

**PERBANDINGAN METODE *CONTRAST STRETCHING*
DAN METODE *RETINEX* UNTUK PENINGKATAN
KECERAHAN CITRA DIGITAL**

SKRIPSI

DELLA DARMAWAN LUBIS

0701162041



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**PERBANDINGAN METODE *CONTRAST STRETCHING*
DAN METODE *RETINEX* UNTUK PENINGKATAN
KECERAHAN CITRA DIGITAL**

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

DELLA DARMAWAN LUBIS

0701162041



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp :-

Kepada Yth.,

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama	:	Della Darmawan Lubis
Nomor Induk Mahasiswa	:	0701162041
Program Studi	:	Ilmu Komputer
Judul	:	Perbandingan Metode <i>Contrast Stretching</i> dan Metode <i>Retinex</i> Untuk Peningkatan Citra Digital

dapat disetujui untuk segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terimakasih.

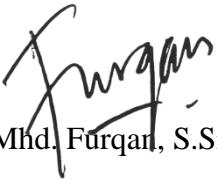
Medan, 26 Maret 2021 M

12 Sya'ban 1442 H

Komisi Pembimbing,

Pembimbing Skripsi I,

Pembimbing Skripsi II,


Dr. Mhd. Furqan, S.Si, M.Comp.Sc
NIP. 198008062006041003


Muhammad Ikhsan, S.T, M.Kom.
NIP. 198503162015031003

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama : Della Darmawan Lubis
Nomor Induk Mahasiswa : 0701162041
Program Studi : Ilmu Komputer
Judul : Perbandingan Metode *Contrast Stretching*
dan Metode *Retinex* Untuk Peningkatan
Citra Digital

menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 26 Maret 2021



Della Darmawan Lubis

NIM. 0701162041



PENGESAHAN SKRIPSI

Nomor: B.100/ST/ST.V.2/PP.01.1/05/2021

Judul : Perbandingan Metode *Contrast Stretching* dan Metode *Retinex* Untuk Peningkatan Kecerahan Citra Digital
Nama : Della Darmawan Lubis
Nomor Induk Mahasiswa : 0701162041
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Pengaji Skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Jum'at, 26 Maret 2021
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,
Ketua,

Ilka Zufria, M.Kom
NIP. 198506042015031006

Dewan Pengaji,

Pengaji I,

Dr. Mhd. Furqán, S.Si., M.Comp.Sc.
NIP. 198008062006041003

Pengaji II,

Muhammad Ikhsan, ST., M.Kom
NIP. 198304152011011008

Pengaji III,

Syahnan, S.Kom, M.Kom
NIB. 1100000108

Pengaji IV,

Abdul Halim Hasugian, M.Kom
NIB. 1100000113

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd. Syahnan, MA
NIP. 196609051991031002

ABSTRAK

Peristiwa yang terjadi pada kehidupan kita sehari – hari sering kali kitaabadikan dalam bentuk gambar, tulisan, audio dan video sebagai bahan dokumentasi untuk kita sendiri, keluarga, teman maupun orang lain. Gambar atau foto adalah hal yang paling sering kita ambil menggunakan kamera untuk menangkap kejadian yang kita alami. Pada saat ini, teknologi sudah berkembang pesat dan mempengaruhi kamera yang kita miliki mempunyai kemampuan yang beragam, Tetapi kemajuan tersebut tidak menutup kemungkinan bahwa gambar yang dihasilkan dari proses digitalisasi tersebut mengalami permasalahan dalam hal kualitas. Metode *Contrast Stretching* dan Metode *Retinex* adalah bagian dari metode perbaikan kualitas citra yang dapat meningkatkan tingkat kecerahan pada gambar. Kedua metode dibandingkan untuk mencari yang terbaik. Penelitian ini mencoba melakukan perbandingan kepada kedua metode tersebut dengan menggunakan citra RGB dan mendapatkan hasil bahwa Metode *Retinex* sangat baik dalam menangani citra yang berkondisi gelap dan kontras tinggi, sedangkan Metode *Contrast Stretching* sangat baik dalam citra yang memiliki kontras rendah.

Kata Kunci: *Image, Noise, Contrast Stretching, Retinex.*

ABSTRACT

Events that occur in our daily lives are often immortalized in the form of images, writing, audio and video as documentation for ourselves, family, friends and others. Pictures or photos are the things we most often take using a camera to capture the events we experience. At this time, technology has developed rapidly and affects the cameras that we have diverse capabilities, but these advances do not rule out the possibility that images resulting from the digitization process are experiencing problems in terms of quality. The Contrast Stretching method and the Retinex method are part of the image quality improvement method that can increase the brightness level in the image. The two methods are compared to looking their best. This study tried to compare the two methods using RGB imagery and found that the Retinex Method is very good at handling dark and high contrast images, while the Contrast Stretching method is very good in images that have low contrast.

Keywords: Image, Noise, Contrast Stretching, Retinex.

KATA PENGANTAR

Syukur Alhamdulillah saya ucapan kepada Allah SWT atas segala limpahan anugerah dan rahmat-Nya sehingga penelitian skripsi ini dapat diselesaikan sebagaimana yang diharapkan. Tidak lupa shalawat berangkaikan salam kepada nabi Muhammad SAW yang merupakan suri tauladan bagi kehidupan manusia menuju jalan yang diridhoi Allah SWT. Skripsi yang berjudul “Perbandingan Metode *Contrast Stretching* dan Metode *Retinex* untuk Peningkatan Kecerahan Citra Digital” dan diajukan untuk memenuhi salah satu persyaratan untuk memperoleh gelar sarjana S1 Fakultas Sains dan Teknologi Jurusan Ilmu Komputer UIN Sumatera Utara.

Peneliti menyadari bahwa skripsi ini dapat diselesaikan berkat dukungan dan bantuan dari berbagai pihak. Oleh karena itu peneliti berterima kasih kepada semua pihak yang secara langsung maupun tidak langsung dalam memberikan kontribusi untuk menyelesaikan skripsi ini. Peneliti menyampaikan ucapan terima kasih kepada:

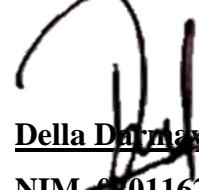
1. Bapak Prof. Dr. H Syahrin Harahap, M.A selaku rektor UIN Sumatera Utara yang telah memberikan fasilitas yang baik.
2. Bapak Dr. Mhd Syahnan, M.A selaku dekan Fakultas Sains dan Teknologi UIN Sumatera Utara yang telah memberikan kesempatan untuk peneliti menimba ilmu di jurusan Ilmu Komputer.
3. Bapak Ilka Zufria, M.Kom selaku ketua jurusan Ilmu Komputer yang telah menyetujui judul ini.
4. Bapak Rakhmat Kurniawan R, S.T., M.Kom selaku sekretaris jurusan Ilmu Komputer yang telah menyetujui judul ini.
5. Ibu Sriani, M.Kom selaku pembimbing akademik yang telah memberikan bimbingan dan arahan sehingga peneliti dapat menjalani studi akademik di UIN Sumatera Utara dengan baik.
6. Bapak Dr. Mhd. Furqan, S.Si., M.Comp.Sc. selaku dosen pembimbing I yang telah banyak memberikan arahan serta bimbingan kepada peneliti.

7. Bapak Muhammad Ikhsan, S.T., M.Kom selaku dosen pembimbing II yang telah banyak memberikan waktu nya untuk membimbing peneliti.
8. Teristimewa peneliti sampaikan terima kasih dengan setulus hati kepada kedua orang tua tercinta, ayah Dedy Darmawan Lubis dan mama Wan Lafitah Hanum yang sampai detik ini telah berjuang membesarkan dan mendidik peneliti, berkat kasih sayangnya dan pengorbanan yang tak terhingga sehingga peneliti dapat menyelesaikan studi ke bangku perkuliahan.
9. Teristimewa kepada adik-adik tercinta M. Ade Darmawan Lubis, Dea Darmawan Lubis, Dwi Darmawan Lubis dan Dewa Darmawan Lubis yang selama ini turut memberikan doa dan dukungan selama perkuliahan dan penyusunan skripsi ini.

Semoga Allah SWT membalas semua kebaikan yang telah diberikan Bapak/Ibu serta Saudara/i, kiranya kita semua tetap berada dalam lindungan-Nya demi penyelesaian skripsi ini. Semoga proposal ini bermanfaat dalam memperkaya khazanah ilmu pengetahuan kita. Aamiin..

Medan, Maret 2021

Penyusun,



Della Darmawan Lubis

NIM. 0701162041

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Masalah	4
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Citra	5
2.2 Citra Digital	6
2.2.1 Elemen Citra Digital	7
2.2.2 Jenis Citra Digital	9
2.2.3 Format File Citra Digital	10
2.3 Pengolahan Citra	12
2.4 Perbaikan Kualitas Citra (<i>Image Enhancement</i>)	16
2.5 Metode <i>Retinex</i>	16
2.5.1 <i>Single-Scale Retinex</i> (SSR)	17
2.5.2 <i>Multi-Scale Retinex</i> (MSR)	18
2.5.3 <i>Multiscale Retinex with Color Restoration</i> (MSRCR) ...	19
2.5.4 <i>Multiscale Retinex with Chromaticity Preservation</i> (MSRCP).....	20
2.6 Metode <i>Contrast Stretching</i>	21

2.7 Histogram	24
2.8 Pengukur Kualitas Citra	25
2.8.1 PSNR dan MSE	25
2.9 <i>Flowchart</i>	26
2.10 Python	29
2.11 PYQT	31
2.12 OpenCV	33
2.13 Numpy	35
2.14 Matplotlib	35
2.15 Kajian Terdahulu	35
BAB III METODOLOGI PENELITIAN	40
3.1 Waktu Penelitian	40
3.2 Bahan Penelitian	40
3.3 Alat Penelitian	40
3.3.1 Perangkat Keras	40
3.3.2 Perangkat Lunak	41
3.4 Teknik Pengumpulan Data	41
3.5 Analisis Kebutuhan	41
3.5.1 Kebutuhan Functional	41
3.5.2 Kebutuhan Non-Functional	42
3.6 Kerangka Penelitian	43
3.7 Perancangan Sistem Penelitian	45
BAB IV IMPLEMENTASI DAN PENGUJIAN	47
4.1 Implementasi Metode	47
4.1.1 <i>Contrast Stretching</i>	48
4.1.2 <i>Retinex</i>	51
4.1.2.1 SSR	51
4.1.2.2 MSR	59
4.1.2.3 MSRCR	62
4.1.2.4 MSRCP	66
4.1.3 Implementasi Histogram	70

4.1.4 Implementasi MSE dan PSNR	70
4.1.5 Rancangan Interface Program Aplikasi	71
4.1.5.1 Rancangan Interface Loading Screen	72
4.1.5.2 Rancangan Interface Home	72
4.1.5.3 Rancangan Interface Work	73
4.1.5.4 Rancangan Interface About	75
4.2 Pengujian Sistem	76
4.2.1 Interface Loading Screen	76
4.2.2 Interface Halaman Home	77
4.2.3 Interface Halaman Work	77
4.2.4 Interface Halaman About	78
4.2.5 Pengujian Program Aplikasi	79
BAB V KESIMPULAN DAN SARAN	82
5.1 Kesimpulan	82
5.2 Saran	83
DAFTAR PUSTAKA	84
LAMPIRAN – LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Tahap – Tahap Pengolahan Citra	14
2.2	Gambaran Umum <i>Retinex</i>	17
2.3	Gambaran Proses <i>Contrast Stretching</i>	21
2.4	Proses Transformasi <i>Contrast Stretching</i>	22
2.5	Histogram Citra	25
2.6	Tampilan Qt Designer	32
3.1	<i>Flowchart</i> Sistem	45
4.1	Gambar Sampel RGB 5x5 Piksel	47
4.2	Gambar Tabel Data Citra Sampel RGB	47
4.3	<i>Flowchart Contrast Stretching</i>	48
4.4	Data Citra Hasil <i>Contrast Stretching</i>	50
4.5	Hasil Enhance Metode <i>Contrast Stretching</i>	51
4.6	<i>Flowchart Single Scale Retinex</i>	52
4.7	Contoh Perhitungan Kernel Fungsi Gaussian	53
4.8	Hasil Contoh Kernel Gaussian	53
4.9	Hasil Contoh Kernel Gaussian Baru	54
4.10	Contoh Citra Awal (3x3 piksel) 3 bit warna	54
4.11	Hasil Konvolusi	55
4.12	Data Citra Hasil SSR	58
4.13	Hasil Enhance Metode <i>Retinex</i> (SSR)	58
4.14	<i>Flowchart Multiscale Retinex</i>	59
4.15	Data Citra Hasil MSR sigma [15 ,80, 250]	60
4.16	Hasil Enhance Metode <i>Retinex</i> (MSR)	61
4.17	Hasil Enhance Metode <i>Retinex</i> (MSR) 20 x 20 Piksel	62
4.18	<i>Flowchart Multiscale Retinex with Color Restoration</i>	63
4.19	Algoritma MSRCR	63
4.20	Hasil Metode MSR	64

4.21	Hasil MSRCR pada kanal warna Red (R)	64
4.22	Hasil MSRCR pada kanal warna Green (G)	64
4.23	Hasil MSRCR pada kanal warna Blue (B)	64
4.24	Data Citra Hasil akhir MSRCR	65
4.25	Hasil Enhance Metode <i>Retinex</i> (MSRCR)	65
4.26	<i>Flowchart Multiscale Retinex with Chromaticity Preservation</i> ..	67
4.27	Algoritma MSRCP	68
4.28	Data Citra Hasil akhir MSRCP	68
4.29	Hasil Enhance Metode <i>Retinex</i> (MSRCP)	68
4.30	Contoh Perhitungan MSE dan PSNR	71
4.31	Rancangan Tampilan Loading Screen	72
4.32	Rancangan Tampilan Halaman Home	72
4.33	Rancangan Tampilan Interface Halaman Work 1	73
4.34	Rancangan Tampilan Interface Halaman Work 2	73
4.35	Rancangan Tampilan Halaman About	75
4.36	Interface Loading Screen	76
4.37	Interface Halaman Home	77
4.38	Interface Halaman Work 1	77
4.39	Interface Halaman Work 2	78
4.40	Interface Halaman About	78
4.41	Tampilan Window Input Citra	79
4.42	Tampilan File Citra dan Histogram	79
4.43	Tampilan Pengujian Metode (1)	80
4.44	Tampilan Pengujian Metode (2)	80
4.45	Tampilan Pengujian Metode (3)	81

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1	Kajian Terdahulu	28
2.2	Symbol – Symbol <i>Flowchart</i>	36
4.1	Tabel Perhitungan Nilai Percentile	50
4.2	Tabel Contoh (x,y)	53
4.3	Hasil log I _i (x,y)	56
4.4	Hasil Konvolusi	56
4.5	log (G(x,y))	56
4.6	log (I(x,y)) – log (G(x,y))	57
4.7	Hasil SSR, kanal (Red), Sigma = 250	57
4.8	Histogram	70
4.9	Contoh MSE dan PSNR	70
4.10	MSE dan PSNR	71

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Hasil Pengujian Aplikasi
2.	Listing Program
3.	Daftar Riwayat Hidup

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peristiwa yang terjadi pada kehidupan kita sehari – hari sering kali kita abadikan dalam bentuk gambar, tulisan, audio dan video dengan maksud sebagai bahan pertinggal atau dokumentasi bagi kita sendiri, keluarga, teman maupun orang lain. Gambar atau foto adalah hal yang paling sering kita ambil untuk menangkap kejadian yang kita alami, bisa saja moment bersama keluarga, bersama teman atau hanya gambar diri sendiri. Gambar atau foto tersebut diambil dengan kamera yang kita miliki, mau itu kamera *handphone*, kamera digital maupun kamera yang memiliki lensa beresolusi tinggi.

Pada saat ini, teknologi sudah berkembang sangat pesat dan mempengaruhi keadaan alat elektronik yang kita miliki mempunyai kemampuan yang beragam, bahkan dalam hal kamera *handphone* saja sudah terdapat banyak fitur – fitur yang dikembangkan untuk memperindah moment yang ingin kita ambil. Sebagai contoh dari perkembangan fitur tersebut adalah filter warna, tema, dan juga menu percantik. Tetapi kemajuan tersebut tidak menutup kemungkinan gambar atau foto yang dihasilkan dari proses digitalisasi tersebut menghasilkan citra mengandung derau (*Noise*), citra terlalu terang atau gelap, citra kurang tajam, kabur (*Blur*), cacat saat akuisisi citra, lensa *object blurring* atau *background blurring*, objek bergerak atau kamera bergerak, *motion blurring* dan distorsi geometrik disebabkan oleh lensa atau sudut pengambilan. Untuk itu, diperlukanlah perbaikan kualitas gambar. Seperti Firman Allah SWT yang terdapat dalam Al-Qur'an Surat Al-Baqarah Ayat 11:

وَإِذَا قِيلَ لَهُمْ لَا تُفْسِدُوا فِي الْأَرْضِ قَالُوا إِنَّمَا نَحْنُ مُصْلِحُونَ

Artinya: "Dan bila dikatakan kepada mereka: "Janganlah kamu membuat kerusakan di muka bumi". Mereka menjawab: "Sesungguhnya kami orang - orang yang mengadakan perbaikan" (QS. Al-Baqarah:11).

Perbaikan kualitas citra (*Image Enhancement*) merupakan salah satu proses awal dalam pengolahan citra (*Image Processing*). Dengan pengolahan citra, citra dapat diubah menjadi citra yang sesuai keinginan. Adapun proses-proses yang termasuk dalam perbaikan kualitas citra adalah perubahan kecerahan gambar (*Image brightness*), peregangan kontras (*Contrast Stretching*), perubahan histogram (*Histogram equalization*), pelembutan citra (*Image smoothing*), penajaman tepi (*Sharpening edge*), pewarnaan semu (*Pseudocolouring*), perubahan geometric (Munir, 2004).

Tujuan dari perbaikan kualitas gambar adalah untuk memperoleh citra yang lebih sesuai digunakan untuk aplikasi lebih lanjut, dan meningkatkan kualitas dengan menghormati persepsi visual manusia atau *Human Vision System* (HVS) bertanggung jawab atas keteguhan warna dan jangkauan dinamis, memiliki toleran pada variasi dalam kondisi pencahayaan yang kuat ataupun lemah. Pencahayaan mempengaruhi kecerahan dan kontras yang dihasilkan dalam sebuah gambar, karena hal itulah pencahayaan yang dibutuhkan harus cukup dan sesuai. Gambar yang memiliki kecerahan dan kontras yang terlalu terang ataupun terlalu gelap membutuhkan perbaikan ataupun peningkatan kecerahan citra digital.

Penulis mengambil judul Perbandingan Metode *Contrast Stretching* Dan Metode *Retinex* Untuk Peningkatan Kecerahan Citra Digital. Dimana, metode *Contrast Stretching* merupakan salah satu metode yang sangat berguna dalam meningkatkan kontras terutama terhadap citra yang memiliki kontras rendah (Putra, 2010), sedangkan metode *Retinex* merupakan metode yang berusaha untuk mempertahankan ketetapan warna (*color constancy*) dimana warna suatu objek yang dilihat memiliki warna yang relatif sama meskipun dalam keadaan pencahayaan yang berbeda. Seperti suatu objek yang dilihat tetap memiliki warna yang sama meskipun dilihat saat pagi yang cerah maupun sore yang kemerah. (Murinto, 2009).

Penelitian ini diharapkan dapat menjadi pengembangan pembelajaran dari metode *Contrast Stretching* dan metode *Retinex* untuk peningkatan kecerahan citra dalam perbaikan kualitas citra (*Image Enhancement*), dan bertujuan untuk mengetahui cara kerja, kelebihan dan kekurangan dari kedua metode yaitu metode

Contrast Stretching dan metode *Retinex*, juga membandingkan hasil output dari kedua metode tersebut untuk mengetahui metode mana yang lebih baik dalam menangani gambar dengan kondisi terlalu gelap maupun terlalu terang tanpa membatasi jumlah pixel gambar dengan indikator nilai PSNR, MSE dan Histogram gambar hasil tersebut. Penelitian ini menggunakan aplikasi desktop sebagai alat uji atau tempat implementasi metode *Contrast Stretching* dan metode *Retinex* yang dibangun menggunakan Bahasa Pemrograman Python dan Library Opencv, numpy, Matplotlib, PYQT. Adapun data yang digunakan sebagai data uji adalah gambar warna (citra warna RGB) berformat .JPG dan .PNG.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana proses peningkatan kecerahan citra digital dengan metode *Contrast Stretching* dan metode *Retinex*?
2. Apa saja kelebihan dan kekurangan dari metode *Contrast Stretching* dan metode *Retinex*?
3. Bagaimana perbandingan hasil dari peningkatan kecerahan citra digital metode *Contrast Stretching* dan metode *Retinex*?

1.3 Batasan Masalah

Agar penelitian ini tidak meluas, maka penulis mencantumkan batasan masalah sebagai berikut:

1. Data gambar yang digunakan adalah gambar warna (citra warna RGB).
2. Nilai pixel dari data gambar tidak ditentukan.
3. Data gambar yang digunakan hanya berupa gambar atau citra .JPG dan .PNG.
4. Perbandingan dilakukan melihat hasil warna dan size (jumlah pixel) dari citra hasil.
5. Metode yang digunakan adalah Metode *Contrast Stretching* dan Metode *Retinex*.

6. Bahasa pemrograman yang digunakan dalam membangun aplikasi desktop adalah Bahasa Pemrograman Python dengan *library* Opencv.
7. Menghitung nilai PSNR dan MSE dari citra hasil keluaran dari metode *Contrast Stretching* dan metode *Retinex*.
8. Menampilkan hasil histogram dari citra keluaran metode *Contrast Stretching* dan metode *Retinex*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui proses peningkatan kecerahan citra digital dengan metode *Contrast Stretching* dan metode *Retinex*.
2. Mengetahui kelebihan dan kekurangan metode *Contrast Stretching* dan metode *Retinex*.
3. Memberikan rekomendasi metode mana yang lebih baik antara metode *Contrast Stretching* dan metode *Retinex* dalam peningkatan kecerahan citra digital.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penulisan skripsi ini adalah sebagai pengembangan penelitian perbaikan kualitas citra dari metode *Contrast Stretching* dan metode *Retinex* untuk peningkatan kecerahan citra digital serta sebagai bahan rekomendasi pemilihan metode untuk aplikasi yang lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek (Sutoyo, 2009). Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Citra merupakan output alat perekaman, seperti kamera yang bersifat analog maupun digital. Citra analog berupa sinyal-sinyal video seperti gambar pada televisi, sedangkan citra digital dapat langsung disimpan pada suatu media penyimpanan (Dinata, 2014). Citra (image) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang 2 dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat - alat optik, misal mata manusia, kamera, scanner, sensor satelit, dan sebagainya, kemudian direkam.

Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat:

1. Optik berupa foto
2. Analog berupa sinyal video seperti gambar pada monitor televisi
3. digital yang dapat langsung disimpan pada suatu pita magnetic

Citra juga dapat dikelompokkan menjadi 2 yaitu:

1. Citra tampak, misalnya foto, gambar, lukisan, apa yang nampak di layer monitor/televisi, hologram, dan lainnya.
2. Citra tidak tampak, misalnya data foto / gambar dalam file, citra yang direpresentasikan dalam fungsi matematis (Munir,2004).

Citra juga dapat menjadi gambar atau citra diskrit melalui proses sampling.

Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel.

Contohnya adalah gambar/titik diskrit pada baris (n) dan kolom (m) disebut dengan piksel [n.m].

Citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y dan nilai amplitudo f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital.

2.2 Citra Digital

Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu (Putra, 2010). Citra yang disimpan dalam memori komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel tersebut. Citra tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat (x,y) dan amplitudo $f(x,y)$. Koordinat (x,y) menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitudo $f(x,y)$ menunjukkan nilai intensitas warna citra.

Citra digital adalah suatu citra $f(x,y)$ yang memiliki koordinat spatial, dan tingkat kecerahan yang diskrit. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Fungsi $f(x,y)$ dapat dilihat sebagai fungsi dengan dua unsur. Unsur yang pertama adalah kekuatan sumber cahaya yang melingkupi pandangan kita terhadap objek (*illumination*), sedangkan unsur yang kedua adalah besarnya cahaya yang direfleksikan oleh objek ke dalam pandangan kita (*reflectance components*). Keduanya dapat dituliskan sebagai fungsi $I(x,y)$ dan $r(x,y)$ yang digabungkan sebagai perkalian fungsi untuk membentuk fungsi $f(x,y)$. Adapun fungsi tersebut dapat dituliskan sebagai berikut:

$$f(x,y) = i(x,y) * r(x,y) \text{ dimana } 0 < i(x,y) < \infty \text{ dan } 0 < r(x,y) < 1 \dots \dots \dots (2.1)$$

Suatu citra agar bisa diolah komputer digital, maka harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Digitalisasi merupakan representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit. Suatu citra digital berukuran N

(tinggi) x M (lebar) dapat dinyatakan sebagai matriks dengan ukuran N x M (Fatmawati, 2011).

