

$$SSRn(x, y) = \log I_n(x, y) - \log G_n(x, y)$$

Proses selanjutnya dengan mengkonversi citra ke dalam matriks untuk mengetahui nilai pikselnya dengan menggunakan aplikasi Matlab. Untuk ukuran matriks yang akan digunakan adalah matriks 5x5.

0	0	0	0	0	0	0
0	5	8	13	19	18	0
0	10	15	19	19	17	0
0	14	21	20	20	13	0
0	15	23	20	20	12	0
0	15	25	21	21	12	0
0	0	0	0	0	0	0

Gambar 2. Contoh nilai piksel

Selanjutnya adalah perhitungan untuk mendapatkan nilai *Gaussian* sebagai berikut:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Keterangan:

σ = sebuah sigma

$$\pi = \frac{22}{7}$$

$$e = 2,71828182846$$

Misalnya $\sigma = 7.5$, maka

$$\begin{pmatrix} \frac{1}{2\pi(7,5)^2} e^{-\frac{-1^2+1^2}{2(7,5)^2}} & \frac{1}{2\pi(7,5)^2} e^{-\frac{0^2+1^2}{2(7,5)^2}} & \frac{1}{2\pi(7,5)^2} e^{-\frac{1^2+1^2}{2(7,5)^2}} \\ \frac{1}{2\pi(7,5)^2} e^{-\frac{-1^2+0^2}{2(7,5)^2}} & \frac{1}{2\pi(7,5)^2} e^{-\frac{0^2+0^2}{2(7,5)^2}} & \frac{1}{2\pi(7,5)^2} e^{-\frac{1^2+0^2}{2(7,5)^2}} \\ \frac{1}{2\pi(7,5)^2} e^{-\frac{-1^2+-1^2}{2(7,5)^2}} & \frac{1}{2\pi(7,5)^2} e^{-\frac{0^2+-1^2}{2(7,5)^2}} & \frac{1}{2\pi(7,5)^2} e^{-\frac{1^2+-1^2}{2(7,5)^2}} \end{pmatrix}$$

$$\begin{pmatrix} 0.0028294212 & 0.0028043822 & 0.0027795649 \\ 0.0028546837 & 0.0028294212 & 0.0028043822 \\ 0.0028801718 & 0.0028546837 & 0.0028294212 \end{pmatrix}$$

Sehingga total elemen matriks = 0.0254661321

Total keseluruhan elemen matriks setelah dijumlahkan harus bernilai 1, jika tidak bernilai 1 maka harus dilakukan perhitungan ulang. Perhitungan total keseluruhan elemen matriks adalah sebagai berikut:

$$New \alpha = \alpha \times \frac{1}{Total}$$

$$New \alpha_{(1,1)} = 0.0028294212 \times \frac{1}{0.0254661321} = 0.111105259$$

$$New \alpha_{(1,2)} = 0.0028546837 \times \frac{1}{0.0254661321} = 0.112097263$$

$$New \alpha_{(1,3)} = 0.0028801718 \times \frac{1}{0.0254661321} = 0.113098125$$

$$New \alpha_{(2,1)} = 0.0028043822 \times \frac{1}{0.0254661321} = 0.110122031$$

$$New \alpha_{(2,2)} = 0.0028294212 \times \frac{1}{0.0254661321} = 0.111105259$$

$$New \alpha_{(2,3)} = 0.0028546837 \times \frac{1}{0.0254661321} = 0.112097263$$

$$New \alpha_{(3,1)} = 0.0027795649 \times \frac{1}{0.0254661321} = 0.109147510$$

$$New \alpha_{(3,2)} = 0.0028043822 \times \frac{1}{0.0254661321} = 0.110122031$$

$$New \alpha_{(3,3)} = 0.0028294212 \times \frac{1}{0.0254661321} = 0.111105259$$

Masukkan nilai-nilai yang telah dihitung tadi ke dalam matriks seperti berikut:

$$\begin{pmatrix} 0.111105259 & 0.110122031 & 0.109147510 \\ 0.112097263 & 0.111105259 & 0.110122031 \\ 0.113098125 & 0.112097263 & 0.111105259 \end{pmatrix}$$

Langkah selanjutnya ialah menghitung nilai konvolusi dengan memindahkan *kernel* piksel per piksel, kemudian dilakukan perkalian dan penjumlahan dengan nilai *Gaussian* agar mendapatkan hasil konvolusi, yaitu $f(x, y) * G(x, y)$.

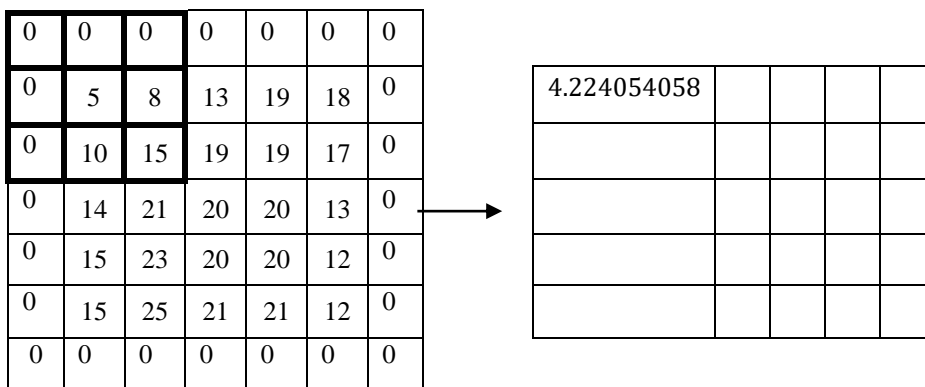
0	0	0	0	0	0	0
0	5	8	13	19	18	0
0	10	15	15	19	17	0
0	14	21	20	20	13	0
0	15	23	20	20	12	0
0	15	25	21	21	12	0
0	0	0	0	0	0	0

$f(x, y)$

0.111105259	0.110122031	0.109147510
0.112097263	0.111105259	0.110122031
0.113098125	0.112097263	0.111105259

$G(x, y)$

Untuk mendapatkan nilai konvolusi maka lakukan perhitungan nilai matriks $f(x, y)$ dan nilai gaussian $G(x, y)$ dengan melakukan perhitungan dari sudut paling kiri nilai matriks [8].



- Cara Perhitungan
 $(0 \times 0.111105259) + (0 \times 0.110122031) + (0 \times 0.109147510) + (0 \times 0.112097263) + (5 \times 0.111105259) + (8 \times 0.110122031) + (0 \times 0.113098125) + (10 \times 0.112097263) + (15 \times 0.111105259) = 4.224054058$

Selanjutnya geser *kernel* satu piksel ke kanan, lalu lakukan perhitungan seperti yang telah dilakukan sebelumnya sampai mendapatkan seluruh nilai hasil konvolusi. Untuk hasil konvolusi SSR 1 aditunjukkan pada tabel 1.

Tabel 1. Hasil konvolusi SSR1

4.224054058	7.804354906	10.37078485	11.71796267	8.184260505
8.089245411	13.89481173	17.23021695	17.56254071	11.8310608
10.84334914	17.43851259	19.6765958	17.79163389	11.27651773
12.49228216	19.32242089	21.23609019	17.6913445	10.94912773
8.581852855	13.15398887	14.38901644	11.74303325	7.230875183

- Hasil SSR1
 $SSRn(x, y) = \log In(x, y) - \log Gn(x, y)$
 Keterangan, $\log In(x, y)$ adalah nilai piksel citra awal.