2.2.1 Elemen Citra Digital

Citra digital mengandung sejumlah elemen – elemen dasar. Elemen – elemen dasar tersebut dimanipulasi dalam pengolahan citra dan dieksplorasi lebih lanjut dalam computer vision. Elemen – elemen dasar yang penting antaranya adalah:

1. Kecerahan (*brightness*)

Kecerahan adalah kata lain untuk intensitas cahaya. Kecerahan pada sebuah titik (pixel) di dalam citra bukanlah intensitas yang riil, tetapi sebenarnya adalah intensitas rata – rata dari suatu area yang melingkapinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkat kecerahan (*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi dengan jangkauan sebesar 10^{10} .

2. Kontras (*contras*)

Kontras menyatakan sebaran terang (*lighteness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur (*contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada pixel – pixel yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendekripsi tepi – tepi (edge) objek di dalam citra.

4. Warna (*color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang (λ) yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (violet) mempunyai panjang gelombang paling rendah.

Warna – warna yang diterima oleh mata (sistem visual manusia) merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (R), *green* (G), dan *blue* (B). Persepsi sistem visual manusia terhadap warna sangat relatif sebab dipengaruhi oleh banyak kriteria, salah satunya disebabkan oleh adaptasi yang menimbulkan distorsi. Misalnya bercak abu – abu di sekitar warna hijau akan tampak keungu – unguan (distorsi terhadap ruang), atau jika mata melihat warna hijau lalu langsung dengan cepat melihat warna abu – abu, maka mata menangkap kesan warna abu – abu tersebut sebagai warna ungu (distorsi terhadap waktu).

5. Bentuk (*shape*)

Shape adalah properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa *shape* merupakan properti intrinsik utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuk ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolaha dan segmentasi citra. Salah satu tantangan utama pada computer vision adalah merepresentasikan bentuk, atau aspek – aspek penting dari bentuk.

6. Tekstur (*textur*)

Textur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan pixel – pixel yang bertetangga. Jadi, tekstur tidak dapat didefinisikan untuk pixel. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap pixel, melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipresepsi. Sebagai contoh, jika kita mengamati citra lantai bertubin dari jarak jauh, maka kita mengamati bahwa tekstur terbentuk oleh penempatan ubin – ubin secara keseluruhan, bukan dari presepsi pola di dalam ubin itu sendiri. Tetapi, jika kita mengamati citra yang sama dari jarak yang dekan, maka hanya beberapa ubin yang tampak dalam bidang pengamatan sehingga kita

mempresepikan bahwa tekstur terbentuk oleh penempatan pola – pola rinci yang menyusun tiap ubin (Munir, 2004).

2.2.2 Jenis Citra Digital

Nilai suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum, jangkauan yang digunakan berbeda - beda tergantung dari jenis warnanya. Secara umum jangkauannya adalah 0 - 255. Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya. Adapun jenis – jenis citra digital adalah sebagai berikut:

- 1. Citra Grayscale**

Citra *grayscale* adalah jenis citra yang hanya memiliki satu channel warna. Hal ini menyebabkan nilai yang ditampilkan pada jenis citra ini hanyalah nilai intensitas atau lebih sering disebut *grayscale*. Sebenarnya, citra jenis ini sudah sering teman-teman lihat, tetapi istilah yang sering digunakan adalah istilah ‘*Black and White*’. Padahal, secara istilah citra *Black and White* lebih tepat mengacu pada Citra biner.

Untuk mengubah citra berwarna menjadi citra *grayscale*, ada beberapa rumus yang dapat kita gunakan yaitu, $y = \frac{1}{3}(R + G + B)$ atau $y = (0.2989*R + 0.5870*G + 0.1140*B)$

- 2. Citra Berwarna dan Citra Berwarna dengan Transparansi**

Citra berwarna adalah jenis citra yang memiliki tiga channel warna. Beberapa representasi warna dalam citra jenis ini adalah citra RGB (*Red, Green, dan Blue*), HSV (*Hue, Saturation, Value*), YCbCr (*Luma, Chroma blue, Chrome Red*), dan Lab ($L^* a^* b^*$). Jenis citra berwarna yang paling sering digunakan adalah jenis RGB.

Pada citra RGB, setiap kanal memiliki 256 kemungkinan nilai yang berada pada rentang 0-255. Pada kanal merah, warna merah sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0. Pada kanal hijau, warna hijau sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0. Begitu juga pada kanal biru, warna biru sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0.

Seiring perkembangan teknologi dan desain, muncul fitur baru dalam citra berwarna, yaitu fitur transparansi dalam citra berwarna. Citra berwarna dengan transparansi memiliki bagian yang transparan, biasanya pada bagian *background*. Umumnya citra jenis ini dibentuk menggunakan komponen RGB dan Alpha (A), dan dimodelkan ke dalam ruang warna RGBA. Citra jenis ini memiliki 4 channel warna, mengingat kita memiliki satu channel tambahan untuk Alpha.

3. Citra Biner

Citra biner atau citra hitam putih. Citra ini hanya memiliki satu channel warna dan hanya memiliki dua kemungkinan nilai, yaitu 0 dan 1. Jenis citra ini adalah jenis citra yang sering digunakan untuk proses segmentasi citra.

Untuk membuat citra jenis ini dari citra berwarna, kita harus mengubah sebuah citra berwarna menjadi citra *grayscale* terlebih dahulu. Kemudian, kita menentukan nilai *threshold* (batas). Jika nilai pada piksel sama dengan atau melebihi nilai *threshold*, maka nilai piksel tersebut dikonversi menjadi 1, dan jika sebaliknya, dikonversi menjadi 0, seperti yang dinyatakan dalam rumus berikut:

$$Y^I = \begin{cases} 1, & y \geq threshold \\ 0, & otherwise \end{cases} \dots \quad (2.2)$$

Permasalahan yang sering muncul dalam mengkonversi citra *grayscale* menjadi citra biner adalah menentukan nilai *threshold*. Sebuah nilai *threshold* belum tentu bisa menghasilkan citra biner yang baik. Salah satu metode yang dapat membantu menentukan nilai *threshold* yang paling bagus adalah metode yang diusulkan oleh *Otsu*.

2.2.3 Format File Citra Digital

File citra berfungsi untuk menyimpan sebuah gambar yang dapat ditampilkan di layar ke dalam suatu media penyimpanan data. Untuk penyimpanan tersebut digunakan format gambar. Setiap format gambar memiliki karakteristik masingmasing. Beberapa format umum saat ini, yaitu bitmap (.bmp), tagged image format (.tif, tiff), portable network graphics (.png), graphics interchange format (.gif), jpeg (.jpg), mpeg (.mpg), dll.

1. Format File Citra JPG/JPEG

Joint Photographic Experts Group (JPEG) merupakan skema kompresi file bitmap. Awalnya, file yang menyimpan hasil foto digital memiliki ukuran yang besar sehingga tidak praktis. Dengan format baru ini, hasil foto yang semula berukuran besar berhasil dikompresi (dimampatkan) sehingga ukurannya kecil. JPEG adalah ekstensi file gambar yang pertama kali muncul yang mulai diperkenalkan pada tahun 1992 sebagai standar ISO. Saat itu format file gambar banyak yang menggunakan JPEG dan juga pada kamera-kamera zaman itu semua filenya memiliki format JPEG.

Format file JPEG dapat mengkompres objek dengan tingkat kualitas sesuai dengan pilihan yang disediakan dan dapat dimanfaatkan untuk menyimpan gambar yang akan digunakan untuk keperluan halaman web, multimedia, dan publikasi elektronik lainnya. Format file JPEG ini juga mampu menyimpan gambar dengan model warna RGB, CMYK, dan *Grayscale* serta mampu menyimpan alpha channel, namun karena orientasinya ke publikasi elektronik maka format ini berukuran relatif lebih kecil dibandingkan dengan format file lainnya.

2. Format File Citra PNG

PNG (*Portable Network Graphics*) adalah salah satu format penyimpanan citra yang menggunakan metode pemasangan yang tidak menghilangkan bagian dari citra tersebut. Format PNG ini diperkenalkan untuk menggantikan format penyimpanan citra GIF.

Untuk keperluan pengolahan citra, meskipun format PNG bisa dijadikan alternatif selama proses pengolahan citra - karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah (sehingga penyimpanan berulang ulang dari citra tidak akan menurunkan kualitas citra) namun format JPEG masih menjadi pilihan yang lebih baik.

PNG (Format berkas grafik yang didukung oleh beberapa web browser. PNG mendukung transparansi gambar seperti GIF, dan merupakan gambar bitmap yang terkompresi.

PNG (diucapkan 'ping') namun biasanya dieja apa adanya - untuk menghindari kerancuan dengan istilah "ping" pada jaringan komputer. PNG adalah

kependekan dari Portabel Network Graphics dan merupakan standar terbuka format image raster yang didukung oleh W3C dan IETF. Pada dasarnya, format PNG bukan merupakan format baru karena telah dikembangkan pada tahun 1995 untuk mengganti format GIF dan format TIFF. Format ini tidak digunakan lagi secara luas oleh browser dan perangkat lunak aplikasi pengolah gambar, sehingga dukungan terhadap format tidak begitu besar hingga tahun 2003, di mana format PNG semakin dikenal dan dipergunakan untuk aplikasi manipulasi gambar.

Secara garis besar, format PNG mempunyai fitur sebagai berikut:

1. Sebagai pengganti format GIF dan TIFF.
2. Format terbuka atau open, efisien, gratis, dan kompresi jenis lossless.
3. Tiga mode warna], yaitu: paletted (8 bit), greyscale (16 bit), truecolour (hingga 48 bit)
4. Dukungan terhadap profile colour, gamma, dan metadata.
5. Mempunyai fitur transparansi serta dukungan penuh terhadap alpha channel.
6. Dukungan luas bagi software manipulasi grafis dan web browser.

2.3 Pengolahan Citra

Operasi yang dilakukan untuk mentransformasikan suatu citra menjadi citra lain dapat dikategorikan berdasarkan tujuan transformasi maupun cakupan operasi yang dilakukan terhadap citra.

Berdasarkan tujuan transformasi operasi pengolahan citra dikategorikan sebagai berikut:

1. Grafika Komputer (Computer Graphics)
2. Peningkatan Kualitas Citra (Image Enhancement)

Operasi peningkatan kualitas citra bertujuan untuk meningkatkan fitur tertentu pada citra.

3. Pemulihan Citra (Image Restoration)

Operasi pemulihan citra bertujuan untuk mengembalikan kondisi citra pada kondisi yang diketahui sebelumnya akibat adanya pengganggu yang menyebabkan penurunan kualitas citra.

Berdasarkan cakupan operasi yang dilakukan terhadap citra, Operasi pengolahan citra dikategorikan sebagai berikut:

1. Operasi titik, yaitu operasi yang dilakukan terhadap setiap piksel pada citra yang keluarannya hanya ditentukan oleh nilai piksel itu sendiri.
2. Operasi area, yaitu operasi yang dilakukan terhadap setiap piksel pada citra yang keluarannya dipengaruhi oleh piksel tersebut dan piksel lainnya dalam suatu daerah tertentu. Salah satu contoh dari operasi berbasis area adalah operasi ketetanggaan yang nilai keluaran dari operasi tersebut ditentukan oleh nilai piksel-piksel yang memiliki hubungan ketetanggaan dengan piksel yang sedang diolah.
3. Operasi global, yaitu operasi yang dilakukan terhadap setiap piksel pada citra yang keluarannya ditentukan oleh keseluruhan piksel yang membentuk citra.

Tujuan dari pengolahan citra digital sebagai berikut:

1. Memperbaiki kualitas gambar dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek geometrik (rotasi, translasi, skala, transformasi geometrik).
2. Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.
3. Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

Proses citra, khususnya dengan menggunakan komputer akan menghasilkan hasil yang lebih baik dari sebelumnya. Berikut adalah alur dari pengolahan citra.

Citra Asli Proses → Pengolahan Citra → Citra Hasil

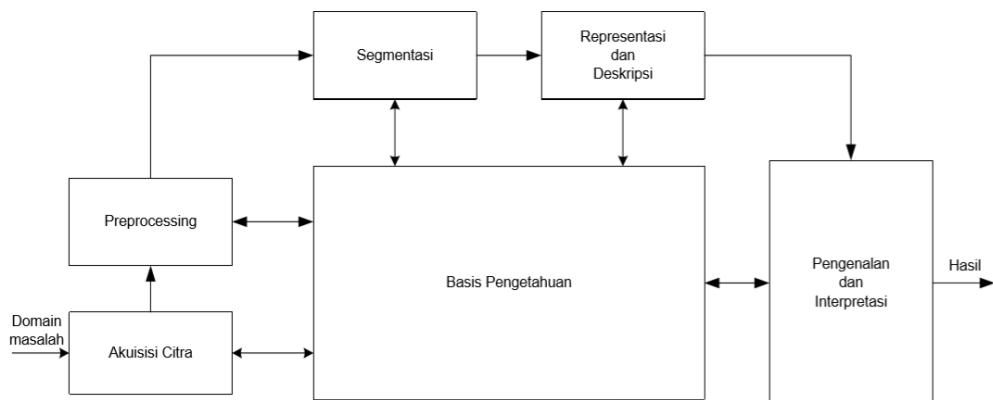
Pada umumnya, operasi-operasi pengolahan citra diterapkan pada citra apabila:

1. Perbaikan atau modifikasi citra untuk meningkatkan kualitas visual atau menonjolkan beberapa aspek informasi yang terkandung dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
3. Sebagian citra perlu di gabung dengan bagian citra yang lain.

Pengolahan citra juga mempunyai manfaat-manfaat pada bidang tertentu, antara lain:

1. Bidang perdagangan: Pembacaan barcode dan pengenalan huruf atau angka pada suatu formulir secara otomatis
2. Bidang militer: Mengenali sasaran peluru kendali melalui sensor visual dan mengidentifikasi jenis pesawat musuh
3. Bidang kedokteran: Mammografi dan rekonstruksi foto janin hasil USG
4. Bidang biologi: Pengenalan jenis kromosom melalui citra mikroskopik
5. Komunikasi data: Kompresi citra yang akan ditransmisikan
6. Hiburan: *Game* dan kompresi video
7. Hukum: Pengenalan sidik jari dan pengenalan foto narapidana

Dalam pengolahan citra terdapat langkah-langkah penting. Gambar 2.1 merupakan tahapan dalam pengolahan citra (Sutoyo, 2009).



Gambar 2.1 Tahap-Tahap Pengolahan Citra

1. Akuisisi citra

Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat, dan pada pencitraannya. Dimana pencitraan adalah kegiatan transformasi dari citra tampak (foto, gambar, lukisan, dll) menjadi citra digital.

2. *Preprocessing*

Preprocessing memerlukan tahapan untuk menjamin kelancaran pada proses berikutnya, antara lain:

1. Peningkatan kualitas citra (kontras, kecerahan, dll)
2. Menghilangkan *noise*
3. Perbaikan citra (*image restoration*)
4. Transformasi (*image transformation*)
5. Menentukan bagian citra yang akan diobservasi

3. Segmentasi

Segmentasi bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting, misalnya pada pemisahan objek dan latar belakang.

4. Representasi dan deskripsi

Representasi adalah suatu proses untuk merepresentasikan suatu wilayah sebagai suatu daftar titik-titik koordinat dalam kurva yang tertutup, dengan deskripsi luasan dan perimeternya. Proses selanjutnya dilakukan deskripsi citra dengan cara seleksi ciri dan ekstraksi ciri (*Feature Extraction and Selection*). Dimana seleksi ciri bertujuan untuk memilih informasi kuantitatif dari ciri yang ada, dan dapat membedakan kelas-kelas objek dengan baik, sedangkan ekstraksi ciri mempunyai tujuan untuk mengukur besaran kuantitatif ciri setiap piksel, misalnya rata-rata, standar deviasi, dan lain-lain.

5. Pengenalan dan interpretasi

Tahap pengenalan bertujuan untuk memberi label pada sebuah objek yang informasinya disediakan oleh *descriptor*, berbeda dengan tahap interpretasi yang bertujuan untuk memberi arti atau makna kepada kelompok objek-objek yang dikenali.

6. Basis pengetahuan

Basis pengetahuan bertujuan untuk memandu operasi dari masing-masing modul proses dan mengontrol interaksi antara modul-modul tersebut, dan dapat sebagai referensi pada proses pengenalan pola (*template matching*).

2.4 Perbaikan Kualitas Citra (*Image Enhancement*)

Perbaikan kualitas citra (*image enhancement*) merupakan salah satu proses awal dalam pengolahan citra (*image preprocessing*). Perbaikan kualitas diperlukan karena seringkali citra yang diuji mempunyai kualitas yang buruk, misalnya citra mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur, dan sebagainya. Melalui operasi pemrosesan awal inilah kualitas citra diperbaiki sehingga citra dapat digunakan untuk aplikasi lebih lanjut, misalnya untuk aplikasi pengenalan (*recognition*) objek di dalam citra.

Yang dimaksud dengan perbaikan kualitas citra adalah proses memperjelas dan mempertajam ciri atau fitur tertentu dari citra agar citra lebih mudah dipersepsi maupun dianalisis secara lebih teliti. Secara matematis, *image enhancement* dapat diartikan sebagai proses mengubah citra $f(x, y)$ menjadi $f'(x, y)$ sehingga ciri-ciri yang dilihat pada $f(x, y)$ lebih ditonjolkan. *Image enhancement* tidak meningkatkan kandungan informasi, melainkan jangkauan dinamis dari ciri agar bisa dideteksi lebih mudah dan tepat.

Operasi-operasi yang digolongkan sebagai perbaikan kualitas citra cukup beragam antara lain, pengubahan kecerahan gambar (*image brightness*), peregangan kontras (*Contrast Stretching*), perataan histogram (*histogram equalization*), pelembutan citra (*image smoothing*), penajaman (*sharpening*) tepi (*edge*), pewarnaan semu (*pseudocolouring*), pengubahan geometrik, dan sebagainya. (Munir, 2004)

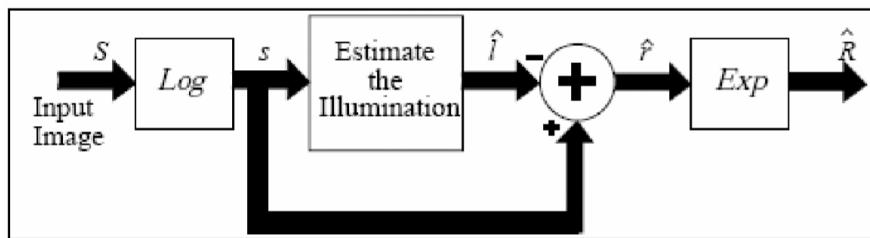
2.5 Metode *Retinex*

Metode *Retinex* ini dikemukakan oleh Edwin Land pada tahun 1971. Melalui eksperimen yang dilakukan olehnya, terlihat bahwa sistem penglihatan manusia mampu secara praktis mengenal dan mencocokkan warna-warna di bawah sebuah range illumination berbeda yang luas, hal ini dikenal dengan *Color Constancy Phenomenon*. *Color Constancy* atau ketetapan warna adalah salah satu keistimewaan dari sistem penglihatan manusia, yang mengusahakan agar warna yang diterima dari suatu benda terlihat sama meskipun berada pada kondisi

pencahayaan yang berbeda beda. Metode *Retinex* ini dikembangkan kembali oleh McCann dan kawan kawan pada tahun 1976.

Kata “*Retinex*” adalah penggabungan kata yang terbentuk dari kata “Retina” dan “Cortex”, yang menunjukkan bahwa baik mata dan otak terlibat dalam pemrosesan. Teori *Retinex* berhubungan dengan kompensasi untuk efek illumination (pencahayaan) pada citra.

$$S(x,y) = R(x,y) \cdot L(x,y) \dots \quad (2.3)$$



Gambar 2.2 Gambaran Umum Retinex

Tujuan utama dalam metode *Retinex* adalah untuk memisahkan image S ke dalam dua buah image yang berbeda, yaitu *reflectance image* R dan *illumination image* L, di mana pada setiap titik (x,y) dalam image domain. Terdapat beberapa perkembangan pada teori *Retinex* yaitu, *Single-Scale Retinex* (SSR), *Multi-Scale Retinex* (MSR), *Multi-Scale Retinex with Color Restoration* (MSRCR), dan *Multi-Scale Retinex with Chromaticity Preservation* (MSRCP).

2.5.1 Single-Scale Retinex (SSR)

Single-Scale Retinex (SSR) merupakan *Retinex* versi dinamis yang dikemukakan oleh Edwind Land untuk meniru sistem kerja neuron pada sistem persepsi warna manusia. Berikut ini adalah formulasi *Single-Scale Retinex*:

Dimana $R_i(x,y)$ merupakan output Retinex II (x,y) adalah distribusi citra pada piksel ke (x,y) . Simbol “ $*$ ” menyatakan operator konvolusi. Simbol i menyatakan channel warna, misalnya Red(R), Green(G), dan Blue(B). $F(x,y)$ merupakan fungsi Gaussian yang didefinisikan sebagai berikut:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \dots \quad (2.6)$$

Dimana, $G(x,y)$ merupakan Gaussian kernel pada piksel (x,y) . Simbol π atau pi merupakan konstanta yaitu $22/7$. Simbol σ merupakan nilai sigma. Simbol (x,y) adalah kordinat piksel. Sedangkan simbol e merupakan ketetapan yaitu 2.7182818246 .

Gaussian filter digunakan pada citra supaya citra lebih halus dalam memproses gambar. Gaussian filter juga bertujuan untuk menghilangkan noise pada citra dan meningkatkan kualitas detail citra.

Konvolusi merupakan perkalian total dari dua buah fungsi matriks f dan g . Untuk mengaplikasikan penapis atau filter pada citra, digunakan metode konvolusi. Konvolusi bisa dinyatakan dalam matriks, dimana setiap elemen matriks penapis merupakan koefisien konvolusi. Operasi konvolusi merupakan penggeseran kernel piksel per piksel dan hasil disimpan dalam matriks baru.

2.5.2 Multi-Scale Retinex (MSR)

Multi-Scale Retinex (MSR) perkembangan dari *Single Scale Retinex* karena keterbatasan yang dimiliki oleh *Single-Scale Retinex* (SSR), yaitu sebagai berikut:

1. SSR dapat melakukan kompresi jarak dinamis pada citra dengan skala rendah, sehingga memungkinkan citra dengan jarak dinamis sangat lebar dikompresi dengan melakukan penguatan bagian gelap dan melemahkan bagian yang terang. Sedangkan skala besar, SSR dapat menghasilkan citra lebih alami. Tetapi SSR tidak dapat melakukan kedua hal tersebut sekaligus.
 2. SSR cenderung menghasilkan area berwarna sama menjadi berwana abu-abu, terutama jika digunakan skala rendah.

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. Berikut adalah formulasi original *Multiscale Retinex*:

R_{msr_i} merupakan output dari *Multi-Scale Retinex* (MSR) yang merupakan jumlah dari output SSR yang masing-masing diberi bobot. N adalah jumlah skala yang digunakan. W_n adalah bobot yang diasosiasikan dengan skala ke- n . R_{ni} merupakan output dari SSR yang diasosiasikan dengan skala ke- n . Simbol i menyatakan channel warna, misalnya *Red(R)*, *Green(G)*, dan *Blue(B)*.

Berdasarkan formula MSR tersebut, konstruksi MSR ditentukan oleh parameter-parameter berikut ini.

1. Jumlah Skala (N)

Jumlah skala yang digunakan adalah 3 sebagai jumlah skala minimum yang menyediakan output yang baik berdasarkan persepsi visual dan waktu komputasi yang cepat.

2. Sigma yang digunakan (σ_n)

Sigma yang digunakan sebaiknya kombinasi dari tiga tingkatan yaitu rendah, sedang, dan tinggi tergantung dengan dimensi pada citra.

3. Bobot untuk setiap input SSR (W_n)

Bobot yang digunakan dalam proses MSR harus sama dengan satu. Jika $N=3$, maka $W_n = W_1+W_2+W_3$ dimana $W_n = 1$. Nilai W_1 , W_2 , dan W_3 bisa disesuaikan dengan kebutuhan.

2.5.3 Multiscale 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.

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 tetap terlihat sama walaupun pada kondisi pencahayaan yang berbeda-beda. Sebelum melakukan perhitungan MSRCR, terlebih dahulu harus melakukan perhitungan untuk *Color Restoration* (CR) sebagai berikut:

$$C_i(x, y) = \beta(\log(\alpha I_i(x, y)) - \log(I_R(x, y) + I_G(x, y) + I_B(x, y))) \dots (2.8)$$

Dimana: $C_i(x, y)$ adalah Color Restoration

αI_i adalah nilai ketentuan alpha

$I_R(x, y) + I_G(x, y) + I_B(x, y)$ adalah nilai intensitas kanal warna RGB.

2.5.4 Multiscale Retinex with Chromaticity Preservation (MSRCP)

Karna adanya masalah warna yang berubah dengan cara yang tidak terduga pada MSRCR yang cenderung membuat gambar menjadi keabuan untuk itu diambil keputuan untuk melakukan pemulihan beberapa warna yang hilang. Untuk itu terdapatlah solusi untuk menerapkan MSR pada saluran intensitas. Lalu, hasil warna keluaran dikomputasi sehingga kromatisitasnya sama dengan gambar aslinya.

Dapat dijelaskan saluran intensitas sebagai berikut:

Dimana S adalah jumlah saluran 3 kanal, RGB. Versi MSR ini menggunakan rumus:

$$R_{MSR_i} = \sum_{n=1}^N w_n R_{ni} = \sum_{n=1}^N w_n [\log I_i(x, y) - \log(F_n(x, y) * I_i(x, y))] \quad (2.10)$$

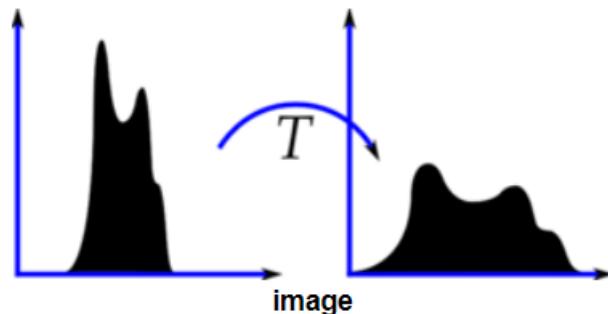
Untuk I, memperoleh hasil R_i . Kemudian transformasi linear diterapkan pada intensitas keluaran untuk meregangkan hasilnya ke [0, 255]. Akhirnya, mempertimbangkan intensitas keluaran akhir ini, saluran warna disimpulkan dengan mempertahankan inisial kromatisitasnya oleh:

Hasil yang diperoleh dengan bekerja pada saluran intensitas dan mempertahankan kromatisitas disebut dengan MSRPC (*Multiscale Retinex with Chromaticity Preservation*). MSRPC didasarkan pada MSR yang diterapkan pada saluran intensitas, dan menyesuaikan RGB sesuai dengan hasil penyaringan saluran intensitas untuk menjaga kromatisitas.

Karena MSRCP mempertahankan kromatisitas, MSRCP sangat baik untuk kompresi rentang dinamis dan peningkatan kontras lokal, sementara MSRCP tidak menghilangkan warna yang dilemparkan.

2.6 Metode *Contrast Stretching*

Contrast Stretching merupakan metode perbaikan kualitas citra yang bertujuan untuk meningkatkan atau menurunkan kontras suatu citra dengan cara memperlebar atau mempersempit range nilai intensitas piksel citra. Kontras adalah tingkat penyebaran piksel-piksel ke dalam intensitas warna.



Gambar 2.3 Gambaran Proses *Contrast Stretching*

Citra dikelompokkan ke dalam tiga kategori kontras, antara lain:

1. Citra Kontras-Rendah memiliki ciri dengan sebagian besar komposisi citranya adalah terang atau sebagian besarnya adalah gelap. Citra kontras-rendah dapat diperbaiki kualitasnya dengan operasi peregangan kontras. Nilai-nilai keabuan piksel akan merentang dari 0 sampai 255.
2. Citra Kontras-Bagus memperlihatkan jangkauan nilai keabuan yang lebar tanpa ada nilai keabuan yang mendominasi.
3. Citra Kontras-Tinggi memiliki jangkauan nilai keabuan yang lebar, tetapi terdapat area yang lebar dengan didominasi oleh warna gelap dan terang.

Proses peregangan kontras termasuk proses perbaikan citra yang hanya tergantung dari nilai intensitas satu piksel, tidak tergantung dari piksel lain yang ada disekitarnya.

$$\begin{cases} 0, & g(x,y) < rmin \\ g(x,y), & rmin \leq g(x,y) \leq rmax \\ 255, & g(x,y) \geq rmax \end{cases} \dots \quad (2.12)$$

Dimana: $g(x,y)$ = Nilai citra output

$f(x,y)$ = Nilai citra input

rmin = batas bawah

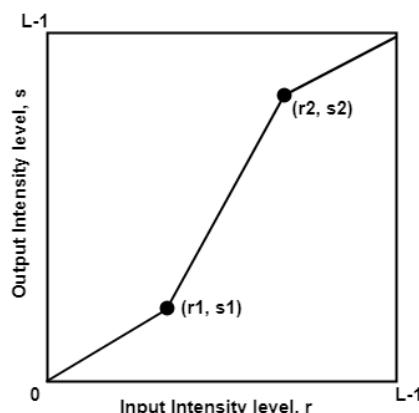
rmax = batas atas

Cara kerja proses peregangan kontras antara lain:

1. Cari batas bawah pengelompokan piksel dengan memindai histogram dari nilai keabuan terkecil ke nilai keabuan terbesar (0 sampai 255) untuk mendapatkan piksel pertama.
 2. Cari batas atas pengelompokan piksel dari nilai keabuan terbesar ke nilai keabuan terkecil (255 sampai 0).
 3. Piksel-piksel yang berada di bawah diset sama dengan 0, sedangkan piksel piksel yang berada di atas di set sama dengan 255.
 4. Piksel-piksel yang berada di antara nilai pertama dan kedua diskalakan untuk memenuhi rentang nilai-nilai keabuan yang lengkap dengan persamaan sebagai berikut:

$$g(x,y) = \frac{f(x,y) - r_{min}}{r_{max} - r_{min}} \times 255 \quad \dots \dots \dots \quad (2.13)$$

Dimana, r adalah nilai keabuan dalam citra semula, S adalah nilai keabuan yang baru, r_{min} adalah nilai keabuan terendah dari kelompok piksel, dan r_{max} adalah nilai keabuan tertinggi dari kelompok piksel.



Gambar 2.4 Proses Transformasi *Contrast Stretching*

Terdapat beberapa cara atau *function* dalam proses perenggangan kontras, yaitu sebagai berikut:

1. *None Stretch*

None stretch mempunyai maksud bahwa tidak ada metode perenggangan yang diaplikasikan, *function* ini hanya mengambil data piksel gambar awal dan data statistic gambar awal menjadi data tampilan gambar akhir.

2. Manimun – Maximum

Nilai *actual* minimum piksel menjadi 0 dan nilai *actual* maximum piksel menjadi 255. Pada *function* ini nilai min adalah nilai terkecil dari data histogram gambar awal dan nilai max adalah nilai terbesar dari data histogram gambar awal. Sebagai contoh sebuah gambar awal (8 bit) mempunyai nilai min = 34, dan nilai max = 250. Maka nilai histogram < dari 34 akan dipetakan menjadi 0 dan nilai histogram gambar ≥ 250 akan menjadi 255.

3. Percentile Stretching (Percent Clip)

Function ini memiliki cara yang sama dengan Max-Min, hanya saja nilai min dan nilai max dari gambar awal ditetukan dengan *clipped values*. Penelitian ini menggunakan *function percent clip*.

Berikut cara kerja dan rumus percentile:

- Urutkan seluruh data gambar RGB dari yang terkecil hingga terbesar: $x_1, x_2, x_3, \dots, x_n$.
 - Hitung peringkat r untuk persentil p yang ingin ditemukan:

Dimana: $r = \text{Data ke-}n$

p = Nilai persen low dan high

n = Banyak jumlah piksel

- Jika r bilangan bulat maka nilai data di lokasi r , x_r , adalah persentil p ; $p = x_r$
 - Jika r bukan bilangan bulat, p diinterpolasi menggunakan r_i , bagian bilangan bulat dari r , dan r_f , bagian pecahan dari r :

Dimana: x_{ri} = Data x_r ke- i

r_f = Bagian Pecahan dari r

4. Standard Deviation Stretch

Function ini menggunakan nilai mean dan nilai standar deviasi data histogram gambar awal untuk menentukan nilai min dan nilai max.

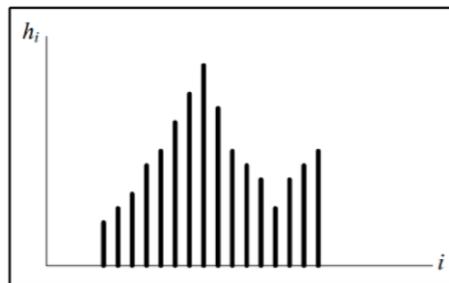
2.7 Histogram

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relative*) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar. Karena itu, histogram adalah alat bantu yang berharga dalam pengerjaan pengolahan citra baik secara kualitatif maupun kuantitatif.

Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna. Bila digambarkan pada koordinat maka sumbu x (absis) menunjukkan tingkat warna, sedangkan sumbu Y(ordinat) menunjukkan frekuensi kemunculan titik dan diagram yang menunjukkan jumlah kemunculan nilai grey-level pada suatu citra, dimana sumbu x diagram menggambarkan nilai grey-level dan sumbu y mewakili jumlah kemunculan grey-level tertentu.

Histogram mempunyai manfaat, antara lain:

1. Sebagai indikasi visual untuk menentukan skala keabuan yang tepat sehingga diperoleh kualitas citra yang diinginkan. Contoh: pengubahan kontras, kecemerlangan, dan lain-lain.
 2. Untuk pemilihan batas ambang (threshold) Contoh: proses segmentasi citra (memisahkan objek dari latar belakangnya) pada hakikatnya adalah menentukan batas-batas nilai keabuan dari objek dan batas-batas nilai keabuan latar belakangnya sehingga antara objek dan latar belakang dipisahkan.



Gambar 2.5 Histogram Citra

Pada umumnya tampilan histogram, yaitu:

1. Apabila gambar gelap maka histogram cenderung ke sebelah kiri
2. Apabila gambar terang maka histogram cenderung ke sebelah kanan
3. Apabila gambar low contrast maka histogram mengumpul di suatu tempat
4. Apabila gambar high contrast maka histogram merata di semua tempat.

2.8 Pengukur Kualitas Citra

Mengukur kualitas citra digital pada prinsipnya adalah membandingkan citra yang diukur dengan citra aslinya. Kualitas citra dapat diukur dengan parameter-parameter sebagai berikut (Roopaei, 2016):

2.8.1 PSNR dan MSE

Peak Signal to Noise Ratio (PSNR) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR biasanya diukur dalam satuan decibel (db). PSNR digunakan untuk mengetahui perbandingan kualitas citra cover sebelum dan sesudah disisipkan pesan. Untuk menentukan PSNR, terlebih dahulu harus ditentukan nilai MSE (*Mean Square Error*). MSE adalah nilai error kuadrat rata-rata antara citra asli dengan citra manipulasi.

Tujuan dari peak signal to noise ratio (PSNR) adalah dalam suatu pengembangan dan pelaksanaan rekonstruksi gambar diperlukan perbandingan antara gambar hasil rekonstruksi dengan gambar asli. Nilai PSNR yang lebih tinggi menyiratkan kemiripan yang lebih erat antara hasil rekonstruksi dan gambar asli.

Rumus MSE didefinisikan sebagai:

$$\text{MSE} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (S_{xy} - C_{xy})^2 \dots \quad (2.16)$$

Rumus PSNR didefinisikan sebagai:

$$\text{PSNR} = 10 \log_{10} \frac{C^2}{MSE} \dots \dots \dots \quad (2.17)$$

Dimana x dan y adalah koordinat dari gambar, M dan N adalah dimensi dari gambar.

MSE tidak memiliki satuan sedangkan satuan dari PSNR adalah decibel (dB). Semakin mirip kedua citra maka nilai MSE nya semakin mendekati nilai nol. Sedangkan pada PSNR, dua buah citra dikatakan memiliki tingkat kemiripan yang rendah jika nilai PSNR di bawah 30 dB.

2.9 Flowchart

Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempengaruhi penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut (Indrajani, 2011).

Flowchart di bedakan menjadi 5 jenis *Flowchart*, antara lain system *Flowchart*, document *Flowchart*, schematic *Flowchart*, program *Flowchart*, process *Flowchart*. Masing-masing jenis *Flowchart* akan dijelaskan berikut ini:

1. System *Flowchart*

System *Flowchart* dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

2. Document *Flowchart*

Bagan alir dokumen (document *Flowchart*) atau disebut juga bagan alir formulir (form *Flowchart*) atau paperwork *Flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

3. Schematic *Flowchart*

Bagan alir skematik (schematic *Flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaanya adalah, bagan alir skematik selain menggunakan symbol-simbol bagan alir sistem, juga menggunakan gambar-gambar computer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarnya.

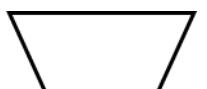
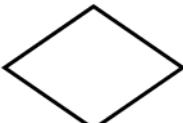
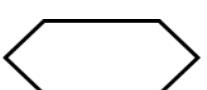
4. Program *Flowchart*

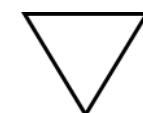
Bagan alir program (program *Flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari deriviasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (program logic flowchart) dan bagan alir program computer terinci (detailed computer program *Flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program computer secara logika. Bagan alir logika program ini dipersiapkan oleh analis sistem. Gambar berikut menunjukkan bagan alir logika program. Bagan alir program computer terinci (detailed computer program *Flowchart*) digunakan untuk menggambarkan instruksi-instruksi program computer secara terinci. Bagan alir ini dipersiapkan oleh pemrogram.

5. Process *Flowchart*

Bagan alir proses (process *Flowchart*) merupakan bagan alir yang banyak digunakan di teknik industry. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Tabel 2.1 Symbol – Symbol Flowchart

No.	Symbol	Nama	Keterangan
1		Arus / Flow	Untuk menyatakan jalannya arus suatu proses.
2		Communication link	Untuk menyatakan bahwa adanya transisi suatu data atau informasi dari suatu lokasi ke lokasi lainnya.
3		Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman / lembaran sama.
4		Offline Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembaran yang berbeda.
5		Proses	Sebuah fungsi pemrosesan yang dilaksanakan oleh computer biasanya menghasilkan perubahan terhadap data atau informasi.
6		Symbol manual	Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh computer (manual).
7		Decision / Logika	Untuk menunjukkan suatu kondisi tertentu, dgn dua kemungkinan, YA / TIDAK.
8		Predefined Process	Offline Storage Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.

9		Terminal	Untuk menyatakan permulaan atau akhir suatu program.
10		<i>Offline Storage</i>	Untuk menunjukkan bahwa data dalam symbol ini akan disimpan ke suatu media tertentu.
11		<i>Manual Input</i>	Untuk memasukkan data secara manual dengan menggunakan <i>online keyword</i> .
12		<i>Input / output</i>	Untuk menyatakan proses input dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
13		<i>Disk Storage</i>	Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
14		<i>Document</i>	Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .

2.10 Python

Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya.

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Versi terakhir yang dikeluarkan CWI adalah 1.2. Tahun 1995, Guido pindah ke CNRI sambil terus melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti Python pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan membentuk BeOpen PythonLabs. Python 2.0 dikeluarkan oleh BeOpen. Setelah mengeluarkan Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations.

Saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan Python Software Foundation. Python Software Foundation adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1 dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 2.6.1 dan versi 3.0.

Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan guido pada acara televisi Monty Python's Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python.

Beberapa fitur yang dimiliki Python adalah:

1. Memiliki kepustakaan yang luas; dalam distribusi Python telah disediakan modul - modul siap pakai untuk berbagai keperluan.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
4. Berorientasi objek.
5. Memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java).
6. Modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.
7. Memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman Java, python memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan komputer secara langsung.

Beberapa contoh built-in library yang sering digunakan:

1. Math — Library untuk rumus matematika
2. String — Library untuk pengolah string
3. Datetime — Library untuk tanggal dan waktu
4. CSV — Library untuk mengolah data csv
5. Json — Library untuk encoder dan decoder json

Selain built-in library ada juga library lainnya yang banyak digunakan untuk mengolah data di python antara lain:

1. NumPy
2. Pandas
3. Matplotlib
4. Seaborn
5. SciPy
6. SciKit-Learn
7. Tensorflow
8. Keras
9. Bokeh
10. Plotly
11. NLTK
12. Scrapy
13. OpenCV

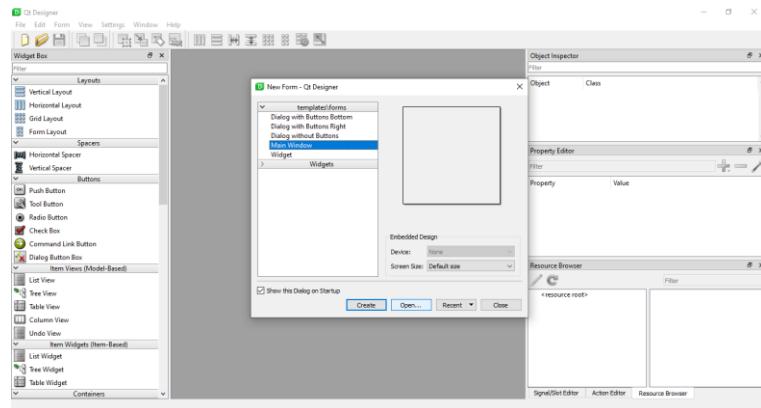
Pada proses penyusunan program penelitian, Python yang digunakan adalah versi 3.8.7.

2.11 PYQT

PyQt adalah toolkit widget GUI. Ini adalah antarmuka Python untuk Qt, salah satu pustaka GUI lintas platform yang paling kuat dan populer. PyQt dikembangkan oleh RiverBank Computing Ltd. Qt juga mencakup Qt Designer, yaitu sebuah aplikasi yang untuk mendesain antar muka grafis. Qt Designer memungkinkan pembuatan widget, dialog, dan main window lengkap dengan hanya melakukan drag-and-drop pada form (Python Software Foundation, 2015). Dengan menggunakan Qt Designer, tiap perubahan pada halaman yang telah dibuat dapat dilihat secara langsung.

Berkas yang dibuat dalam Qt Designer dapat diubah menjadi berkas Python. Kontrol terhadap antar muka grafis yang dibuat dapat ditambahkan dalam bentuk kode Python. Hal ini menjadi suatu kelebihan dari PyQt, dimana

kemudahan desain yang disediakan oleh Qt dapat dimanipulasi dengan bahasa Python yang cenderung mudah dipelajari.



Gambar 2.6 Tampilan Qt Designer

PyQt API adalah sekumpulan modul yang berisi sejumlah besar kelas dan fungsi. Sementara QtCore modul berisi fungsionalitas non-GUI untuk bekerja dengan file dan direktori dll., QtGui modul berisi semua kontrol grafis. Selain itu, ada modul untuk bekerja dengan XML(QtXml), SVG (QtSvg), dan SQL (QtSql), dan lain – lain.

Daftar modul yang sering digunakan diberikan di bawah ini:

- QtCore - Kelas inti non-GUI yang digunakan oleh modul lain
- QtGui - Komponen antarmuka pengguna grafis
- QtMultimedia - Kelas untuk pemrograman multimedia tingkat rendah
- QtNetwork - Kelas untuk pemrograman jaringan
- QtOpenGL - Kelas dukungan OpenGL
- QtScript - Kelas untuk mengevaluasi Qt Scripts
- QSql - Kelas untuk integrasi database menggunakan SQL
- QtSvg - Kelas untuk menampilkan konten file SVG
- QtWebKit - Kelas untuk merender dan mengedit HTML
- QtXml - Kelas untuk menangani XML
- QtWidgets - Kelas untuk membuat UI bergaya desktop klasik
- QtDesigner - Kelas untuk memperluas Qt Designer

Pada proses penyusunan program penelitian, PYQT yang digunakan adalah PYQT5 versi 5.15.2.

2.12 OpenCV

OpenCV (Open Source Computer Vision Library) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini bebas dan berada dalam naungan sumber terbuka dari lisensi BSD. Pustaka ini merupakan pustaka lintas platform. Program ini didedikasikan sebagian besar untuk pengolahan citra secara real-time. Jika pustaka ini menemukan pustaka Integrated Performance Primitives dari intel dalam sistem komputer, maka program ini akan menggunakan rutin ini untuk mempercepat proses kerja program ini secara otomatis.

OpenCV adalah sebuah library (perpustakaan) yang digunakan untuk mengolah gambar dan video hingga kita mampu meng-ekstrak informasi didalamnya. Intel meluncurkan versi pertama dari OpenCV pada 1999, dan awalnya memerlukan library dari Intel Image Processing Library. Kemudian dependecy tersebut dihilangkan sehingga terciptalah OpenCV seperti yang sekarang sebagai standalone library.

OpenCV dapat berjalan di berbagai bahasa pemograman, seperti C, C++, Java, Python, dan juga support diberbagai platform seperti Windows, Linux, Mac OS, iOS dan Android.

OpenCV pertama kali diluncurkan secara resmi pada tahun 1999 oleh Inter Research sebagai lanjutan dari bagian proyek bertajuk aplikasi intensif berbasis CPU, real-time ray tracing dan tembok penampil 3D. Para kontributor utama dalam proyek ini termasuk mereka yang berkecimpung dalam bidang optimasi di Intel Russia, dan juga Tim Pusatka Performansi Intel. Pada awalnya, tujuan utama dari proyek OpenCV ini dideskripsikan sebagai berikut:

1. Penelitian penginderaan citra lanjutan tidak hanya melalui kode program terbuka, tetapi juga kode yang telah teroptimasi untuk infrastruktur penginderaan citra.

2. Menyebarluarkan ilmu penginderaan citra dengan menyediakan infrastruktur bersama dimana para pengembang dapat menggunakannya secara bersama-sama, sehingga kode akan tampak lebih mudah dibaca dan ditransfer.
3. Membuat aplikasi komersial berbasiskan penginderaan citra, dimana kode yang telah teroptimasi tersedia secara bebas dengan lisensi yang tersedia secara bebas yang tidak mensyaratkan program itu harus terbuka atau gratis.

OpenCV dapat dijalankan pada Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, BlackBerry 10, Linux and OS X.

OpenCV mempunyai banyak fitur yang dapat dimanfaatkan, dintaranya adalah:

1. Image and Video I/O, dengan antar muka ini kita dapat membaca data gambar dari file, atau dari umpan video langsung. Dan juga dapat menciptakan file gambar maupun video.
2. Computer Vision secara umum dan pengolahan citra digital (untuk low dan mid level API), dengan antar muka ini kita dapat melakukan eksperimen uji coba dengan berbagai standar algorima computer vision. Termasuk juga deteksi garis, tepi, pucuk, proyeksi elips, image pyramid untuk pemrosesan gambar multi skala, pencocokan template, dan berbagai transform (Fourier, cosine diskrit, distance transform) dan lain lain.
3. Modul Computer Vision High Level, didalam OpenCV juga termasuk kemampuan “high level”, seperti kemampuan tambahan untuk deteksi wajah, pengenalan wajah, termasuk optical flow.
4. Metode untuk AI dan Machine Learning, aplikasi computer vision sering kali memerlukan machine learning atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket OpenCV machine learning.
5. Sampling Gambar dan Transformasi, didalam OpenCV sudah terdapat antar muka untuk substraksi subregion dari gambar, random sampling, rotating, dan lain lain.

Kelebihan OpenCV adalah memiliki library dokumen yang cukup banyak, dapat bekerja secara cepat pada komputer yang menggunakan processor intel,

komputasi yang lebih ringan bila dibandingkan menggunakan Matlab dalam hal pengolahan citra digital (gambar) dan dapat bekerja secara real time. Pada proses penyusunan program penelitian, OpenCV yang digunakan adalah versi 4.5.1.48.

2.13 Numpy

NumPy (Numerical Python) adalah library Python yang fokus pada scientific computing. NumPy memiliki kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada Python. Keunggulan NumPy array dibandingkan dengan list pada Python adalah konsumsi memory yang lebih kecil serta runtime yang lebih cepat. Pada proses penyusunan program penelitian, Numpy yang digunakan adalah versi 1.19.5.

2.14 Matplotlib

Matplotlib adalah library untuk membuat plot dua dimensi (The Matplotlib Development Team, 2015). Hasil dapat ditampilkan dalam bentuk sebaran titik-titik, diagram garis, diagram batang, histogram, spektrum, dan lain-lain. Penyusun program juga memiliki kontrol atas model titik, garis, bidang, huruf, sumbu, dan berbagai unsur lainnya dalam tampilan hasil.

Matplotlib juga mencakup mplot3d, sehingga hasil yang dapat ditampilkan tidak hanya dalam bentuk dua dimensi, namun juga tiga dimensi (The Matplotlib Development Team, 2015). Hasil mplot3D bersifat interaktif, memungkinkan pengguna untuk memperbesar dan memperkecil, serta melakukan rotasi pada gambar. Pada proses penyusunan program penelitian, Matplotlib yang digunakan adalah versi 3.3.3.