Contoh:

- $5 - 4.224054058 = 0.775945942$
- $10 - 8.089245411 = 1.910754589$

- $14 - 10.84334914 = 3.15665086$
- $15 - 12.49228216 = 2.50771784$
- $15 - 8.581852855 = 6.418147145$
- $8 - 7.804354906 = 0.195645094$
- $15 - 13.89481173 = 1.10518827$
- $21 - 17.43851259 = 3.56148741$
- $23 - 19.32242089 = 3.67757911$
- $25 - 13.15398887 = 11.84601113$
- $13 - 10.37078485 = 2.62921515$
- $19 - 17.23021695 = 1.76978305$
- $20 - 19.6765958 = 0.3234042$
- $20 - 21.23609019 = -1.23609019$
- $21 - 14.38901644 = 6.61098356$
- $19 - 11.71796267 = 7.28203733$
- $19 - 17.56254071 = 1.43745929$
- $20 - 17.79163389 = 2.20836611$
- $20 - 17.6913445 = 2.3086555$
- $21 - 11.74303325 = 9.25696675$
- $18 - 8.184260505 = 9.815739495$
- $17 - 11.8310608 = 5.1689392$
- $13 - 11.27651773 = 1.72348227$
- $12 - 10.94912773 = 1.05087227$
- $12 - 7.230875183 = 4.769124817$

Maka seluruh hasil perhitungan SSR1 ditunjukkan pada tabel 2.

Tabel 2. Nilai piksel hasil SSR1

0.775945942	0.195645094	2.62921515	7.28203733	9.815739495
1.910754589	1.10518827	1.76978305	1.43745929	5.1689392
3.15665086	3.56148741	0.3234042	2.20836611	1.72348227
2.50771784	3.67757911	-1.23609019	2.3086555	1.05087227
6.418147145	11.84601113	6.61098356	9.25696675	4.769124817

Lakukan langkah tersebut sampai ditemukan nilai piksel dari SSR1, SSR2, dan SSR3. Perbedaan antara SSR 1, 2, dan 3 hanya di nilai sigma. Misalnya pada SSR1 nilai sigma adalah 7.5, maka untuk SSR2 dan 3 bisa diubah menjadi 8.5 dan 9.5

Tabel 3. Nilai piksel hasil SSR2

0.776152522	0.159843454	2.57132515	7.18681519	10.75438522
1.908092267	1.02788435	1.76570789	1.29007318	5.0271464
3.15112498	3.45153934	0.1607684	2.05400539	1.57401262
2.72226243	3.55922136	-1.41385445	2.15560069	0.90212415
6.407890182	11.72070832	6.40920685	9.08889689	4.606303703

Tabel 4. Nilai piksel hasil SSR3

0.234330462	-0.801439658	1.30465398	5.78704788	8.780383603
0.868694862	-0.67689767	-0.31450178	-0.81514186	3.65964262
1.75658003	1.32256039	-2.20079255	-0.07256469	0.28581662
0.89265403	1.19591	-3.95998388	0.04239788	-0.34377298
5.300228676	10.14496202	4.75516588	7.74399957	3.843772967

3.2 Implementasi *Multiscale Retinex*

Setelah tiga tahapan *Single Scale Retinex*, langkah selanjutnya adalah melakukan tahapan *Multiscale Retinex* (MSR) yang merupakan generalisasi dari *Single Scale Retinex* (SSR), dengan rumus sebagai berikut:

$$R_{MSR_i} = \sum_{n=1}^N w_n R_{n_i}$$

R_{MSR_i} merupakan keluaran dari *Multiscale Retinex* (MSR) yang masing-masing diberi bobot dan merupakan jumlah dari keluaran SSR. N adalah jumlah keseluruhan dari skala yang digunakan. w_n merupakan bobot dengan skala ke- n yang diasosiasikan. R_{n_i} merupakan keluaran yang diasosiasikan dari SSR dengan skala ke- n . i merupakan kanal warna, yaitu *Red*(R), *Green*(G), dan *Blue*(B) [9].

Guna mendapatkan hasil MSR maka dilakukan perhitungan dengan menjumlahkan nilai piksel hasil perhitungan SSR_1 , SSR_2 , dan SSR_3 lalu dikalikan dengan 1 yang dimana merupakan nilai bobot. Setelah dilakukan perhitungan, kita akan mendapatkan hasil dari nilai *Multiscale Retinex*.

Tabel 5. Hasil penjumlahan SSRn

1.786428926	-0.445951110	6.50519428	20.25590039	29.350508318
4.687541718	1.45617495	3.22098916	1.912390609	13.85572822
8.06435587	8.33558714	-1.71661995	4.18980681	3.58331151
6.12263430	8.43271047	-6.60992852	4.50665407	1.60922343
18.1262660	33.71168147	17.77535629	26.08986320	13.219201487

Perhitungan tersebut menggunakan rumus $MSR = SSR_1 + SSR_2 + SSR_3 \times 1$. Setelah mendapatkan nilai *Multiscale Retinex* (MSR), nilai tersebut harus di normalisasikan terlebih dahulu. Perhitungannya adalah seperti berikut:

$$New\ Value = abs \frac{(value - min)}{(max - min)} \times 255$$

Dari perhitungan MSR, didapatkan nilai:

Nilai terbesar: 33.71168147

Nilai terkecil: -6.60992852

Karena rentang warna adalah 0 – 255 dari terkecil sampai terbesar, maka nilai (33.71168147) adalah 255, dan (-6.60992852) adalah 0. Setelah melakukan normalisasi, hasil yang diperoleh ditunjukkan pada table 6.

Tabel 6. Hasil normalisasi *Multiscale Retinex*

53	38	82	169	227
71	51	62	53	129
92	94	30	68	64
80	95	0	70	51
156	255	154	206	125

3.3 Implementasi Median Filter [10]

Proses *Median Filter* menggunakan nilai piksel *Multiscale Retinex* yang telah di normalisasi, kemudian dilakukan proses untuk mendapatkan nilai tengah.

Tabel 7. Proses *Median Filter*

53	38	82	169	227
71	51	62	53	129
92	94	30	68	64
80	95	0	70	51
156	255	154	206	125

$F(x) = 30, 38, 51, 53, (62), 71, 82, 92, 94$

Maka hasil dari pencarian nilai tengah piksel adalah 62, sehingga nilai tersebut akan menjadi nilai tengah piksel yang baru. Lakukan langkah tersebut sampai ditemukan nilai tengah piksel *Median Filter*.

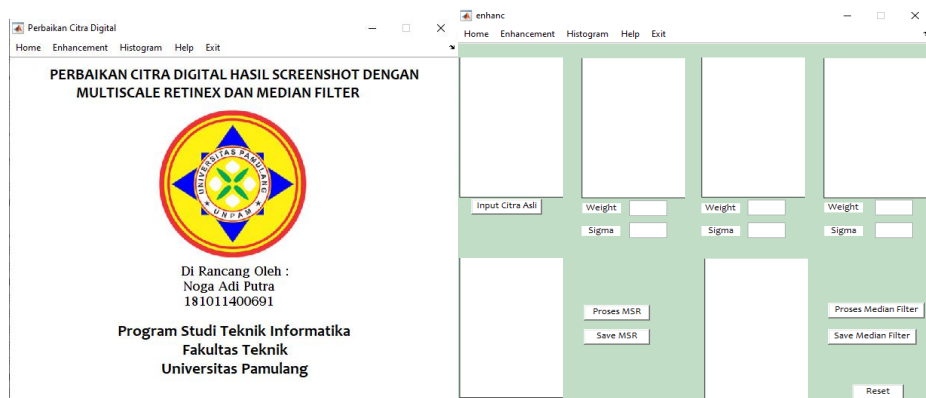
Tabel 8. Hasil *Median Filter*

53	38	82	169	227
71	62	62	68	129
92	71	62	62	64
80	94	94	68	51
156	255	154	206	125