2.15 Kajian Terdahulu

Kajian terdahulu ataupun penelitian sebelumnya menjadi bahan acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mempelajari penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian yang diangkat penulis. Namun penulis mengangkat beberapa

penelitian sebagai referensi bahan kajian pada penelitian penulis. Adapun beberapa penelitian sebelumnya yang berkaitan dengan perbaikan kualitas citra untuk peningkatan kecerahan citra digital, sebagai berikut:

Tabel 2.2 Kajian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1.	Hotmantri Simbolon, 2015	Penerapan metode <i>Contrast Stretching</i> untuk peningkatan kualitas citra bidang biomedis.	Dengan Mengkombinasikan metode <i>Contrast Stretching</i> dapat memperjelas hasil pengambilan gambar dengan sinar-X yang dulunya gambarnya susah disimpulkan hasilnya menjadi dapat dilihat dengan hasil gambar setelah terapkan metode <i>Contrast Stretching</i> . Sehingga nantinya bidang biomedisnya bisa menyimpulkan penyakit yang ada didalam paru-paru pasien.
		Perbedaan: Penelitian yang dilakukan Hotmantri Simbolon menggunakan bahan penelitian berupa citra grayscale, sedangkan penelitian ini menggunakan citra warna (RGB) sebagai bahan penelitian.	
2.	Nurliadi, Poltak Sihombing dan	Analisis <i>Contrast Stretching</i> menggunakan algoritma	<i>Contrast Stretching</i> tidak hanya dapat digunakan pada citra grayscale namun citra berwarna dapat juga

	Marwan Ramli, 2016	Euclidean untuk meningkatkan kontras pada citra berwarna.	diterapkan dalam metode tersebut.
	Perbedaan: Penelitian yang dilakukan Nurliadi, Poltak Sihombing dan Marwan Ramli mengkombinasi algoritma Euclidean sebagai penentu jarak kontras citra, sedangkan penelitian ini hanya menggunakan metode <i>Contrast Stretching</i> saja.		
3.	Fadhillah Azmi, Sherly, Sirano Lahagu dan David, 2019	Implementasi Metode <i>Retinex</i> dan Histogram Equalization Pada Kecerahan Citra Digital.	Algoritma histogram equalization lebih tepat untuk memperbaiki citra berdasarkan hasil gambar yang didapat. Sedangkan berdasarkan histogram, histogram equalization hasilnya lebih merata dan tinggi pikselnya hampir sama, dibandingkan <i>Retinex</i> yang memiliki perbedaan tinggi histogram, sehingga algoritma histogram equalization lebih cocok untuk memperbaiki citra. Nilai rata – rata SNR dari algoritma <i>Retinex</i> adalah 12.7769 dan untuk algoritma histogram equalization, nilai rata–rata SNR-nya adalah 10.2613,

			dari hasil tersebut maka algoritma <i>Retinex</i> lebih cocok untuk perbaikan citra dari segi SNR citra.
Perbedaan: Penelitian yang dilakukan Fadhillah Azmi, Sherly, Sirano Lahagu dan David menggunakan SNR sebagai salah satu indikator pembanding hasil penelitian sedangkan penelitian ini menggunakan nilai PSNR dan MSE sebagai indikator pembanding citra hasil.			
4.	Soeb Aripin, Guidio Leonarde Ginting, Natalia Silalahi, 2017	Penerapan Metode <i>Retinex</i> Untuk Meningkatkan Kecerahan Citra Pada Hasil Screenshot.	Metode <i>Retinex</i> dapat diterapkan dalam pencerahan citra dengan cara proses dengan menggunakan perkalian karnel dan menghitung nilai tengah pixel yang telah dikalikan dengan karnel dan menjadi sebuah citra baru dengan kecerahan citra pixel yang lebih jelas dari aslinya.
Perbedaan: Penelitian yang dilakukan Soeb Aripin, Guidio Leonarde Ginting, Natalia Silalahi menggunakan operasi titik pixel penyeleksian terhadap proses pembesaran citra hasil, sedangkan penelitian ini tidak melakukan proses penyeleksian.			
5.	Bainun Harahap, 2018	Implementasi Metode <i>Retinex</i> Untuk Meningkatkan Kualitas Citra Underwater.	Metode <i>Retinex</i> dapat diterapkan dalam pencerahan citra dengan cara proses dengan menggunakan perkalian karnel dan konvolusi pixel

		<p>yang telah dikalikan dengan karnel dan menghasilkan nilai SSR1, SSR2 dan SSR3 dan dihasilkan citra output dengan kecerahan citra pixel yang lebih jelas dari aslinya.</p>
		<p>Perbedaan: Penelitian yang dilakukan Bainun Harahap hanya berfokus pada tahap SSR, sedangkan penelitian ini membahas SSR, MSR, MSRCR dan MSRCP sebagai varian metode <i>Retinex</i>.</p>

BAB III

METODOLOGI PENELITIAN

3.1 Waktu Penelitian

Waktu penyusunan penulisan penelitian dimulai pada akhir bulan april 2020, dilanjutkan dengan pembelajaran dan pengumpulan data teori dari metode *Contrast Stretching* dan metode *Retinex* sampai pada bulan juni 2020, proses penulisan proposal penelitian selesai pada bulan agustus 2020.

3.2 Bahan Penelitian

Bahan penelitian sebagai data analisis yang digunakan dalam penelitian perbandingan metode *Contrast Stretching* dan metode *Retinex* ini adalah berupa gambar warna (*true color*) atau image warna (RGB) yang berekstensi .JPG dan .PNG. Data yang berupa gambar tersebut adalah gambar dengan keadaan terlalu terang, terlalu gelap, gambar dengan kontras tinggi dan gambar dengan kontras rendah. Data gambar tersebut didapat atau diambil dari kamera *handphone* dan gambar digital yang diunduh dari internet dengan jumlah pixel yang tidak dibatasi.

3.3 Alat Penelitian

Alat penelitian adalah alat bantu yang digunakan oleh peneliti untuk mengumpulkan data penelitian. Adapun alat – alat yang digunakan pada penelitian ini adalah sebagai berikut:

3.3.1 Perangkat Keras

Perangkat keras yang digunakan sebagai alat penelitian ini adalah Laptop Acer Aspire A314-33 dengan spesifikasi prosesor Intel® Celeron® N4000 CPU @ 1.10GHz, RAM 4GB 64bit, 1000GB HDD dan Smartphone Samsung Galaxy A20s sebagai alat pengambil data gambar.

3.3.2 Perangkat Lunak

Perangkat lunak yang digunakan sebagai alat penelitian ini adalah Bahasa Pemrograman Python versi 3.8.7 dengan *library* OpenCV versi 4.5.1.48, Numpy versi 1.19.5, Matplotlib versi 3.3.3, PyQt5 versi 5.15.2 dan sistem operasi *Windows 10 Home Single Language* dari *Microsoft Corporation* 2019.

3.4 Teknik Pengumpulan Data

Salah satu hal yang sangat penting dalam melakukan penelitian adalah teknik pengumpulan data. Kesalahan yang terjadi dalam proses pengumpulan data dapat menyebabkan proses analisis data menjadi rumit dan rentan terhadap tingkat akurasi hasil penelitian. Teknik pengumpulan data yang digunakan pada penelitian ini adalah teknik studi dokumen atau dokumentasi. Studi dokumen adalah metode pengumpulan data yang tidak ditujukan langsung kepada subjek penelitian. Studi dokumen adalah jenis pengumpulan data yang meneliti berbagai macam dokumen yang berguna untuk bahan analisis.

Data gambar yang dikumpulkan sebagai bahan analisis adalah berupa data primer, yaitu data yang dikumpulkan dan diolah sendiri oleh peneliti. Data diambil dan dikumpulkan pada *dictionary* alat penelitian.

3.5 Analisis Kebutuhan

Analisis kebutuhan merupakan satu diantara banyak aktivitas kritis pada proses rekayasa kebutuhan perangkat lunak untuk memahami ranah permasalahan dari sistem yang berjalan dan ranah solusi dari sistem yang akan dibuat (Yen et.al., 1998). Analisis kebutuhan terbagi menjadi dua bagian penting, yaitu kebutuhan functional dan kebutuhan non-functional.

3.5.1 Kebutuhan Functional

Kebutuhan functional merupakan jenis kebutuhan yang berisi proses – proses apa saja yang nantinya akan dilakukan oleh sistem, kebutuhan functional juga berisi informasi apa saja yang dihasilkan oleh sistem. Adapun kebutuhan

functional yang digunakan dalam sistem perbaikan kualitas citra dengan metode *Contrast Stretching* dan metode *Retinex* adalah sebagai berikut:

1. Sistem akan melakukan peningkatan kecerahan citra digital berformat .JPG dan .PNG. Citra yang digunakan adalah citra true color dengan jumlah pixel yang tidak dibatasi.
2. Sistem melakukan peningkatan kecerahan citra menggunakan metode *Contrast Stretching* dan metode *Retinex*.
3. Sistem akan melakukan proses penyimpanan data hasil dari proses metode *Contrast Stretching* dan metode *Retinex* dengan format .JPG ataupun .PNG.
4. Sistem akan menampilkan hasil histogram dari citra awal dan citra hasil.
5. Sistem akan menampilkan hasil perhitungan dari PSNR dan MSE dari citra hasil proses sebagai nilai indikator perbandingan kualitas citra.

3.5.2 Kebutuhan non-Functional

Kebutuhan non-functional merupakan kebutuhan yang berisi properti perilaku yang dimiliki suatu sistem. Adapun kebutuhan non-functional pada sistem yang harus terpenuhi adalah sebagai berikut:

1. Performa (*Performance*)

Sistem yang dibangun harus dapat memperbaiki kualitas citra dan menunjukkan proses peningkatan kecerahan pada citra hasil.

2. Informasi (*Information*)

Sistem yang akan dibangun harus mudah dimengerti oleh pengguna agar informasi yang dibutuhkan oleh user tersampaikan dengan baik.

3. Hemat biaya (*Economy*)

Sistem yang dibangun tidak membutuhkan perangkat tambahan untuk implementasinya sehingga tidak menambah biaya.

4. Dokumentasi

Sistem yang dibangun harus dapat menyimpan citra dengan format .JPG dan .PNG, serta sistem juga harus mampu menyimpan citra hasil filtering.

5. Manajemen kualitas

Sistem yang dibangun harus memiliki kualitas yang baik, dimana sistem dapat memperbaiki kualitas citra, menghasilkan citra yang lebih baik dari citra aslinya dan dapat menghitung nilai PSNR dan MSE dari citra hasil.

6. Kontrol

Sistem yang akan dibangun harus dapat menampilkan kotak dialog error ketika user salah dalam melakukan penginputan.

3.6 Kerangka Penelitian

Kerangka dalam penelitian adalah kumpulan konsep yang tersusun secara sistematis agar tujuan penelitian yang dilakukan menjadi baik. Pengertian kerangka penelitian adalah konsep suatu penelitian yang menghubungkan antara visualisasi satu variabel dengan variable lainnya, sehingga penelitian menjadi tersusun secara sistematis dan dapat diterima oleh semua pihak.

Kerangka penelitian ini adalah membandingkan hasil dari proses dua metode perbaikan kualitas gambar yaitu metode *Retinex* dan metode *Contrast Stretching* untuk mencapai tujuan mengetahui metode mana yang lebih baik dalam menangani gambar dengan kondisi yang sudah ditentukan. Kondisi tersebut adalah gambar awal dengan keadaan terlalu terang, terlalu gelap, berkontras tinggi dan berkontras rendah.

Kerangka penelitian ataupun langkah – langkah yang dilakukan dalam penelitian ini dijabarkan sebagai berikut:

Langkah 1: Mengambil dan mengumpulkan gambar melalui kamera (handphone maupun laptop) atau pun diunduh dari internet sebagai bahan analisis sistem. Data gambar tersebut disimpan pada *dictionary* laptop tempat program peningkatan kecerahan citra metode *Contrast Stretching* dan metode *Retinex* dijalankan.

Langkah 2: Membuat program sebagai tempat implementasi sekaligus alat pengujian perbandingan bahan data analisis dari metode *Contrast Stretching* dan metode *Retinex*. Program yang digunakan adalah program desktop dibuat menggunakan Bahasa Pemrograman Python 3 dan Library OpenCV,

Matplotlib, Numpy, PyQt5, QtDesigner. Program diharapkan mampu mengolah bahan data analisis sesuai dengan algoritma metode *Contrast Stretching* dan metode *Retinex* agar mendapatkan hasil yang akurat. Program juga diharapkan mampu menghitung nilai PSNR dan MSE dari citra awal dan citra hasil proses sebagai indikator nilai pembanding dari kedua metode tersebut dan menampilkan histogram dari citra awal dan citra hasil proses pengujian. Dan juga program diharapkan mampu menyimpan citra hasil proses pengujian dengan format citra awal yaitu .JPG dan .PNG.

Langkah 3: Program akan melakukan pengujian menggunakan gambar dan foto yang telah diambil sebagai bahan penelitian. Proses pengujian dianalisis oleh peneliti dan data – data hasil yang didapat dari hasil pengujian akan disimpan dicatat secara rinci sebagai proses dokumentasi.

Langkah 4: Membandingkan gambar hasil dari metode *Contrast Stretching* dan metode *Retinex*. Pada langkah ini, citra hasil proses pengujian kedua metode akan dibandingkan dengan melihat citra awal. Apakah citra hasil menghasilkan citra yang lebih baik dari citra awal ataupun tidak.

Langkah 5: Menjabarkan hasil perhitungan nilai PSNR dan MSE dari gambar (citra) awal dan gambar (citra) hasil. Hasil perhitungan nilai PSNR dan MSE citra awal dan citra hasil proses pengujian antara metode *contrast stretching* dan metode *Retinex* dicatat dan dibandingkan.

Langkah 6: Menampilkan dan membandingkan histogram dari citra awal dan citra hasil proses pengujian dari metode *Contrast Stretching* dan metode *Retinex*.

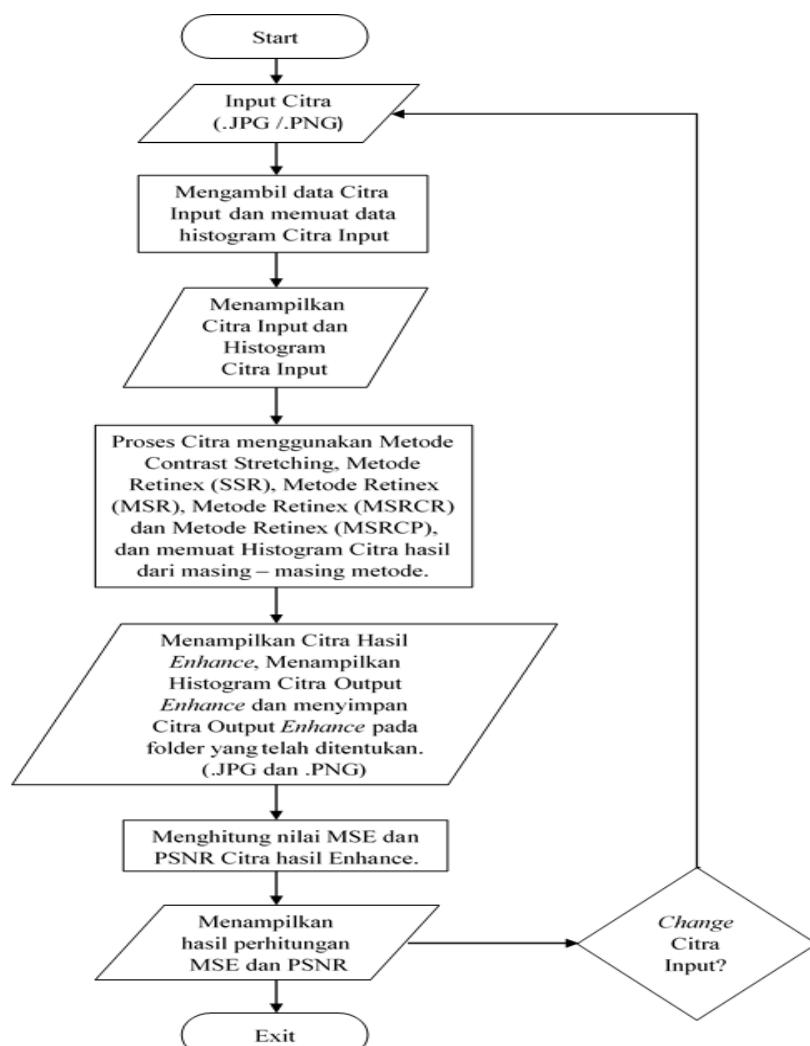
Langkah 7: Membuat kelebihan dan kekurangan dari metode *Contrast Stretching* dan metode *Retinex* berdasarkan hasil tes penelitian data analisis.

Langkah 8: Membandingkan metode *Contrast Stretching* dan metode *Retinex* dengan memberikan simpulan metode mana yang lebih baik pada hasil penelitian.

3.7 Perancangan Sistem Penelitian

Menurut Satzinger, Jackson, dan Burd (2012:5), Perancangan Sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Perancangan sistem merupakan gambaran tentang apa yang dikerjakan pada sistem penelitian. Pada tahap perancangan sistem, alur kerja sistem dapat dilihat dari *Flowchart* sistem. *Flowchart* adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

Flowchart dari rancangan sistem peningkatan kecerahan citra digital dapat dilihat sebagai berikut:



Gambar 3.1 *Flowchart* sistem

Pada *Flowchart* diatas dapat kita lihat proses perancangan alur kendali dari program yang digunakan dalam tahap implementasi penelitian ini. Dapat diuraikan sebagai berikut:

1. Mulai, Input citra awal atau gambar asli yang berformat .JPG ataupun .PNG.
2. Menampilkan citra awal dan histogram citra awal.
3. Proses *Enhancement*, gambar input (citra awal) akan diproses dengan metode peningkatan kecerahan citra digital. Adapun metode terkait ialah Metode *Contrast Stretching*, Metode *Retinex* (SSR), Metode *Retinex* (MSR), Metode *Retinex* (MSRCR), Metode *Retinex* (MSRCP) dan memuat histogram dari masing – masing metode.
4. Gambar output akan tampil setelah proses selesai juga akan tersimpan secara otomatis pada folder yang telah ditentukan dengan format .JPG dan .PNG sebagai dokumentasi hasil citra output (citra enhance) dan menampilkan dan menyimpan histogram dari masing – masing citra hasil proses enhance yang bertujuan untuk mempermudah dalam perbandingan hasil.
5. Proses perhitungan MSE dan PSNR dari citra hasil enhance.
6. Menampilkan hasil perhitungan MSE dan PSNR dari citra hasil enhance.
7. Jika ingin mengganti ataupun memperbaharui citra awal (input) dapat langsung kembali ke tahap proses input citra, citra akan otomatis terganti dengan citra input yang baru. Jika tidak, maka proses peningkatan kecerahan citra digital selesai.
8. Exit atau selesai.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Metode

Implementasi adalah kegiatan yang dilakukan untuk menguji data dan menerapkan sistem yang diperoleh dari hasil analisis. Implementasi merupakan salah satu unsur tahapan dari keseluruhan pembangunan sistem komputerisasi, dan unsur yang harus dipertimbangkan dalam pembangunan sistem.

Berikut adalah sampel gambar yang digunakan dalam proses perhitungan implementasi metode:

R: 5 G: 24 B: 39	R: 146 G: 165 B: 180	R: 118 G: 135 B: 151	R: 62 G: 80 B: 94	R: 64 G: 77 B: 93
R: 3 G: 19 B: 32	R: 84 G: 100 B: 113	R: 224 G: 238 B: 251	R: 54 G: 66 B: 78	R: 25 G: 35 B: 45
R: 14 G: 21 B: 27	R: 16 G: 23 B: 29	R: 107 G: 112 B: 118	R: 18 G: 23 B: 27	R: 0 G: 2 B: 7
R: 14 G: 18 B: 21	R: 21 G: 25 B: 26	R: 25 G: 26 B: 28	R: 5 G: 6 B: 8	R: 3 G: 1 B: 2
R: 1 G: 5 B: 6	R: 4 G: 8 B: 7	R: 2 G: 4 B: 3	R: 8 G: 10 B: 9	R: 25 G: 23 B: 24

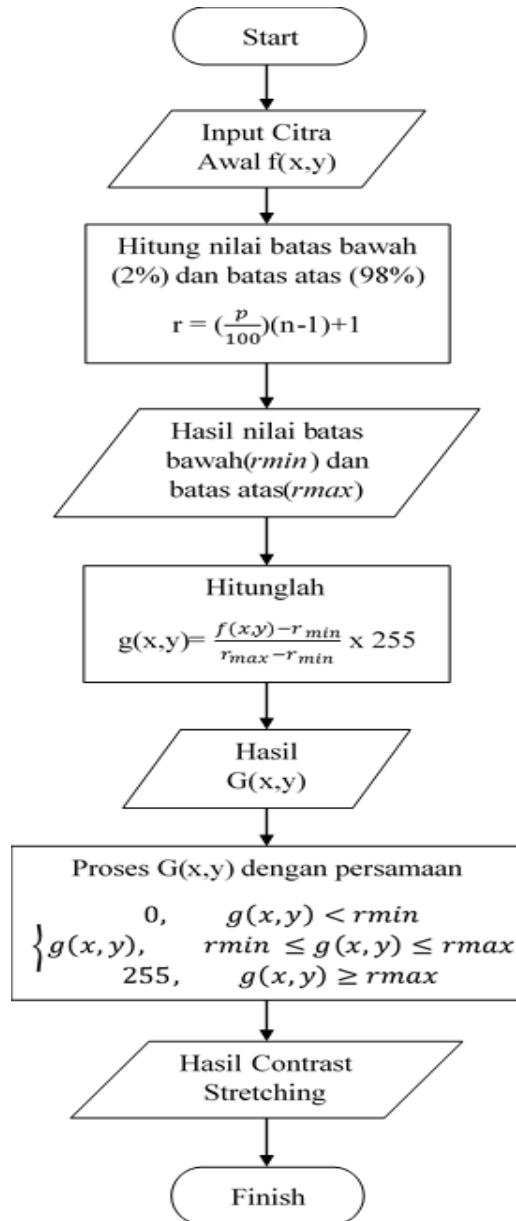
Gambar 4.1 Gambar Sampel RGB 5x5 Piksel

5	146	118	62	64	24	165	135	80	77	39	180	151	94	93
3	84	224	54	25	19	100	238	66	35	32	113	251	78	45
14	16	107	18	0	21	23	112	23	2	27	29	118	27	7
14	21	25	5	3	18	25	26	6	1	21	26	28	8	2
1	4	2	8	25	5	8	4	10	23	6	7	3	9	24
Red					Green					Blue				

Gambar 4.2 Gambar Tabel Data Citra Sampel RGB

4.1.1 Contrast Stretching

Berikut ini adalah penjelasan implementasi dari metode *Contrast Stretching* yang menggunakan *function percentile stretching* atau *percent clip contrast stretch*:



Gambar 4.3 Flowchart Contrast Stretching

Cara kerja atau tahap penyelesaian dari *function percent clip* sama dengan *function Max-Min* tetapi perbedaannya adalah terletak pada menetuan nilai min

(batas bawah) dan nilai max (batas atas). Jika pada function Max-Min, nilai min (batas bawah) dan nilai max (batas atas) ditentukan dari nilai piksel terkecil dan terbesar dari channel atau kanal warna RGB, sedangkan pada *percent clip*, nilai min dan nilai max ditentukan dengan menggunakan nilai persen atau *percentile* suatu channel atau kanal warna RGB. Penelitian ini menggunakan sample Gambar 4.6, nilai min (batas bawah) 2% dan nilai max (batas atas) 98%.

Penyelesaian:

Tahap 1 – Cari nilai min 2% dan nilai max 98% .

Cara mencari nilai batas bawah (2%) dan batas atas (98%)

- Urutkan seluruh data gambar RGB dari yang terkecil hingga terbesar: $x_1, x_2, x_3, \dots, x_n$ (4.1)

Dari data sample Gambar 4.2, didapat $n = 75$, urutan nilai sebagai berikut:

0, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 5, 5, 5, 6, 6, 7, 7, 8, 8, 8, 8, 9, 10, 14, 14, 16, 18, 18, 19, 21, 21, 21, 23, 23, 23, 24, 24, 25, 25, 25, 25, 26, 26, 27, 27, 28, 29, 32, 35, 39, 45, 54, 62, 64, 66, 77, 78, 80, 84, 93, 94, 100, 107, 112, 113, 118, 118, 135, 146, 151, 165, 180, 224, 238, 251.

- Hitung peringkat r untuk persentil p yang ingin ditemukan:

$$r = \left(\frac{p}{100}\right)(n-1)+1 \quad \dots \quad (4.2)$$

Dimana: r = Data ke- n

p = Nilai persen low dan high

n = Banyak jumlah piksel

- Jika r bilangan bulat maka nilai data di lokasi r , x_r , adalah persentil p : $p = x_r$ (4.3)
- Jika r bukan bilangan bulat, p diinterpolasi menggunakan r_i , bagian bilangan bulat dari r , dan r_f , bagian pecahan dari r :

$$p = x_{r_i} + r_f * (x_{r_i + 1} - x_{r_i}) \quad \dots \quad (4.4)$$

Dimana: x_{r_i} = Data x_r ke- i

r_f = Bagian Pecahan dari r

Hitung:

Tabel 4.1 Tabel Perhitungan Nilai Percentile

high = 98%	low = 2%
$r = \left(\frac{98}{100}\right)(75-1)+1=73,52$	$r = \left(\frac{2}{100}\right)(75-1)+1=2,48$
$r = 73, rf = 0,52$	$r = 2, rf = 0,48$
$r73,52 = v_{73}+rf(v_{74} - v_{73})$	$r2,48 = v_2+rf(v_3 - v_2)$
$r73,52 = 224 + 0,52(238-224) = 231,28$	$r2,48 = 1 + 0,48(1-1) = 1$

low = 1; high = 231,28; max = 255; min = 0;

Tahap 2 - dari Hasil diatas maka diperoleh persamaan sebagai berikut:

$$\begin{cases} 0, & g(x,y) < 1 \\ g(x,y) = \frac{f(x,y) - 1}{231,28 - 1} * 255, & 1 \leq g(x,y) \leq 231,28 \\ 255, & g(x,y) \geq 231,28 \end{cases}$$

- Jika nilai $g(x,y)$ lebih kecil dari 1 maka nilai $g(x,y)$ menjadi 0.
- Jika 1 lebih kecil sama dengan nilai $g(x,y)$, dan nilai $g(x,y)$ lebih kecil sama dengan 231,28, maka nilai tetap menjadi nilai $g(x,y)$.
- Jika nilai $g(x,y)$ lebih besar sama dengan 231,28, maka nilai $g(x,y)$ menjadi 255.

Bentuk perhitungan sebagai berikut:

$$g(x,y) = \frac{f(5)-1}{231,28-1} * 255 = 4$$

$$g(x,y) = \frac{f(146)-1}{231,28-1} * 255 = 160$$

Proses perhitungan dilanjutkan sampai ke semua nilai. Hasil seluruh perhitungan menjadi:

4	160	129	67	69	25	181	148	87	84	42	198	166	102	101
2	91	246	58	26	19	109	255	71	37	34	124	255	85	48
14	16	117	18	0	22	24	122	24	1	28	31	129	28	6
14	22	26	4	2	18	26	27	5	0	22	27	29	7	1
0	3	1	7	26	4	7	3	9	24	5	6	2	8	25
Red					Green					Blue				

Gambar 4.4 Data Citra Hasil *Contrast Stretching*



Gambar Sampel

R: 5 G: 24 B: 39	R:146 G:165 B:180	R:118 G:135 B:151	R: 62 G: 80 B: 94	R: 64 G: 77 B: 93
R: 3 G: 19 B: 32	R: 84 G:100 B:113	R:224 G:238 B:251	R: 54 G: 66 B: 78	R: 25 G: 35 B: 45
R: 14 G: 21 B: 27	R: 16 G: 23 B: 29	R:107 G:112 B:118	R: 18 G: 23 B: 27	R: 0 G: 2 B: 7
R: 14 G: 18 B: 21	R: 21 G: 25 B: 26	R: 25 G: 26 B: 28	R: 5 G: 6 B: 8	R: 3 G: 1 B: 2
R: 1 G: 5 B: 6	R: 4 G: 8 B: 7	R: 2 G: 4 B: 3	R: 8 G: 10 B: 9	R: 25 G: 23 B: 24

Hasil *Contrast Stretching*

R: 4 G: 25 B: 42	R:160 G:181 B:198	R:129 G:148 B:166	R: 67 G: 87 B:102	R: 69 G: 84 B:101
R: 2 G: 19 B: 34	R: 91 G:109 B:124	R:246 G:255 B:255	R: 58 G: 71 B: 85	R: 26 G: 37 B: 48
R: 14 G: 22 B: 28	R: 16 G: 24 B: 31	R:117 G:122 B:129	R: 18 G: 24 B: 28	R: 0 G: 1 B: 6
R: 14 G: 18 B: 22	R: 22 G: 26 B: 27	R: 26 G: 27 B: 29	R: 4 G: 5 B: 7	R: 2 G: 0 B: 1
R: 0 G: 4 B: 5	R: 3 G: 7 B: 6	R: 1 G: 3 B: 2	R: 7 G: 9 B: 8	R: 26 G: 24 B: 25

Gambar 4.5 Hasil Enhance Metode *Contrast Stretching*

Berikut *coding* program untuk metode *Contrast Stretching*:

```
def constrech(self):
    self.image = self.tmp
    img = self.image
    min_percent = 2
    max_percent = 98
    low, high = np.percentile(img, (min_percent, max_percent))
    res_img = (img.astype(float) - low) / (high-low)
    return np.maximum(np.minimum(res_img*255, 255), 0).astype(np.uint8)
```

4.1.2 Retinex

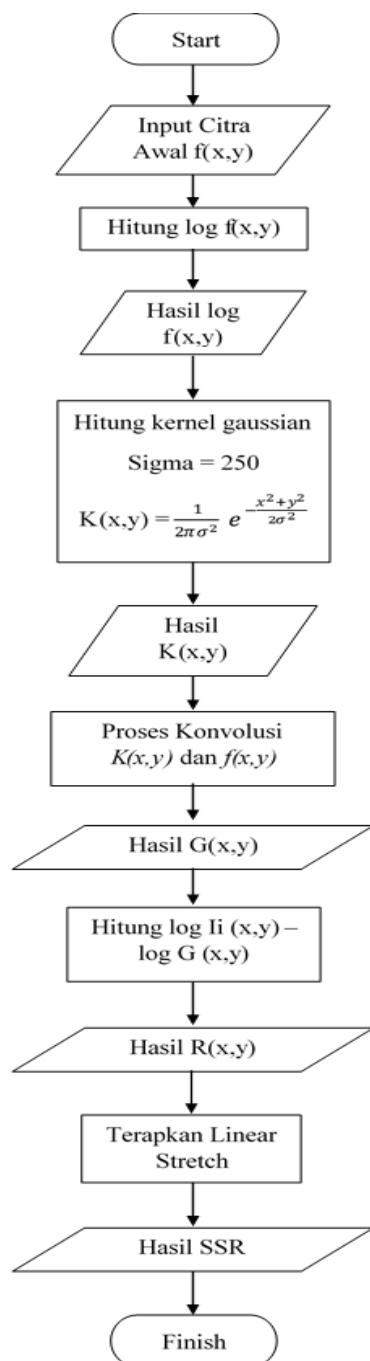
Berikut ini adalah penjelasan implementasi dari metode *Retinex*. Penelitian ini memakai 4 varian dari metode *Retinex*, yaitu:

1. *Single Scale Retinex (SSR)*
2. *Multiscale Retinex (MSR)*
3. *Multiscale Retinex with Color Restoration (MSRCR)*
4. *Multiscale Retinex with Chromaticity Presevation (MSRCP)*

4.1.2.1 SSR

Single Scale Retinex mempunyai formulasi sebagai berikut:

$$R_i(x,y) = \log I_i(x,y) - \log [F(x,y) * I_i(x,y)] \dots \dots \dots \quad (4.5)$$



Gambar 4.6 Flowchart Single Scale Retinex

Tahapan algoritma implementasi SSR:

Tahap 1: Hitunglah atau tentukan $\log I_i(x,y)$.

Tahap 2: Hitunglah atau tentukan nilai $G(x,y)$, dimana $G(x,y)$ adalah hasil konvolusi dari $I_i(x,y)$ citra awal dan $F(x,y)$ kernel fungsi Gaussian.

Tahap 3: Hitunglah atau tentukan $\log G(x,y)$ lalu lakukan proses perhitungan $\log I_i(x,y) - \log G(x,y)$.

Tahap 4: Terapkan linear stretch pada hasil tahap 3. Selesai.

$F(x,y)$ merupakan fungsi Gaussian yang didefinisikan sebagai berikut:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \dots \quad (4.6)$$

Berikut contoh proses perhitungan kernel fungsi Gaussian:

- Contoh perkalian dengan matriks 3×3 , $\sigma = 0,6$.

Tabel 4.2 Tabel Contoh (x,y)

-	x	<table border="1"> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	0	1	-1	0	1	-1	0	1	y	<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	1	1	1	0	0	0	-1	-1	-1
-1	0	1																				
-1	0	1																				
-1	0	1																				
1	1	1																				
0	0	0																				
-1	-1	-1																				

$\frac{1}{2\pi 0,6^2} e^{-\frac{-1^2+1^2}{20,6^2}}$	$\frac{1}{2\pi 0,6^2} e^{-\frac{0^2+1^2}{20,6^2}}$	$\frac{1}{2\pi 0,6^2} e^{-\frac{1^2+1^2}{20,6^2}}$
$\frac{1}{2\pi 0,6^2} e^{-\frac{-1^2+0^2}{20,6^2}}$	$\frac{1}{2\pi 0,6^2} e^{-\frac{0^2+0^2}{20,6^2}}$	$\frac{1}{2\pi 0,6^2} e^{-\frac{1^2+0^2}{20,6^2}}$
$\frac{1}{2\pi 0,6^2} e^{-\frac{-1^2+-1^2}{20,6^2}}$	$\frac{1}{2\pi 0,6^2} e^{-\frac{0^2+-1^2}{20,6^2}}$	$\frac{1}{2\pi 0,6^2} e^{-\frac{1^2+-1^2}{20,6^2}}$

Gambar 4.7 Contoh Perhitungan Kernel Fungsi Gaussian

Hasil perhitungan menjadi:

0,0275	0,1102	0,0275
0,1102	0,4423	0,1102
0,0275	0,1102	0,0275

Gambar 4.8 Hasil Contoh Kernel Gaussian

SUM atau Jumlah nilai = 0,9931

Jumlah total dari seluruh nilai yang dihasilkan harus sama dengan 1, jika tidak memenuhi maka dinormalisasi atau percobaan akan diulang kembali dengan rumus:

Contoh: $New_{(1,1)} = 0,0275 * \frac{1}{0,9931} = 0,0276$

Perhitungan dilanjutkan sampai seluruh nilai kernel dan sampai menghasilkan Jumlah nilai = 1.

Berikut hasil perhitungan nilai kernel dengan Jumlah nilai = 1.

0,0276	0,1109	0,0276
0,1109	0,4456	0,1109
0,0276	0,1109	0,0276

Gambar 4.9 Hasil Contoh Kernel Gaussian Baru

Setelah kernel nilai fungsi Gaussian didapatkan, maka proses selanjutnya adalah proses konvolusi dengan citra awal. Contoh citra awal (3×3 piksel) 3 bit warna.

7	5	1
2	4	6
1	0	1

Gambar 4.10 Contoh Citra Awal (3x3 piksel) 3 bit warna

0	0	0	0	0
0	7	5	1	0
0	2	4	6	0
0	1	0	1	0
0	0	0	0	0

0,0275	0,1102	0,0275
0,1102	0,4423	0,1102
0,0275	0,1102	0,0275

Proses konvolusi didapatkan dengan proses perkalian citra awal dan kernel fungsi Gaussian.

Proses perkalian:

$$(0*0,0276)+(0*0,1109)+(0*0,0276)+(0*0,1109)+(7*0,4456)+(5*0,1109)+(0*0,0276)+(2*0,1109)+(4*0,0276) = 4,0059$$

$$(0*0,0276)+(0*0,1109)+(0*0,0276)+(7*0,1109)+(5*0,4456)+(1*0,1109)+(2*0,0276)+(4*0,1109)+(6*0,0276) = 3,7796$$

$$(0*0,0276)+(0*0,1109)+(0*0,0276)+(5*0,1109)+(1*0,4456)+(0*0,1109)+(4*0,0276)+(6*0,1109)+(0*0,0276) = 1,7759$$

$$(0*0,0276)+(7*0,1109)+(5*0,0276)+(0*0,1109)+(2*0,4456)+(4*0,1109)+(0*0,0276)+(1*0,1109)+(0*0,0276) = 2,36$$

$$(7*0,0276)+(5*0,1109)+(1*0,0276)+(2*0,1109)+(4*0,4456)+(6*0,1109)+(1*0,0276)+(0*0,1109)+(1*0,0276) = 3,5001$$

$$(5*0,0276)+(1*0,1109)+(0*0,0276)+(4*0,1109)+(6*0,4456)+(0*0,1109)+(0*0,0276)+(1*0,1109)+(0*0,0276) = 3,477$$

$$(0*0,0276)+(2*0,1109)+(4*0,0276)+(0*0,1109)+(1*0,4456)+(0*0,1109)+(0*0,0276)+(0*0,1109)+(0*0,0276) = 0,7778$$

$$(2*0,0276)+(4*0,1109)+(6*0,0276)+(1*0,1109)+(0*0,4456)+(1*0,1109)+(0*0,0276)+(0*0,1109)+(0*0,0276) = 0,8862$$

$$(4*0,0276)+(6*0,1109)+(0*0,0276)+(0*0,1109)+(1*0,4456)+(0*0,1109)+(0*0,0276)+(0*0,1109)+(0*0,0276) = 1,2214$$

Berikut hasil dari proses konvolusi:

4,0059	3,7796	1,7759
2,36	3,5001	3,477
0,7778	0,8862	1,2214

Gambar 4.11 Hasil Konvolusi

Sebelumnya telah dijabarkan alur proses perhitungan kernel fungsi Gaussian dan proses konvolusi yang digunakan pada metode *Single Scale Retinex*. Selanjutnya akan dijelaskan tahapan proses SSR dengan menggunakan Gambar

4.6 sebagai sampel citra dan nilai Sigma = 250. Pada citra RGB, proses dilakukan per-kanal warna. Perhitungan dimulai dari kanal Red (R), data diambil dari Gambar 4.6 Citra sample.

Tahap 1: $\log(I(x,y) + 1)$

Untuk mencegah hasil 0, maka setiap intensitas ditambah 1.

Tabel 4.3 Hasil $\log I(x,y)$

0.77815125	2.16731733	2.07554696	1.79934055	1.81291336
0.60205999	1.92941893	2.35218252	1.74036269	1.41497335
1.17609126	1.23044892	2.03342376	1.2787536	0
1.17609126	1.34242268	1.41497335	0.77815125	0.60205999
0.30103	0.69897	0.47712125	0.95424251	1.41497335

Tahap 2: $G(x,y)$

Tabel 4.4 Hasil Konvolusi

48.57812462	48.57814596	48.57817507	48.57813499	48.57812997
48.57812705	48.57814839	48.5781775	48.57813742	48.5781324
48.57809767	48.57811901	48.57814811	48.57810804	48.57810302
48.57808676	48.5781081	48.57813721	48.57809713	48.57809211
48.57808277	48.57810411	48.57813321	48.57809314	48.57808811

Tahap 3: $\log(G(x,y) + 1)$

Untuk mencegah hasil 0, maka setiap intensitas ditambah 1.

Tabel 4.5 $\log(G(x,y))$

1.69529009	1.69529028	1.69529054	1.69529019	1.69529014
1.69529012	1.6952903	1.69529056	1.69529021	1.69529016

1.69528986	1.69529005	1.6952903	1.69528995	1.69528991
1.69528976	1.69528995	1.69529021	1.69528985	1.69528981
1.69528973	1.69528992	1.69529017	1.69528982	1.69528978

Tabel 4.6 $\log(I(x,y)) - \log(G(x,y))$

-0.91713884	0.47202705	0.38025642	0.10405036	0.11762322
-1.09323012	0.23412862	0.65689196	0.04507248	-0.28031681
-0.5191986	-0.46484112	0.33813345	-0.41653635	-1.69528991
-0.5191985	-0.35286727	-0.28031686	-0.9171386	-1.09322982
-1.39425973	-0.99631991	-1.21816892	-0.74104731	-0.28031643

Tahap 4: Linear stretch, nilai min (batas bawah) dan max (batas atas) diambil dari hasil Tahap 3.

Nilai Min = -1.6952899055327368; Nilai Max = 0.6568919601403076;

Tabel 4.7 Hasil SSR, kanal (Red), Sigma = 250

84	234	225	195	196
65	209	255	188	153
127	133	220	138	0
127	145	153	84	65
32	75	51	103	153

Tahapan proses perhitungan yang sama dilanjutkan pada kanal warna Green (G) dan Blue (B). Berikut hasil SSR pada ketiga kanal warna (RGB):

84	234	225	195	196	134	235	224	197	195	149	235	225	198	198
65	209	255	188	153	122	209	255	187	154	138	209	255	188	157
127	133	220	138	0	127	132	215	132	21	128	132	211	128	56
127	145	153	84	65	120	136	138	66	0	114	126	130	63	0
32	75	51	103	153	58	80	48	90	132	48	56	16	69	122
Red					Green					Blue				

Gambar 4.12 Data Citra Hasil SSR

Gambar Sampel					Hasil Retinex (SSR)				
R: 5 G: 24 B: 39	R: 146 G: 165 B: 180	R: 118 G: 135 B: 151	R: 62 G: 80 B: 94	R: 64 G: 77 B: 93	R: 84 G: 134 B: 149	R: 234 G: 235 B: 235	R: 225 G: 224 B: 225	R: 195 G: 195 B: 198	R: 196 G: 195 B: 198
R: 3 G: 19 B: 32	R: 84 G: 100 B: 113	R: 224 G: 238 B: 251	R: 54 G: 66 B: 78	R: 25 G: 35 B: 45	R: 65 G: 122 B: 138	R: 209 G: 209 B: 209	R: 255 G: 255 B: 255	R: 188 G: 187 B: 188	R: 153 G: 154 B: 157
R: 14 G: 21 B: 27	R: 16 G: 23 B: 29	R: 107 G: 112 B: 118	R: 18 G: 23 B: 27	R: 0 G: 2 B: 7	R: 127 G: 127 B: 128	R: 133 G: 132 B: 132	R: 220 G: 215 B: 211	R: 138 G: 132 B: 128	R: 0 G: 21 B: 56
R: 14 G: 18 B: 21	R: 21 G: 25 B: 26	R: 25 G: 26 B: 28	R: 5 G: 6 B: 8	R: 3 G: 1 B: 2	R: 127 G: 120 B: 114	R: 145 G: 136 B: 126	R: 153 G: 138 B: 130	R: 84 G: 66 B: 63	R: 65 G: 0 B: 0
R: 1 G: 5 B: 6	R: 4 G: 8 B: 7	R: 2 G: 4 B: 3	R: 8 G: 10 B: 9	R: 25 G: 23 B: 24	R: 32 G: 58 B: 48	R: 75 G: 80 B: 56	R: 51 G: 48 B: 16	R: 103 G: 90 B: 69	R: 153 G: 132 B: 122

Gambar 4.13 Hasil Enhance Metode Retinex (SSR)

Berikut coding program untuk metode Single Scale Retinex:

```
def Retinex_SSR(self):
    self.image = self.tmp
    img = self.image
    sigma = 250
    if len(img.shape)==2:
        img=img[... ,None]
    ret=np.zeros(img.shape,dtype='uint8')
    for i in range(img.shape[-1]):
        channel=img[... ,i].astype('double')
        S_log=np.log10(channel+1)
        #kernel gaussian
        ksize=int(np.floor(sigma*6)/2)*2+1
        k_1D=np.arange(ksize)-ksize//2
        k_1D=np.exp(-k_1D**2/(2*sigma**2))
        k_1D=k_1D/np.sum(k_1D)
        qr = k_1D
        #gaussian blur
        row_filter= qr
        t=cv2.filter2D(channel,-1,row_filter[... ,None])
        tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
        gaussian= tr
        L_log=np.log10(gaussian+1)
        r=S_log-L_log
```

```

R=r
mmin=np.min(R)
mmax=np.max(R)
stretch=(R-mmin)/(mmax-mmin)*255
ret[...,i]=stretch
return ret.squeeze()

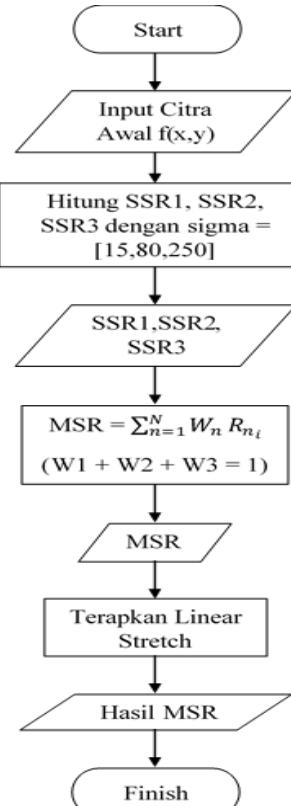
```

Note: Jika citra awal (input) mempunyai dimensi piksel yang kecil seperti 5×5 piksel, disarankan untuk memakai nilai sigma yang kecil pula.

4.1.2.2 MSR

MSR (*Multiscale Retinex*) adalah metode lanjutan dari SSR (*Single Scale Retinex*) yang dimana, jika pada metode SSR memakai 1 buah nilai sigma, maka pada metode MSR ini memakai lebih dari 1 sigma. MSR adalah penggabungan dari hasil metode *Single Scale Retinex* yang diberi bobot. *Multiscale* yang berarti memakai banyak skala atau sigma. Bobot yang dipakai harus sama dengan 1.

Berikut adalah formulasi original *Multiscale Retinex*:



Gambar 4.14 *Flowchart Multiscale Retinex*

Tahapan algoritma implementasi MSR:

Tahap 1: Hitunglah atau tentukan nilai hasil dari metode SSR dari masing – masing nilai sigma.

Tahap 2: Lakukan penjumlahan pada masing – masing hasil SSR dan dikalikan dengan bobot.

Tahap 3: Terapkan linear stretch. Selesai.

Penelitian ini memakai nilai sigma adalah [15, 80, 250]. Gambar 4.6 sebagai citra awal.

Proses implementasi:

Tahap 1: Sebelumnya telah didapat citra hasil metode SSR dengan sigma 250, dilanjutkan dengan mencari citra hasil metode SSR dengan sigma 15 dan 80.

Tahap 2: Setelah ketiga hasil SSR didapatkan, maka proses selanjutnya adalah perkalian bobot.

$$R_{MSR} = (SSR1 * W1) + (SSR2 * W2) + (SSR3 * W3)$$

Dimana: SSR1 = citra hasil SSR dengan sigma = 15

SSR2 = citra hasil SSR dengan sigma = 80

SSR3 = citra hasil SSR dengan sigma = 250

W1 = 0,333; W2 = 0,333; W3 = 0,334. (Jumlah bobot harus sama dengan 1).