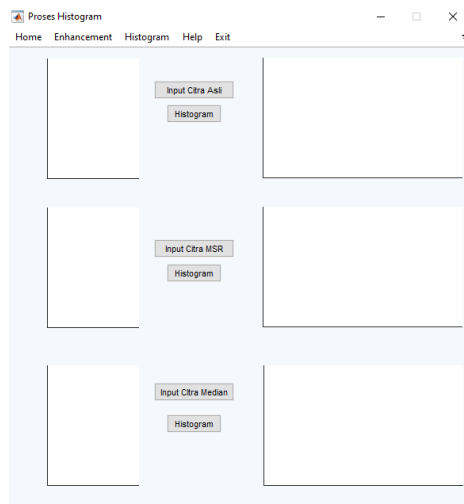
Tabel 8 merupakan hasil dari proses *Median Filter*. Nilai-nilai tengah piksel sebelumnya diganti dengan nilai piksel baru hasil *Median Filter* dengan memasukkan nilainya pada *kernel* tengah.

3.4 Implementasi dan Pengujian

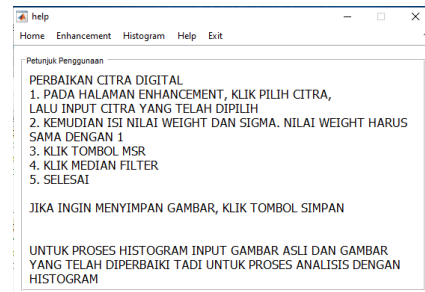
Pada proses implementasi disusun kode program yang merupakan representasi terhadap metode yang diimplementasikan dengan aplikasi Matlab 2015a dan Bahasa pemrograman Java guna membangun antarmuka.