Tahap 3: Terapkan linear stretch dengan nilai min (batas bawah) = 0; Dan nilai max (batas atas) = 255. Berikut hasil dari metode MSR:

84	234	225	195	196	134	235	224	197	195	149	235	225	198	198
65	209	255	188	153	122	209	255	187	154	138	209	255	188	157
127	133	220	138	0	127	132	215	132	21	128	132	211	128	56
127	145	153	84	65	120	136	138	66	0	114	126	130	63	0
32	75	51	103	153	58	80	48	90	132	48	56	16	69	122
Red					Green					Blue				

Gambar 4.15 Data Citra Hasil MSR sigma [15 ,80, 250]



Gambar Sampel

R: 5 G: 24 B: 39	R:146 G:165 B:180	R:118 G:135 B:151	R: 62 G: 80 B: 94	R: 64 G: 77 B: 93
R: 3 G: 19 B: 32	R: 84 G:100 B:113	R:224 G:238 B:251	R: 54 G: 66 B: 78	R: 25 G: 35 B: 45
R: 14 G: 21 B: 27	R: 16 G: 23 B: 29	R:107 G:112 B:118	R: 18 G: 23 B: 27	R: 0 G: 2 B: 7
R: 14 G: 18 B: 21	R: 21 G: 25 B: 26	R: 25 G: 26 B: 28	R: 5 G: 6 B: 8	R: 3 G: 1 B: 2
R: 1 G: 5 B: 6	R: 4 G: 8 B: 7	R: 2 G: 4 B: 3	R: 8 G: 10 B: 9	R: 25 G: 23 B: 24

Hasil Retinex (MSR)

R: 84 G:134 B:149	R:234 G:235 B:235	R:225 G:224 B:225	R:195 G:197 B:198	R:196 G:195 B:198
R: 65 G:122 B:138	R:209 G:209 B:209	R:255 G:255 B:255	R:188 G:187 B:188	R:153 G:154 B:157
R:127 G:127 B:128	R:133 G:132 B:132	R:220 G:215 B:211	R:138 G:132 B:128	R: 0 G: 21 B: 56
R:127 G:120 B:114	R:145 G:136 B:126	R:153 G:138 B:130	R: 84 G: 66 B: 63	R: 65 G: 0 B: 0
R: 32 G: 58 B: 48	R: 75 G: 80 B: 56	R: 51 G: 48 B: 16	R:103 G: 90 B: 69	R:153 G:132 B:122

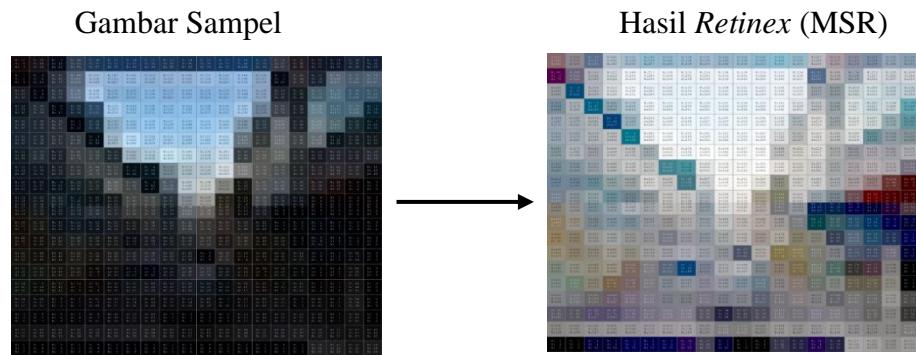
Gambar 4.16 Hasil Enhance Metode *Retinex* (MSR)

Berikut *coding* program untuk metode *Multiscale Retinex*:

```
def MultiScaleRetinex(self):
    self.image = self.tmp
    img = self.image
    flag=True
    weights=None
    sigmas=[15,80,250]
    if weights==None:
        weights=np.ones(len(sigmas))/len(sigmas)
    elif not abs(sum(weights)-1)<0.00001:
        raise ValueError('sum of weights must be 1!')
    r=np.zeros(img.shape,dtype='double')
    img=img.astype('double')
    for i,sigma in enumerate(sigmas):
        #kernel gaussian
        ksize=int(np.floor(sigma*6)/2)*2+1
        k_1D=np.arange(ksize)-ksize//2
        k_1D=np.exp(-k_1D**2/(2*sigma**2))
        k_1D=k_1D/np.sum(k_1D)
        qr = k_1D
        #gaussian blur
        row_filter= qr
        t=cv2.filter2D(img,-1,row_filter[...],None)
        tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
        r+=(np.log10(img+1)-np.log10(tr+1))*weights[i]
    if flag:
        mmin=np.min(r,axis=(0,1),keepdims=True)
        mmax=np.max(r,axis=(0,1),keepdims=True)
        r=(r-mmin)/(mmax-mmin)*255
        r=r.astype('uint8')
    return r
```

Note: Dikarenakan citra awal terlalu kecil, dan nilai sigma yang besar membuat hasil dari metode MSR sama dengan hasil metode SSR. Oleh karna itu penelitian dilanjutkan untuk mencari tahu solusi dari permasalahan ini. Setelah melanjutkan

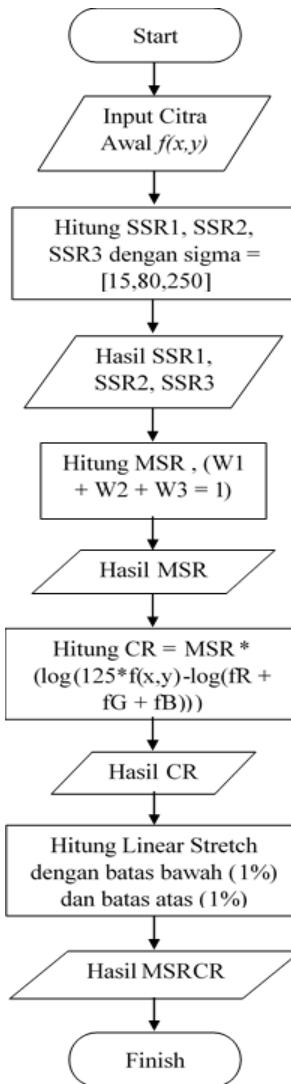
penelitian dengan mengganti citra awal, didapat hasil bahwa metode SSR dan metode MSR dengan nilai sigma = [15, 80, 250] bekerja dengan baik pada citra awal dengan dimensi 20 x 20 piksel.



Gambar 4.17 Hasil Enhance Metode *Retinex* (MSR) 20 x 20 Piksel

4.1.2.3 MSRCR

Multiscale Retinex with Color Restoration adalah metode lanjutan dari metode *Multiscale Retinex* (MSR). Metode ini pada dasarnya adalah hasil perkalian metode MSR dan *color restoration*.



Gambar 4.18 Flowchart Multiscale Retinex with Color Restoration

Tahapan algoritma MSRCR:

Algorithm 1: MSRCR algorithm

Data: I input color image; $\sigma_1, \sigma_2, \sigma_3$ the scales; s_1, s_2 the percentage of clipping pixels on each side

Result: Output color image

begin

foreach $c \in \{R, G, B\}$ **do**

foreach σ_i **do**

 | $Diff_{i,c} = \log(I_c) - \log(I_c * G_{\sigma_i})$

 | **end**

 | $MSR_c = \sum_i \frac{1}{3}Diff_{i,c}$

 | $MSRCR_c = MSR_c \times (\log(125I_c) - \log(I_R + I_G + I_B))$

 | $Out_c = SimplestColorBalance(MSRCR_c, s_1, s_2)$

 | **end**

 | **end**

 | **For each color channel**

 | | **For each scale**

 | | **Single Scale Retinex**

 | | **MultiScale Retinex**

 | | **Color Restoration**

Gambar 4.19 Algoritma MSRCR

Keterangan: alpha = 125.

Pada metode sebelumnya telah didapatkan hasil dari citra awal dengan metode MSR, maka tahap selanjutnya dilanjutkan dengan proses perkalian dengan fungsi color restoration dan diakhiri dengan proses percent clip.

84	234	225	195	196	134	235	224	197	195	149	235	225	198	198
65	209	255	188	153	122	209	255	187	154	138	209	255	188	157
127	133	220	138	0	127	132	215	132	21	128	132	211	128	56
127	145	153	84	65	120	136	138	66	0	114	126	130	63	0
32	75	51	103	153	58	80	48	90	132	48	56	16	69	122
Red					Green					Blue				

Gambar 4.20 Hasil metode MSR

84.97564192	365.92659293	350.0812146	294.46341633	299.33962961
60.35808951	322.23519827	404.92159524	286.87233263	224.72586003
183.97107734	194.84527731	350.14921247	208.84352134	259.52083557
192.12646888	225.25508598	242.90656531	127.20900235	111.66255967
37.87731124	107.5558429	74.74250108	160.55118931	248.79695883
Red				

Gambar 4.21 Hasil MSRCR pada kanal warna Red (R)

218.5976925	379.8501296	361.45055621	318.89143766	313.16571632
198.68988351	337.81490747	411.58030248	301.28070904	247.82768002
204.92883649	212.98804069	346.36104144	212.98804069	29.89700043
193.66959485	220.97098317	221.1810097	103.97304919	413.04613336
96.84129025	134.98965749	80.68484225	147.84514847	209.86985886
Green				

Gambar 4.22 Hasil MSRCR pada kanal warna Green (G)

273.46132503	388.64162352	373.89165895	334.15843632	333.97203556
254.75822632	348.7522708	417.422973	316.28077577	269.29407792
219.85664534	225.68325239	344.60539205	214.98683135	105.64503147
191.09934735	206.65156422	212.26874645	105.94125018	413.04613336
82.99808932	91.16675255	24.29683137	110.14563556	195.99443975
Blue				

Gambar 4.23 Hasil MSRCR pada kanal warna Blue (B)

Dilanjutkan dengan fungsi percent clip, dimana batas bawah 1% dan batas atas 1%. Cara perhitungan fungsi percent clip dapat dilihat pada implementasi 4.2.1 Metode *Contrast Stretching*. Berikut hasil akhir metode MSRCR:

36	254	242	199	203	137	255	241	210	206	163	238	229	203	203
17	221	254	193	145	122	224	255	197	158	151	212	255	191	160
113	122	242	132	172	127	133	230	133	0	128	132	210	125	53
119	145	159	69	57	119	139	139	53	246	109	119	123	53	255
0	54	28	95	163	48	76	37	85	131	38	43	0	56	112
Red					Green					Blue				

Gambar 4.24 Data Citra Hasil akhir MSRCR

Gambar Sampel

R: 5 G: 24 B: 39	R:146 G:165 B:180	R:118 G:135 B:151	R: 62 G: 80 B: 94	R: 64 G: 77 B: 93
R: 3 G: 19 B: 32	R: 84 G:100 B:113	R:224 G:238 B:251	R: 54 G: 66 B: 78	R: 25 G: 35 B: 45
R: 14 G: 21 B: 27	R: 16 G: 23 B: 29	R:107 G:112 B:118	R: 18 G: 23 B: 27	R: 0 G: 2 B: 7
R: 14 G: 18 B: 21	R: 21 G: 25 B: 26	R: 25 G: 26 B: 28	R: 5 G: 6 B: 8	R: 3 G: 1 B: 2
R: 1 G: 5 B: 6	R: 4 G: 8 B: 7	R: 2 G: 4 B: 3	R: 8 G: 10 B: 9	R: 25 G: 23 B: 24

Hasil Retinex (MSRCR)

R: 36 G:137 B:163	R:254 G:255 B:238	R:242 G:241 B:229	R:199 G:210 B:203	R:203 G:206 B:203
R: 17 G:122 B:151	R:221 G:224 B:212	R:254 G:255 B:255	R:193 G:197 B:191	R:145 G:158 B:160
R:113 G:127 B:128	R:122 G:133 B:132	R:242 G:230 B:210	R:132 G:133 B:125	R:172 G: 0 B: 53
R:119 G:119 B:109	R:145 G:139 B:119	R:159 G:139 B:123	R: 69 G: 53 B: 53	R: 57 G:246 B:255
R: 0 G: 48 B: 38	R: 54 G: 76 B: 43	R: 28 G: 37 B: 0	R: 95 G: 85 B: 56	R:163 G:131 B:112

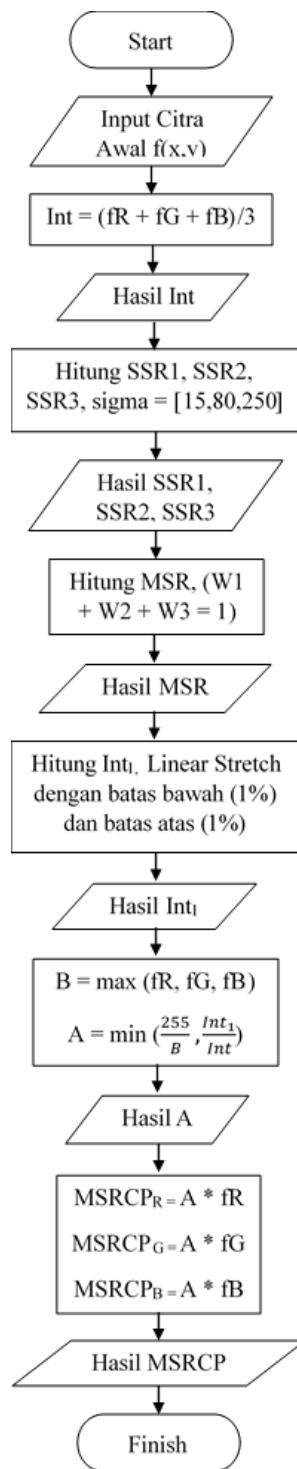
Gambar 4.25 Hasil Enhance Metode Retinex (MSRCR)

Berikut coding program untuk metode *Multiscale Retinex with Color Restoration*:

```
def Retinex_MSRCR(self):
    self.image = self.tmp
    img = self.image
    s1=0.01
    s2=0.01
    alpha=125
    img=img.astype('double')+1
    csum_log=np.log10(np.sum(img,axis=2))
    self.image=self.MultiScaleRetinex()
    pq = self.image
    msr=pq-1
    r=(np.log10(alpha*img)-csum_log[...],None)
    for i in range(r.shape[-1]):
        sort_img=np.sort(r[...,:,i],None)
        N=r[...,:,i].size
        Vmin=sort_img[int(N*s1)]
        Vmax=sort_img[int(N*(1-s2))-1]
        r[...,:,i][r[...,:,i]<Vmin]=Vmin
        r[...,:,i][r[...,:,i]>Vmax]=Vmax
        ab= (r[...,:,i]-Vmin)*255/(Vmax-Vmin)
        r[...,:,i]=ab
    return r.astype('uint8')
```

4.1.2.4 MSRCP

Multiscale Retinex with Chromaticity Preservation adalah metode lanjutan dari MSR, tidak seperti MSRCR yang memakai fungsi *color restoration*, pada metode MSRCP ini melibatkan intensitas dari setiap kanal warna tanpa fungsi *color restoration*.



Gambar 4.26 Flowchart Multiscale Retinex with Chromaticity Preservation

Tahapan algoritma MSRCP:

Algorithm 2: MSRCP algorithm

Data: I input color image; $\sigma_1, \sigma_2, \sigma_3$ the scales; s_1, s_2 the percentage of clipping pixels on each side

Result: MSRCP output color image

```

begin
    Int =  $(I_R + I_G + I_B)/3$                                 ▷ Compute the intensity channel
    foreach  $\sigma_i$  do                                         ▷ For each scale
        | Diff $_i$  = log(Int) - log(Int *  $G_{\sigma_i}$ )          ▷ Single Scale Retinex
        end
        MSR =  $\sum_i \frac{1}{3} \text{Diff}_i$                   ▷ MultiScale Retinex
        Int $_1$  = SimplestColorBalance(MSR,  $s_1, s_2$ )
        foreach pixel  $i$  do
            |  $B = \max(I_R[i], I_G[i], I_B[i])$            ▷ Compute the amplification factor
            |  $A = \min\left(\frac{255}{B}, \frac{\text{Int}_1[i]}{\text{Int}[i]}\right)$           ▷ Compute each color channel
            | MSRCP $_R[i]$  =  $A \cdot I_R[i]$ 
            | MSRCP $_G[i]$  =  $A \cdot I_G[i]$ 
            | MSRCP $_B[i]$  =  $A \cdot I_B[i]$ 
        end
    end
end

```

Gambar 4.27 Algoritma MSRCP

Proses perhitungan metode MSRCP mirip seperti proses perhitungan metode MSRCR hanya saja tahap perhitungan color restoration ditiadakan. Pada akhir proses metode MSRCP dilakukan komputasi pada setiap kanal warna.

Berikut hasil akhir dari metode MSRCP:

29	206	199	164	170	139	233	227	212	205	226	255	255	249	248
19	189	227	161	113	124	225	241	197	158	208	254	255	233	203
85	92	220	104	0	127	133	230	133	12	164	168	242	156	42
92	118	133	44	0	118	140	138	53	0	138	146	149	71	0
8	35	12	68	140	40	71	24	85	129	48	62	18	76	134
Red					Green					Blue				

Gambar 4.28 Data Citra Hasil akhir MSRCP

Gambar Sampel

R: 5 G: 24 B: 39	R: 146 G: 165 B: 180	R: 118 G: 135 B: 151	R: 62 G: 80 B: 94	R: 64 G: 77 B: 93
R: 3 G: 19 B: 32	R: 84 G: 100 B: 113	R: 224 G: 238 B: 251	R: 54 G: 66 B: 78	R: 25 G: 35 B: 45
R: 14 G: 21 B: 27	R: 16 G: 23 B: 29	R: 107 G: 112 B: 118	R: 18 G: 23 B: 27	R: 0 G: 2 B: 7
R: 14 G: 18 B: 21	R: 21 G: 25 B: 26	R: 25 G: 26 B: 28	R: 5 G: 6 B: 8	R: 3 G: 1 B: 2
R: 1 G: 5 B: 6	R: 4 G: 8 B: 7	R: 2 G: 4 B: 3	R: 8 G: 10 B: 9	R: 25 G: 23 B: 24

Hasil Retinex (MSRCP)

R: 29 G:139 B:226	R:206 G:233 B:255	R:199 G:227 B:255	R:164 G:212 B:249	R:170 G:205 B:248
R: 19 G:124 B:208	R:189 G:227 B:254	R:227 G:241 B:255	R:161 G:197 B:233	R:113 G:158 B:203
R: 85 G:127 B:164	R: 92 G:133 B:168	R:220 G:230 B:242	R:104 G:133 B:156	R: 0 G: 12 B: 42
R: 92 G:118 B:138	R:118 G:140 B:146	R:133 G:138 B:149	R: 44 G: 53 B: 71	R: 0 G: 0 B: 0
R: 8 G: 40 B: 48	R: 35 G: 71 B: 62	R: 12 G: 24 B: 18	R: 68 G: 85 B: 76	R: 140 G:129 B:134

Gambar 4.29 Hasil Enhance Metode Retinex (MSRCP)

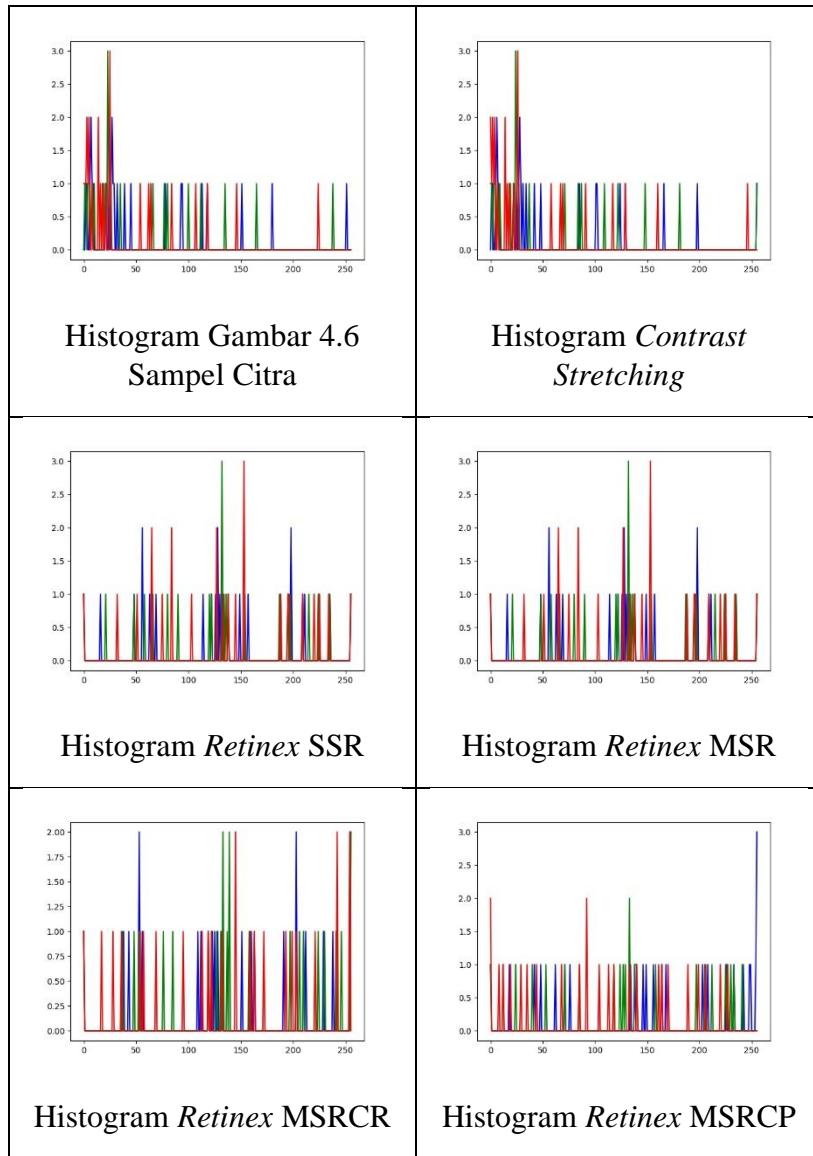
Berikut *coding* program untuk metode *Multiscale Retinex with Chromaticity Preservation*:

```
def Retinex_MSRCP(self):
    self.image = self.tmp
    img = self.image
    eps=np.finfo(np.double).eps
    sigmas=[12,80,250]
    s1=0.01
    s2=0.01
    Int=np.sum(img, axis=2)/3
    Diffs=[]
    for sigma in sigmas:
        #kernel gaussian
        ksize=int(np.floor(sigma*6)/2)*2+1
        k_1D=np.arange(ksize)-ksize//2
        k_1D=np.exp(-k_1D**2/(2*sigma**2))
        k_1D=k_1D/np.sum(k_1D)
        qr = k_1D
        #gaussian blur
        row_filter= qr
        t=cv2.filter2D(Int,-1,row_filter[...],None)
        tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
        Diffs.append(np.log10(Int+1)-np.log10(tr+1))
    MSR=sum(Diffs)/3
    sort_img=np.sort(MSR,None)
    N=MSR.size
    Vmin=sort_img[int(N*s1)]
    Vmax=sort_img[int(N*(1-s2))-1]
    MSR[MSR<Vmin]=Vmin
    MSR[MSR>Vmax]=Vmax
    ab= (MSR-Vmin)*255/(Vmax-Vmin)
    Int1= ab
    B=np.max(img, axis=2)
    A=np.min(np.stack((255/(B+eps), Int1/(Int+eps)), axis=2), axis=-1)
    return (A[...],None]*img).astype('uint8')
```

4.1.3 Implementasi Histogram

Berikut hasil histogram dari masing – masing proses metode:

Tabel 4.8 Histogram



4.1.4 Implementasi MSE dan PSNR

Berikut ini adalah contoh perhitungan mencari nilai MSE dan PSNR:

Tabel 4.9 Contoh MSE dan PSNR

7	5	1	1	4	2
2	4	6	3	1	3

<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>1</td><td>0</td><td>1</td></tr> </table> <p>Citra Awal</p>	1	0	1	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>2</td><td>4</td><td>1</td></tr> </table> <p>Citra Hasil</p>	2	4	1
1	0	1					
2	4	1					

$$MSE = \frac{(7*1) + (5*4) + (1*2) + (2*3) + (4*1) + (6*3) + (1*2) + (0*4) + (1*1)}{3*3} = 6,6667$$

$$PSNR = 10 * \log_{10} \frac{7^2}{6,6667} = 8,662$$

Gambar 4.30 Contoh Perhitungan MSE dan PSNR

Berikut Tabel perhitungan MSE dan PSNR citra hasil dari masing – masing metode yang digunakan pada penelitian:

Tabel 4.10 MSE dan PSNR

No.	Nama Metode	Nilai MSE	Nilai PSNR
1.	<i>Contrast Stretching</i>	0.0006329873125720876	31.98604994777515
2.	<i>Retinex (SSR)</i>	0.13405469691144434	8.727179649187995
3.	<i>Retinex (MSR)</i>	0.13405469691144434	8.727179649187995
4.	<i>Retinex (MSRCR)</i>	0.16871623734461108	7.728431186567526
5.	<i>Retinex (MSRCP)</i>	0.14525449186210432	8.378704285709407

4.1.5 Rancangan Interface Program Aplikasi

Sistem program aplikasi dibangun dengan menggunakan Bahasa pemrograman Python 3 dengan bantuan Library PyQt5, OpenCV, Numpy dan Matplotlib. GUI (*Grafical User Interface*) sistem penelitian dibangun menggunakan Qt Designer. Rancangan Interface program aplikasi disesuaikan dengan kebutuhan yang diperlukan. Rancangan Interface terdiri dari 2 *window*. *Window* pertama berisi halaman loading screen dan *window* kedua berisi halaman Home, Work dan About. Berikut tampilan rancangan sistem:

4.1.5.1 Rancangan Interface Loading Screen

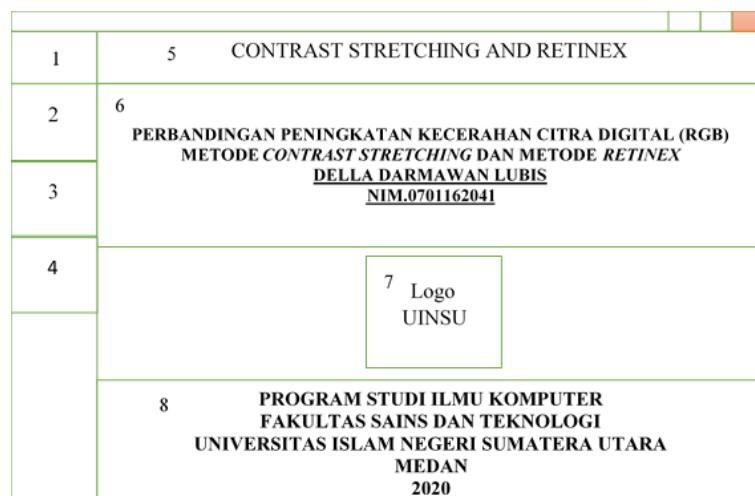


Gambar 4.31 Rancangan Tampilan Loading Screen

Keterangan:

1. Merupakan QLabel yang berisi teks Judul
2. Merupakan QLabel yang berisi teks Deskripsi
3. Merupakan QProgressBar
4. Merupakan QLabel berisi teks “loading”
5. Merupakan QMainWindow untuk loading screen

4.1.5.2 Rancangan Interface Home



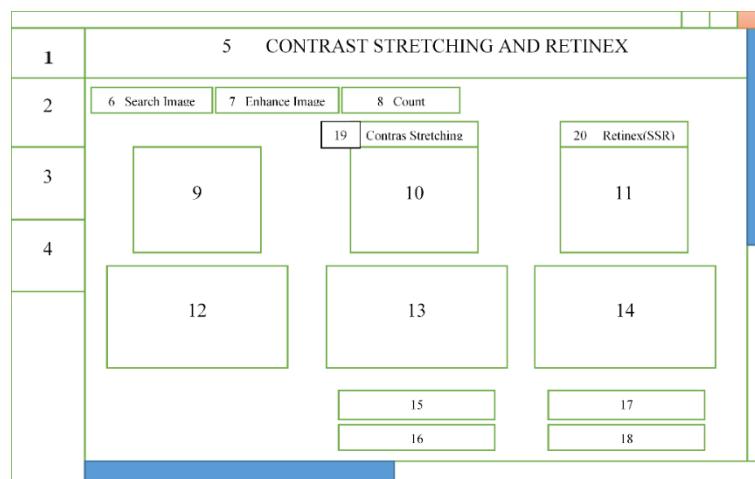
Gambar 4.32 Rancangan Tampilan Halaman Home

Keterangan:

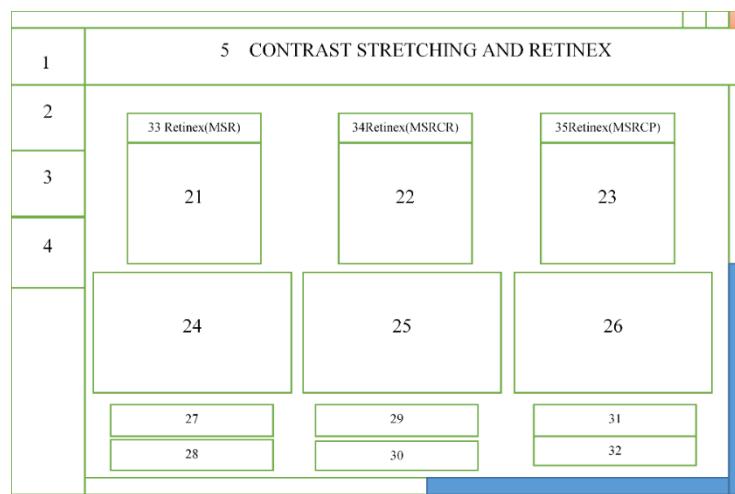
1. Merupakan QPushButton ToggleMenu
2. Merupakan QPushButton menu Home
3. Merupakan QPushButton menu Work

4. Merupakan QPushButton menu About
5. Merupakan QLabel nama Metode
6. Merupakan QLabel yang berisi Judul, Nama dan NIM
7. Merupakan QLabel yang berisi gambar logo UINSU
8. Merupakan QLabel yang berisi Program studi, Fakultas, Universitas.

4.1.5.3 Rancangan Interface Work



Gambar 4.33 Rancangan Tampilan Interface Halaman Work (1)



Gambar 4.34 Rancangan Tampilan Interface Halaman Work (2)

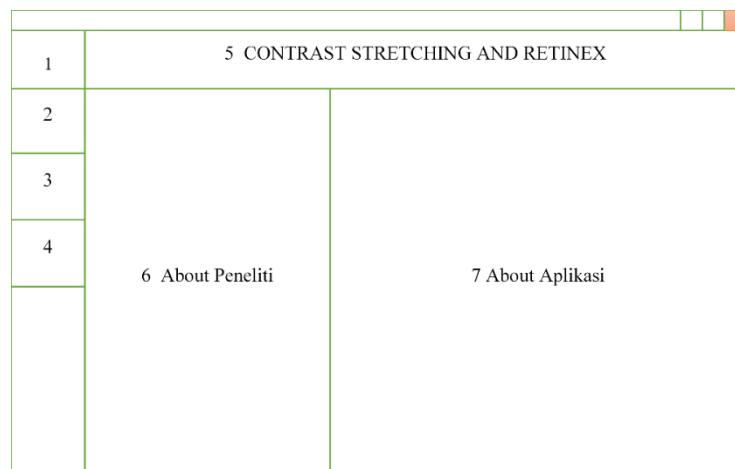
Keterangan:

1. Merupakan QPushButton ToggleMenu

2. Merupakan QPushButton menu Home
3. Merupakan QPushButton menu Work
4. Merupakan QPushButton menu About
5. Merupakan QLabel nama Metode
6. Merupakan QPushButton untuk membuka gambar
7. Merupakan QPushButton untuk memproses gambar
8. Merupakan QPushButton untuk menghitung nilai MSE dan PSNR
9. Merupakan QLabel untuk menampilkan gambar Input
10. Merupakan QLabel untuk menampilkan gambar *Contrast Stretching*
11. Merupakan QLabel untuk menampilkan gambar *Retinex* (SSR)
12. Merupakan QLabel untuk menampilkan gambar histogram gambar Input
13. Merupakan QLabel untuk menampilkan gambar histogram gambar *Contrast Stretching*
14. Merupakan QLabel untuk menampilkan gambar histogram gambar *Retinex* (SSR)
15. Merupakan QLabel untuk menampilkan nilai MSE *Contrast Stretching*
16. Merupakan QLabel untuk menampilkan nilai PSNR *Contrast Stretching*
17. Merupakan QLabel untuk menampilkan nilai MSE *Retinex* (SSR)
18. Merupakan QLabel untuk menampilkan nilai PSNR *Retinex* (SSR)
19. Merupakan QLabel untuk nama *Contrast Stretching*
20. Merupakan QLabel untuk nama *Retinex* (SSR)
21. Merupakan QLabel untuk menampilkan gambar *Retinex* (MSR)
22. Merupakan QLabel untuk menampilkan gambar *Retinex* (MSRCR)
23. Merupakan QLabel untuk menampilkan gambar *Retinex* (MSRCP)
24. Merupakan QLabel untuk menampilkan gambar histogram gambar *Retinex* (MSR)
25. Merupakan QLabel untuk menampilkan gambar histogram gambar *Retinex* (MSRCR)
26. Merupakan QLabel untuk menampilkan gambar histogram gambar *Retinex* (MSRCP)
27. Merupakan QLabel untuk menampilkan nilai MSE *Retinex* (MSR)

28. Merupakan QLabel untuk menampilkan nilai PSNR *Retinex* (MSR)
29. Merupakan QLabel untuk menampilkan nilai MSE *Retinex* (MSRCR)
30. Merupakan QLabel untuk menampilkan nilai PSNR *Retinex* (MSRCR)
31. Merupakan QLabel untuk menampilkan nilai MSE *Retinex* (MSRCP)
32. Merupakan QLabel untuk menampilkan nilai PSNR *Retinex* (MSRCP)
33. Merupakan QLabel untuk nama *Retinex* (MSR)
34. Merupakan QLabel untuk nama *Retinex* (MSRCR)
35. Merupakan QLabel untuk nama *Retinex* (MSRCP)

4.1.5.4 Rancangan Interface About



Gambar 4.35 Rancangan Tampilan Halaman About

Keterangan:

1. Merupakan QPushButton ToggleMenu
2. Merupakan QPushButton menu Home
3. Merupakan QPushButton menu Work
4. Merupakan QPushButton menu About
5. Merupakan QLabel nama Metode
6. Merupakan QLabel berisi nama, nim, program studi, fakultas dan universitas penulis

Merupakan QLabel berisi tentang dan penggunaan aplikasi.

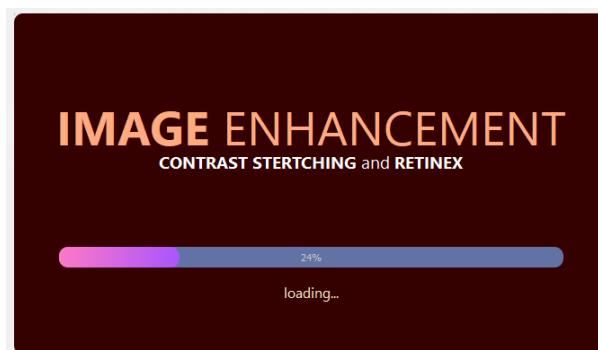
4.2 Pengujian Sistem

Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan. Pengujian sistem dilakukan untuk mendapatkan citra hasil dari proses metode – metode yang digunakan. Adapun kriteria pengujian sebagai berikut:

1. Citra awal (input) yang digunakan adalah citra yang berekstensi .JPG dan .PNG.
2. Program aplikasi mampu menampilkan citra awal dan histogram citra awal tersebut.
3. Program aplikasi mampu memproses citra awal dengan menggunakan metode *Contrast Stretching* dan metode *Retinex* dan menampilkan citra hasil masing – masing metode.
4. Program aplikasi mampu menampilkan histogram dari masing – masing citra hasil.
5. Program aplikasi mampu menghitung nilai mse dan psnr dari masing – masing citra hasil.
6. Citra hasil dan histogram dari masing – masing metode dapat disimpan dengan ekstensi .JPG dan .PNG.

4.2.1 Interface Loading Screen

Tampilan loading screen muncul pertama sekali pada saat program aplikasi dijalankan. Berikut tampilan loading screen:



Gambar 4.36 Interface Loading Screen

4.2.2 Interface Halaman Home

Tampilan halaman Home adalah tampilan yang muncul setelah loading screen selesai. Berikut tampilan halaman home:

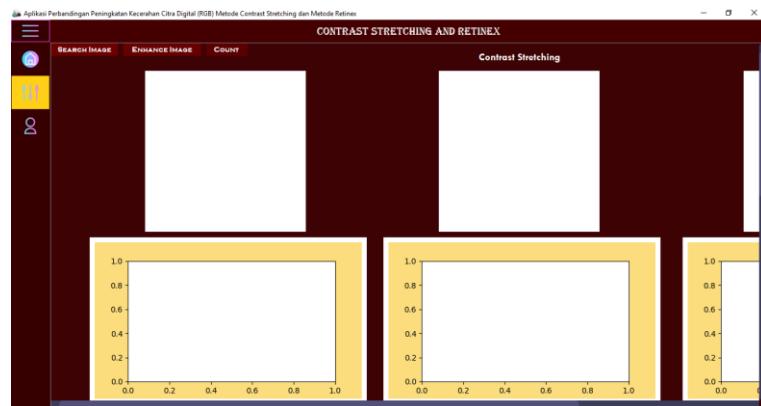


Gambar 4.37 Interface Halaman Home

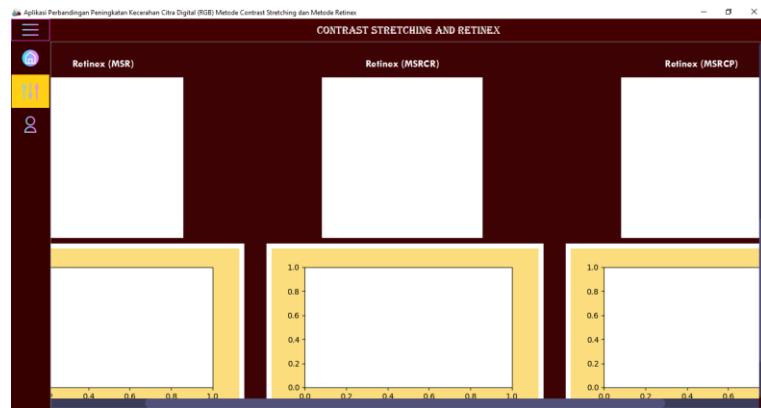
Pada sudut kiri atas terdapat tombol menu yang berisi tombol Home, tombol Work halaman kerja dan tombol About.

4.2.3 Interface Halaman Work

Tampilan halaman Work atau kerja muncul jika *user* mengklik tombol Work pada bagian menu. Berikut tampilan halaman Work:



Gambar 4.38 Interface Halaman Work (1)

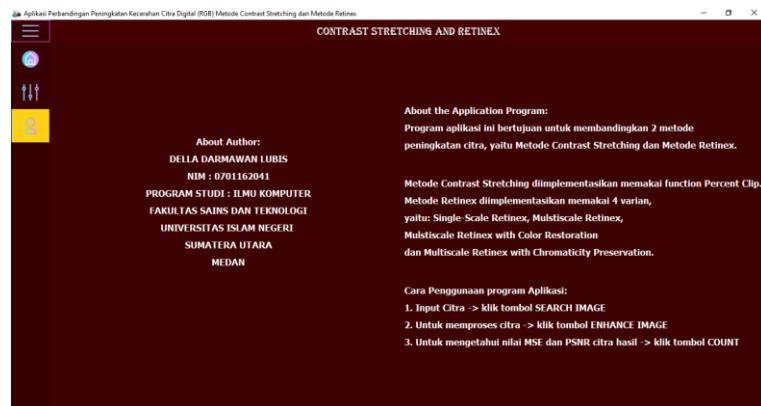


Gambar 4.39 Interface Halaman Work (2)

Pada halaman kerja terdapat tiga buah tombol, yaitu tombol Search Image, tombol Enhance Image dan tombol Count. Dimana tombol Search Image adalah tombol untuk membuka atau mencari citra awal (input), tombol Enhance Image adalah tombol proses kerja metode dan tombol Count adalah tombol untuk menghitung nilai mse dan psnr dari masing – masing metode.

4.2.4 Interface Halaman About

Tampilan halaman About akan muncul jika *user* mengklik tombol About. Berikut tampilan halaman About:



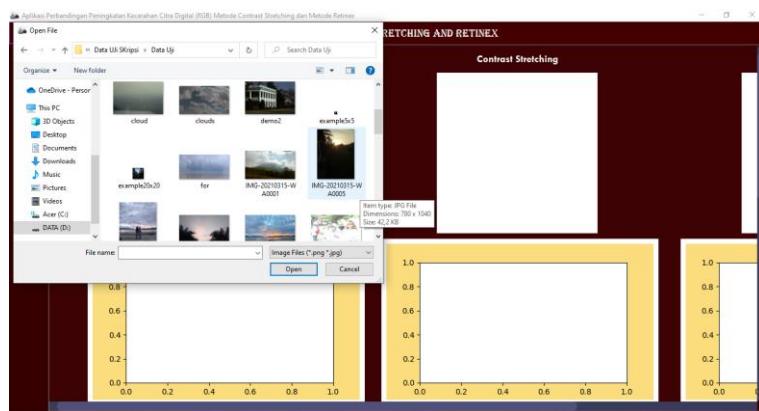
Gambar 4.40 Interface Halaman About

Halaman ini berisi tentang informasi peneliti dan informasi program aplikasi.

4.2.5 Pengujian Program Aplikasi

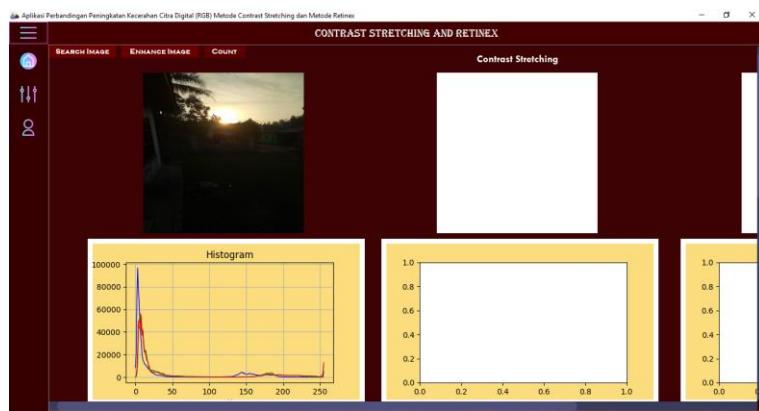
Pengujian Program Aplikasi dilakukan pada citra berformat .jpg dan .png. Untuk melakukan proses *Enhance Image*, langkah awal yang dilakukan adalah memilih file citra yang akan diproses.

User dapat memilih citra dengan mengklik tombol *Search Image*. Selanjutnya akan muncul window untuk memilih file citra yang akan diproses. Gambar yang dapat terbaca hanya gambar yang berformat .jpg dan .png saja, jenis ekstensi gambar lainnya tidak dapat diproses. Tampilan window input image sebagai berikut:



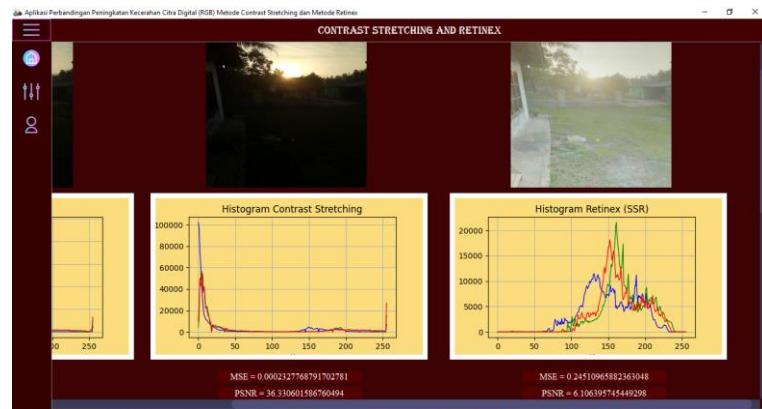
Gambar 4.41 Tampilan Window Input Citra

Setelah user memilih citra yang akan diproses maka citra awal (input) yang dibuka akan ditampilkan pada qlabel input citra beserta histogramnya sebagai berikut:

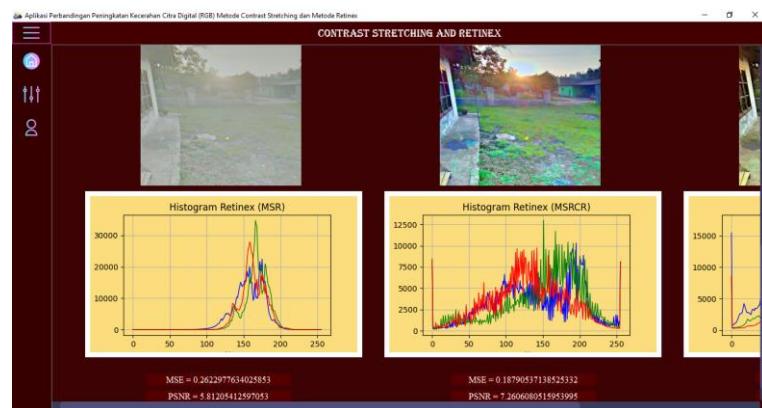


Gambar 4.42 Tampilan File Citra dan Histogram

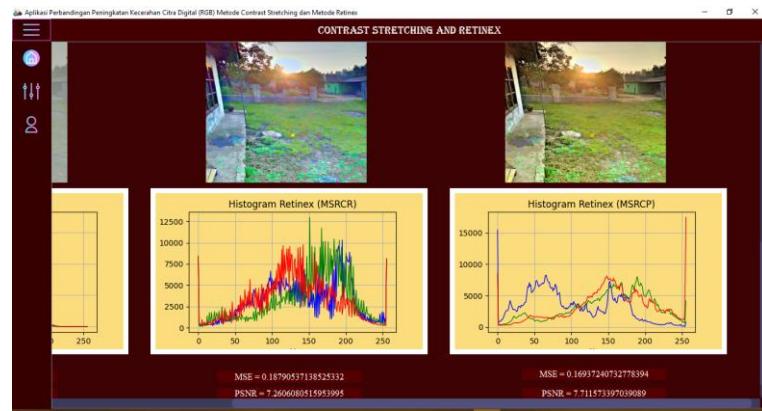
Langkah berikutnya adalah memproses citra awal dengan *user* mengklik tombol *Enhance Image*, maka citra akan secara otomatis diproses dengan metode – metode yang digunakan pada penelitian. Proses perhitungan nilai mse dan psnr citra hasil juga dapat langsung dilakukan dengan cara mengklik tombol *Count*, dimana hasil perhitungan akan muncul pada qlabel hasil yang terletak dibawah tampilan histogram masing – masing citra hasil. Berikut tampilannya:



Gambar 4.43 Tampilan Pengujian Metode (1)



Gambar 4.44 Tampilan Pengujian Metode (2)



Gambar 4.45 Tampilan Pengujian Metode (3)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini bertujuan untuk membandingkan hasil kerja dari Metode *Contrast Stretching* dan Metode *Retinex* pada citra digital RGB. Berdasarkan dari hasil implementasi dan pengujian dari program aplikasi, maka simpulan yang dapat diperoleh pada penelitian ini adalah sebagai berikut:

1. Metode *Contrast Stretching* dan Metode *Retinex*, keduanya dapat bekerja dengan baik pada citra warna RGB.
2. Metode *Contrast Stretching* dan Metode *Retinex* berhasil melakukan proses peningkatan kecerahan citra pada citra yang memiliki kontras rendah, kontras tinggi, citra berkabut, citra terlalu gelap, citra terlalu terang dan citra underwater.
3. Pada citra dengan kondisi terlalu gelap, metode *Retinex* sangat baik dalam menangani citra tersebut dan menghasilkan citra yang terang dan jelas, sedangkan Metode *Contrast Stretching* hanya menghasilkan sedikit peningkatan kecerahan.
4. Pada citra dengan kondisi terlalu terang, metode *Contrast Stretching* dan metode *Retinex* keduanya menghasilkan citra yang baik dan jelas, namun pada varian *Retinex* SSR dan MSR menghasilkan citra yang terlalu terang.
5. Pada citra dengan kondisi kontras rendah, metode *Contrast Stretching* menghasilkan citra yang jelas dan tajam, sedangkan pada metode *Retinex*, khususnya pada varian SSR dan MSR menghasilkan citra cenderung kabur.
6. Pada citra dengan kondisi kontras tinggi, metode *Retinex* menghasilkan citra yang terang dan jelas, sedangkan metode *Contrast Stretching* citra yang dihasilkan tidak terlalu cerah.
7. Pada citra underwater, metode *Retinex* menghasilkan citra dan warna yang jelas dan terang, namun pada varian *Retinex* MSRCP tidak menghasilkan

citra yang baik, sedangkan metode *Contrast Stretching* tidak menunjukkan perubahan.

8. Berdasarkan size citra hasil, metode *Contrast Stretching* menghasilkan size yang lebih kecil dari pada citra hasil metode *Retinex*.
9. Berdasarkan nilai mse dan psnr, metode *Contrast Stretching* menghasilkan nilai mse yang lebih kecil dari pada citra hasil metode *Retinex*. Yang dimana nilai citra hasil metode *Contrast Stretching* sangat mendekati nilai 0. Sedangkan, pada nilai psnr metode *Retinex* menghasilkan nilai yang lebih kecil dari pada nilai citra hasil metode *Contrast Stretching*.
10. Berdasarkan penyebaran histogram citra, penyebaran nilai metode *Contrast Stretching* lebih merata dibandingkan metode *Retinex*.

Dari beberapa simpulan diatas, penulis memberikan rekomendasi untuk citra yang memiliki kontras yang rendah, gunakan Metode *Contrast Stretching* untuk memproses citra tersebut. Jika menggunakan citra yang berkondisi terlalu gelap, kontras tinggi dan citra *underwater*, gunakanlah Metode *Retinex* untuk memproses citra tersebut.

5.2 Saran

Penelitian ini menggunakan metode peningkatan kontras linear yaitu metode *Contrast Stretching*, dan varian metode *Retinex* hanya sampai *Retinex MSRPC* saja, penulis menyarankan untuk melakukan pengembangan perbandingan menggunakan metode lainnya seperti:

1. Menggunakan metode Non-Linear seperti *Adaptive Histogram Equalization*
2. Menggunakan varian metode *Retinex* yang lainnya seperti AMSR (*Automated Multiscale Retinex*).