Gambar 3. Antarmuka home dan antarmuka halaman utama



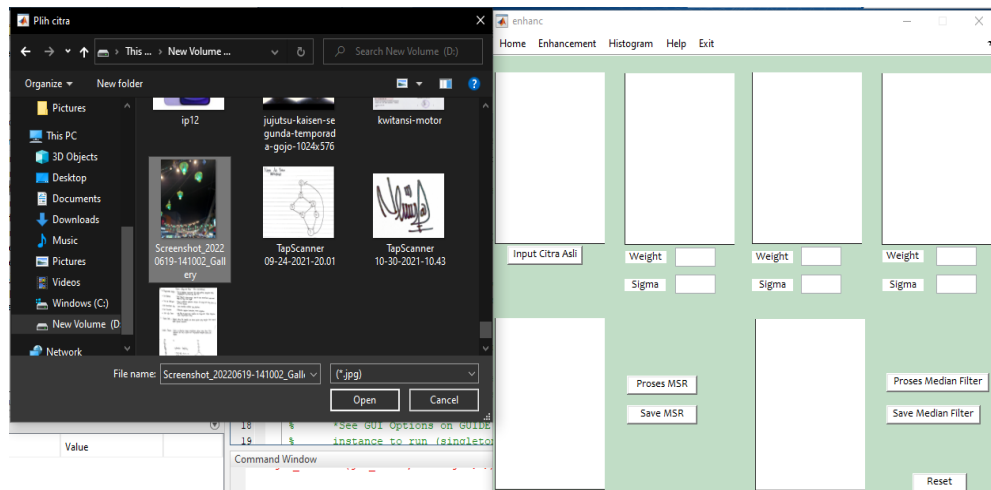
Gambar 2. Desain antarmuka Histogram



Gambar 3. Desain antarmuka Help

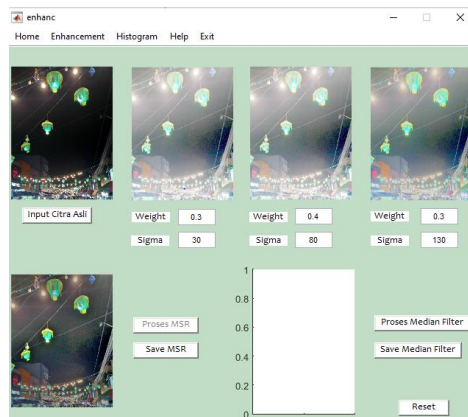
Setelah proses implementasi dari setiap metode selesai dilakukan, maka langkah selanjutnya adalah dengan melakukan pengujian sistem untuk mendapatkan citra yang lebih baik, terutama pada kecerahannya. Pada pengujian sistem ini diperlukan beberapa kriteria, diantaranya adalah sebagai berikut :

- 1) Citra yang digunakan berformat *.jpg.
- 2) Citra hasil *Multiscale Retinex* dan *Median Filter* dapat disimpan.
- 3) Citra awal maupun citra hasil dapat dianalisis dengan melihat *histogram* citra masing-masing.



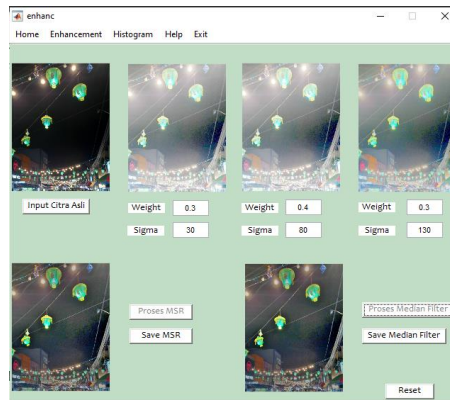
Gambar 4. Proses input citra

Gambar 6 merupakan proses saat sedang memasukkan citra kedalam sistem, yang mana pengujian untuk input citra berhasil.



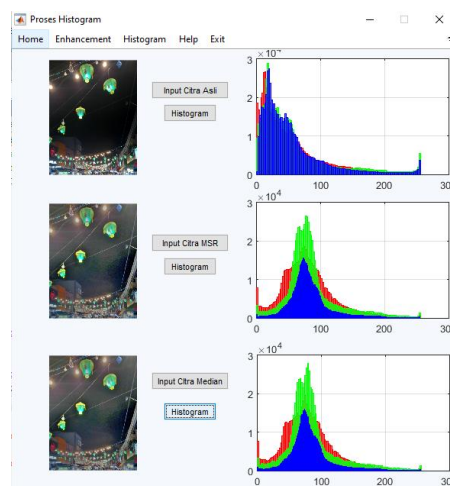
Gambar 5. Proses Multiscale Retinex

Gambar 7 merupakan proses pengolahan dengan *Multiscale Retinex*, yang mana pengujian untuk proses pengolahan citra dengan *Multiscale Retinex* berhasil.



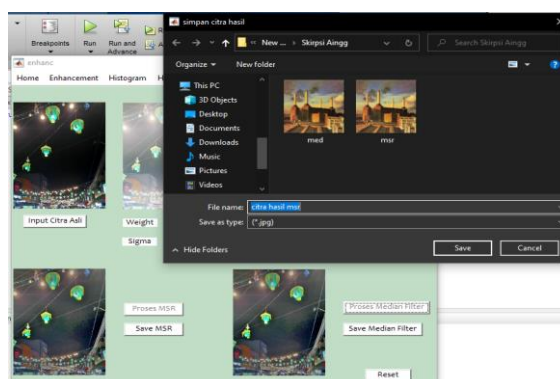
Gambar 6. Proses *Median Filter*

Gambar diatas merupakan proses pengolahan dengan *Median Filter*, yang mana pengujian untuk proses pengolahan citra dengan *Median Filter* berhasil.



Gambar 7. Hasil proses histogram

Gambar 9 merupakan proses analisa dengan *histogram*, yang mana pengujian untuk proses analisa dengan *histogram* berhasil.



Gambar 8. Proses simpan citra

Gambar diatas merupakan proses simpan citra, yang mana pengujian untuk proses simpan citra berhasil.

4. PENUTUP

4.1 Kesimpulan

1. Dari hasil penerapan metode *Multiscale Retinex* dan *Median Filter* yang telah dilakukan dapat memperbaiki kualitas dari sebuah citra digital. Metode *Multiscale Retinex* dapat mencerahkan citra digital dan *Median Filter* dapat mengurangi derau pada citra digital yang membuat kualitas citra digital menjadi lebih jelas.
2. Penggunaan metode *Multiscale Retinex* dan *Median Filter* menghasilkan kerja sama yang sangat baik dalam perbaikan citra digital, dimana proses *Multiscale Retinex* dan *Median Filter* dapat membuat citra digital menjadi lebih baik dan menunjukkan nilai intensitas piksel mendekati nilai 250, dimana jika sebelum proses perbaikan nilai intensitas piksel lebih cenderung ke angka 0.
3. Nilai sigma sangat berpengaruh dalam proses perbaikan citra digital, dimana semakin besar nilai sigma maka citra yang dihasilkan akan lebih terang, namun semakin terang sebuah citra yang sebelumnya gelap akan membuat sebuah citra memiliki banyak derau, sehingga proses *Median Filter* disini berguna untuk mengurangi derau tersebut.

4.2 Saran

1. Sistem ini hanya menggunakan file dengan format *.jpg, untuk pengembangan lebih lanjut dapat ditambahkan jenis format gambar lainnya
2. Penelitian ini hanya sampai pada metode *Multiscale Retinex*, disarankan menggunakan metode *Retinex* yang lainnya untuk dapat membandingkan penggunaan metode untuk perbaikan citra mana yang lebih baik.
3. Pengurangan derau yang peneliti lakukan disini hanya menggunakan *Median Filter*, disarankan menggunakan metode untuk pengurangan derau yang lainnya untuk perbandingan mana yang lebih baik dalam perbaikan citra digital.

DAFTAR PUSTAKA

- [1] H. S. Wahyudi and M. P. Sukmasari, "TEKNOLOGI DAN KEHIDUPAN MASYARAKAT," *J. Anal. Sociol.*, 2018.
- [2] Sulistiyanti, S. Ratna, Setyawan, F. X. Arinto, Komarudin, and Muhamad, *Pengolahan Citra, Dasar dan Contoh Penerapannya*. Yogyakarta: Teknosain, 2017.
- [3] S. Syamsuddin, "Aplikasi Peningkatan Kualitas Citra Menggunakan Metode Median Filtering Untuk Menghilangkan Noise," *SISITI Semin. Ilm. Sist. Inf. dan Teknol. Inf.*, vol. 8, no. 1, pp. 227–236, 2019.
- [4] E. Aribowo *et al.*, "Implementasi Metode Retinex Untuk Pencerahan Citra," *J. Inform.*, vol. 3, no. 2, pp. 323–330, 2009.

- [5] R. A. Sholihin, "Implementasi Median Filter dan Metode Histogram Dalam Perbaikan Citra," 2013.
- [6] Jogiyanto, "Analisis & Desain," *Andi*, vol. 2, no. 5, 2009.
- [7] A. B. Petro, C. Sbert, and J.-M. Morel, "Multiscale Retinex," *Image Process. Line*, vol. 4, pp. 71–88, 2017.
- [8] D. Supriadi and P. Novantara, "IMPLEMENTASI METODE MULTISCALE RETINEX UNTUK IMAGE ENHANCEMENT MENGGUNAKAN," 2017.
- [9] S. Aripin, G. L. Ginting, and N. Silalahi, "Penerapan Metode Retinex Untuk Meningkatkan Kecerahan Citra Pada Hasil Screenshot," *Media Inform. Budidarma*, vol. 1, no. 1, pp. 24–27, 2017.