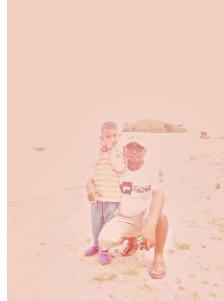
DAFTAR PUSTAKA

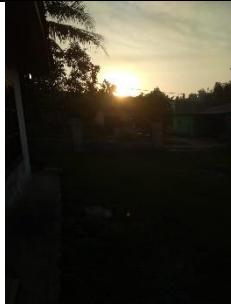
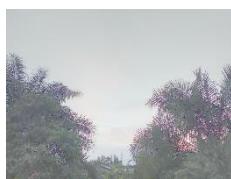
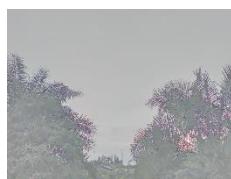
- Aprilia, Rani. 2018. “Mini Project: Eksplorasi Library Python “OpenCV””, <https://medium.com/@rani.apriliaa/mini-project-eksplorasi-library-python-opencv-caec7b13a872> diakses pada 07 September 2020 pukul 13:59 WIB.
- Aripin, Soeb., Leonarde, Guidio, Ginting., Natalia, Silalahi. 2017. “Penerapan metode *retinex* untuk meningkatkan kecerahan citra pada hasil *screenshot*” dalam Media Informatika Budidarma, Volume 1, Nomor 1. Medan: STMIK Budi Darma.
- Azmi, Fadhillah., Sherly., Sirano, Lahagu., David. 2019. “Implementasi Metode *Retinex* dan *Histogram Equalization* Pada Kecerahan Citra Digital” dalam *Journal of Informatics and Telecommunication Engineering*, Volume 2, Nomor 2. Medan: Universitas Prima Indonesia.
- Banjarnahor, Martha. 2018. “Reduksi *Noise Salt and Paper* Pada Citra Pankromatik Menggunakan Metode *Harmonic Mean Filter*” dalam Jurnal Pelita Informatika, Volume 17, Nomor 3. Medan: STMIK Budi Darma.
- Belen, Ana, Petro., Catalina, Sbret., Jean-Michel, Morel. 2014. “*Multiscale Retinex*” dalam IPOL (*Image Processing On Line*). <https://www.ipol.im/pub/art/2014/107/> diakses pada 20 Juli 2020 pukul 12:17 WIB.
- Dinata, Sylvia. 2014. “Implementasi Metode *Multiscale Retinex* Untuk *Image Enhancement*” dalam *Skripsi*. Medan: Universitas Sumatera Utara.
- Dosensosiologi.com. 2018. ““Kerangka Penelitian” Pengertian dan Contohnya Lengkap”, <https://dosensosiologi.com/kerangka-penelitian-pengertian-dan-contohnya-lengkap/> diakses pada 14 Juni 2020 pukul 12:33 WIB.
- Dwi, IB, Satria, Kusuma. 2018. “Pengolahan Citra Digital : Jenis-jenis Citra”, <https://blogdskusuma.wordpress.com/2018/02/23/pengolahan-citra-digital-jenis-jenis-citra/> diakses pada 27 Juni 2020 pukul 21:16 WIB.
- Fatmawati, Diana. 2011. “Implementasi Metode Peregangan Kontras (*Contrast Stretching*) Untuk Memperbaiki Kualitas Citra” dalam *Skripsi*. Medan: Universitas Sumatera Utara.
- Furqan, Mhd., Sriani., Kartika, Yuli, Siregar. 2020. “Perbandingan Algoritma Contraharmonic Mean Filter Dan Arithmetic Mean Filter untuk Mereduksi Exponential Noise” dalam jurnal JISKA, Volume 5, Nomor 2. Medan: Universitas Islam Negeri Sumatera Utara.
- Harahap, Bainun. 2018. “Implementasi Metode *Retinex* Untuk Meningkatkan Kualitas Citra Underwater” dalam jurnal KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), Volume 2, Nomor 1. Medan: STMIK Budi Darma.
- Hermawati, Fajar Astuti. 2013. *Pengolahan Citra Digital Konsep & Teori*. Yogyakarta: Andi Publisher.
- Kadir, Abdul. 2013. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi Publisher.

- Ketutrare. 2014. “Pengertian MSE dan PSNR pada Citra Digital dan Contoh Perhitungannya”, <https://www.ketutrare.com/2014/07/pengertian-mse-dan-psnr-pada-citra.html> diakses pada 3 Juni 2020 pukul 20:35 WIB.
- Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika Bandung.
- Murinto., Eko, Aribowo., Elena, Yustina. 2009. “Implementasi Metode *Retinex* Untuk Pencerahan Citra” dalam Jurnal Informatika Volume 3, Nomor 2. Yogyakarta: Universitas Ahmad Dahlan.
- Nurliadi., Poltak, Sihombing., Marwan, Ramli. 2016. “Analisis *Contrast Stretching* Menggunakan Algoritma *Euclidean* Untuk Meningkatkan Kontras Pada Citra Berwarna” dalam Jurnal Teknologi, Volume 03, Nomor 1. Medan: Universitas Sumatera Utara.
- Pamungkas, Adi. 2015. “Pengolahan Citra”, <https://pemrogramanmatlab.com/pengolahan-citra-digital/> diakses pada 27 juni 2020 pukul 21:22 WIB.
- Pamungkas, Adi. 2017. “Cara Menghitung Nilai MSE, RMSE, dan PSNR pada Citra Digital”, <https://pemrogramanmatlab.com/2017/06/04/cara-menghitung-nilai-mse-rmse-dan-psnr-pada-citra-digital/> diakses pada 3 Juli 2020 pukul 20:48 WIB.
- Prasetyo, Eko. 2011. *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Andi Publisher.
- Purba, Bister. 2017. “Aplikasi Perbaikan Kualitas Citra Hasil Penginderaan Jauh (*Remote Sensing*) Dengan Metode *Contrast Stretching*” dalam Jurnal *Times Technology informatics & computer system*, Volume VI, Nomor 2. Medan: STMIK Budi Darma.
- Sidik., Firmansyah., Syaiful, Anwar. 2019. “Perbaikan Citra Malam (Tidak Infrared) Dengan Metode *Histogram Equalization* Dan *Contrast Stretching*” dalam Jurnal Ilmu Pengetahuan Dan Teknologi Komputer, Volume 4, Nomor 2. Jakarta: STMIK Nusa Mandiri.
- Simbolon, Hotmantri. 2015. “Penerapan Metode Contrast Stretching Untuk Peningkatan Kualitas Citra Bidang Biomedis” dalam Jurnal Mantik Penusa, Volume 18, Nomor 2. Medan: STMIK Pelita Nusantara.
- Sutoyo. T., Suhartono, Dr. Vincent., Nurhayati, O.D. & Wijanarto. M. T. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Andi Publisher.
- Sriani, and M. Iksan, “Implementasi Kompresi Citra Digital Menggunakan Algoritma Wavelet,” Konferensi Nasional Pengembangan Teknologi Informasi dan Komunikasi, pp. 258–266, 2016.

LAMPIRAN – LAMPIRAN
LAMPIRAN A
HASIL PENGUJIAN APLIKASI

1. Hasil Perbandingan Citra Awal dan Citra Hasil Proses

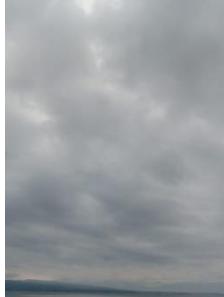
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
1.	 IMG-WA014.jpg					
2.	 IMG-WA001.jpg					

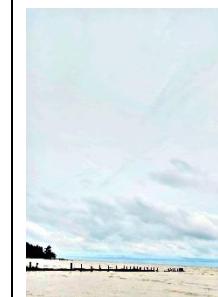
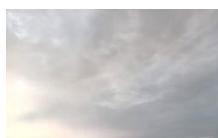
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
3.						
4.						
5.						
6.						

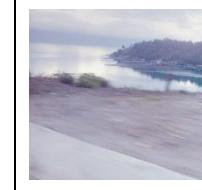
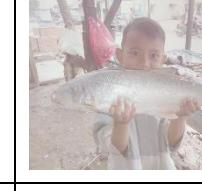
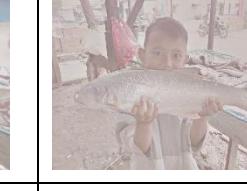
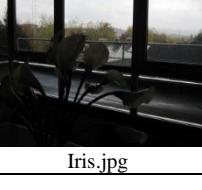
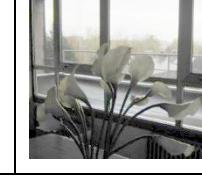
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
7.	 IMG-WA029.jpg					
8.	 IMG-WA025.jpg					
9.	 IMG-WA030.jpg					

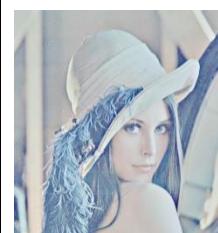
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
10.	 IMG-WA028.jpg					
11.	 IMG-WA012.jpg					
12.	 IMG-WA023.jpg					

No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
13.	 IMG-WA024.jpg					
14.	 IMG-WA026.jpg					
15.	 IMG-WA006.jpg					

No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
16.	 IMG-WA015.jpg					
17.	 IMG-WA017.jpg					
18.	 IMG-WA018.jpg					

No.	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
19.	 IMG-WA019.jpg					
20.	 IMG-WA020.jpg					
21.	 IMG-WA033.jpg					

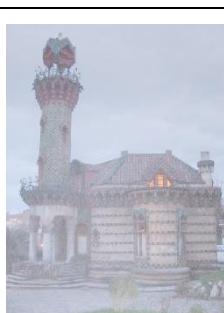
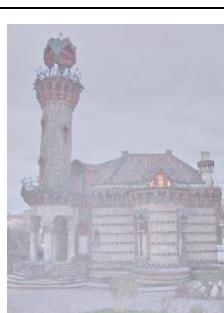
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
22.	 IMG-WA021.jpg					
23.	 IMG-WA031.jpg					
24.	 IMG-WA027.jpg					
25.	 Iris.jpg					

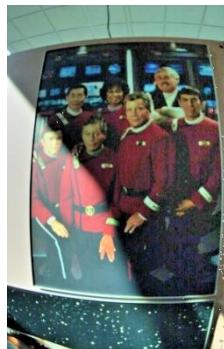
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
26.	 For.jpg					
27.	 19.jpg					
28.	 Lena.png					
29.	 25.jpg					
30.	 10.jpg					

No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
31.	 22.jpg					
32.	 Tes13.png					
33.	 14.jpg					
34.	 City.jpg					
35.	 Clouds.jpg					

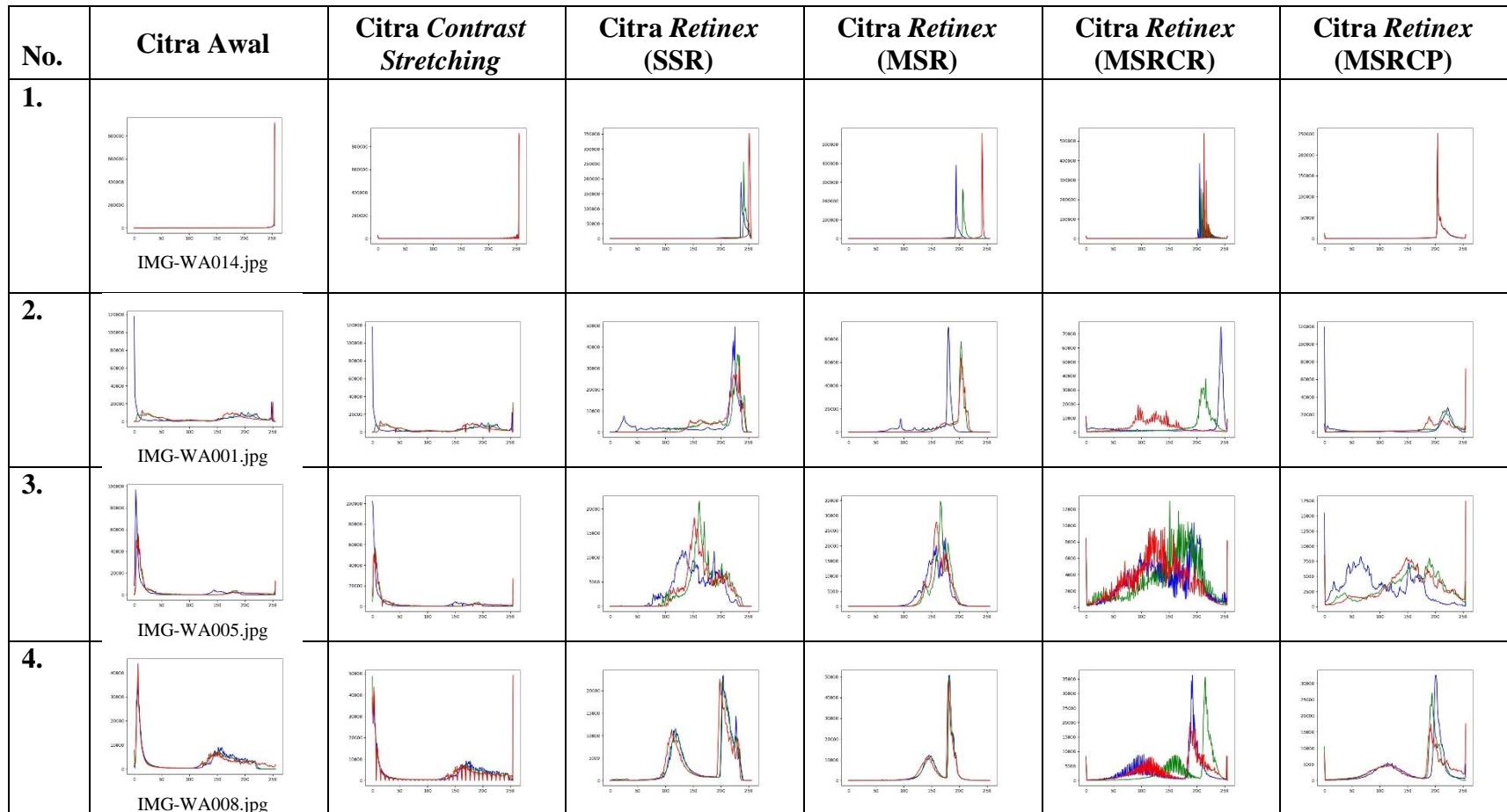
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
36.	 Alone.jpg					
37.	 Test1.png					
38.	 Test4.png					
39.	 Tree.jpg					
40.	 Origin.jpg					

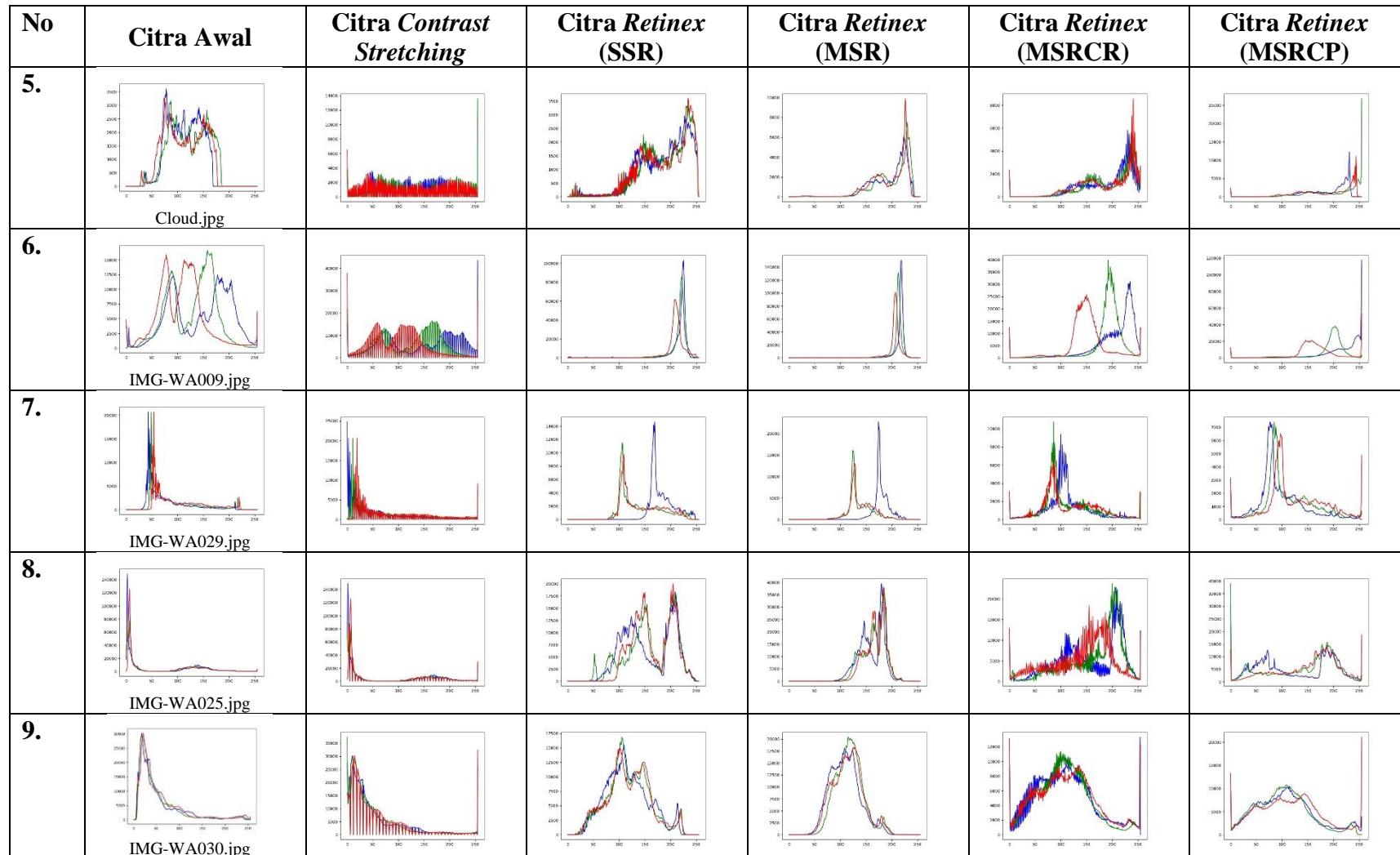
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
41.						
42.						
43.						
44.						
45.						

No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
46.	 Test12.png					
47.	 Test16.png					
48.	 Test17.png					

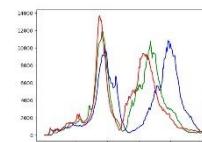
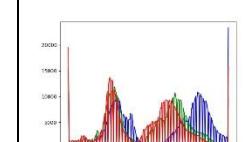
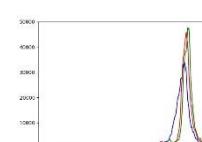
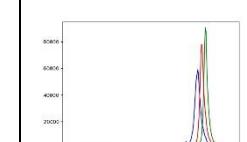
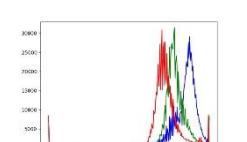
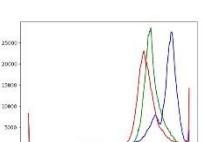
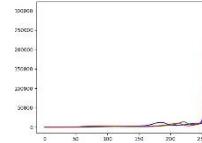
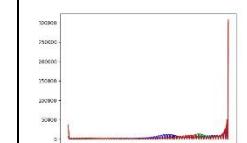
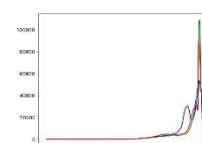
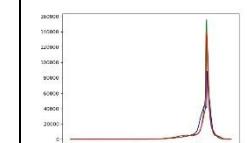
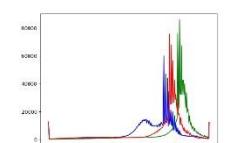
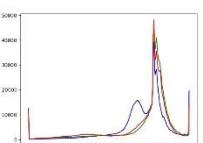
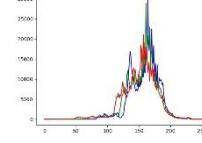
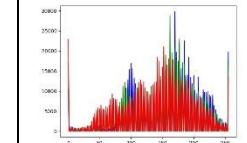
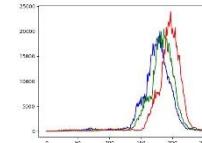
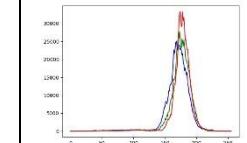
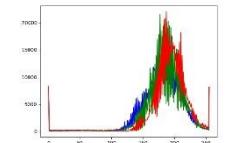
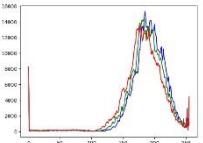
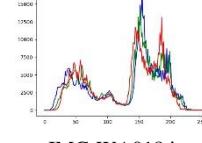
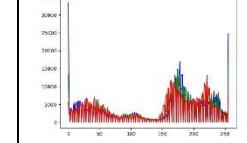
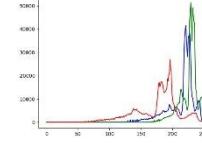
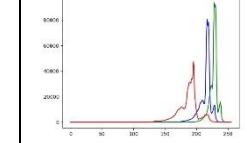
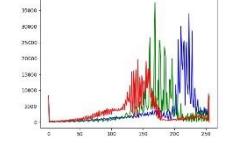
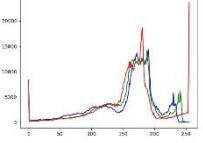
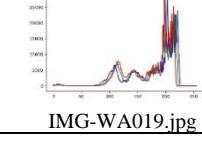
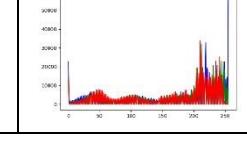
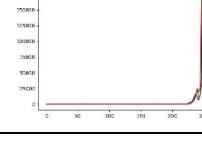
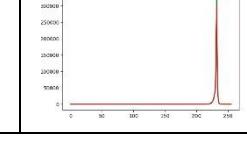
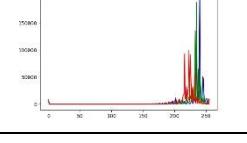
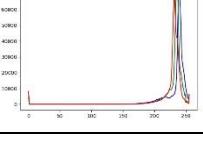
No.	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
49.						
50.						

2. Perbandingan Histogram Citra Awal dan Citra Hasil

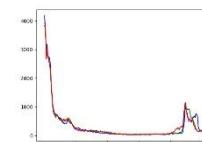
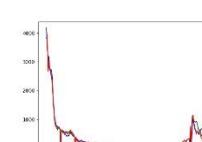
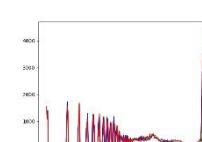
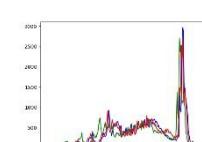
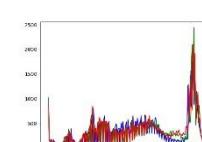
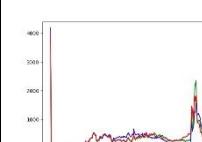
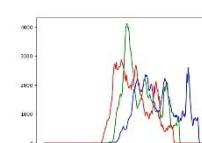
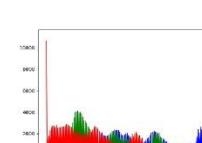
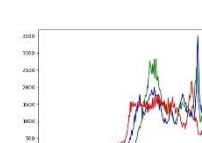
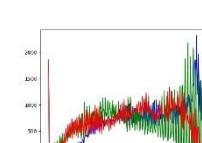
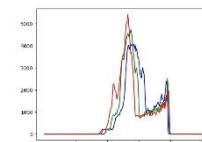
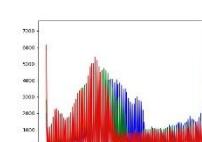
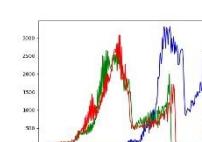
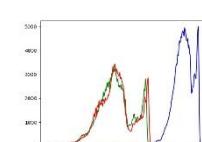
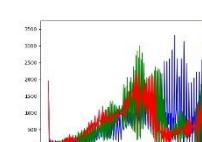
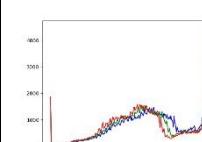
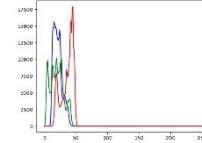
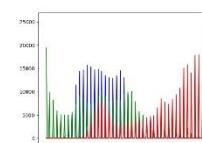
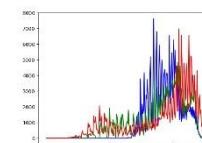
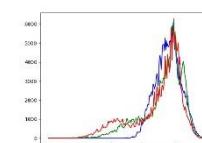
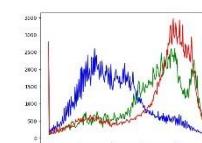
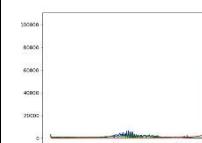
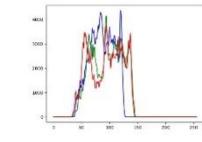
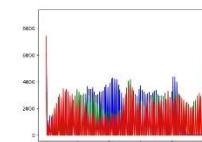
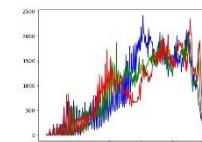
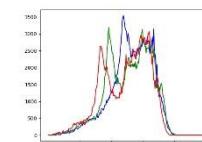
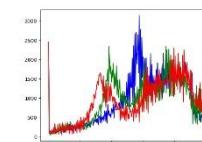
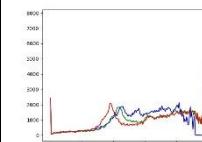


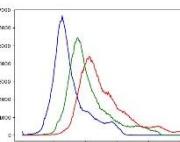
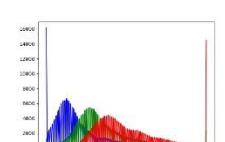
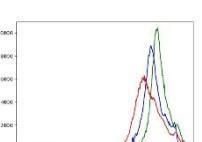
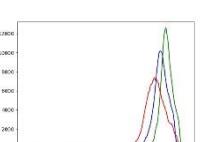
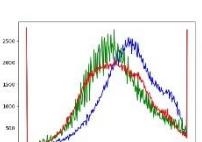
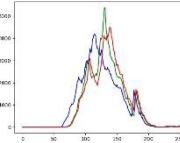
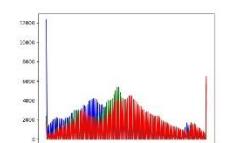
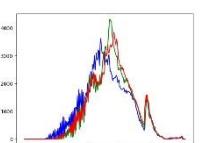
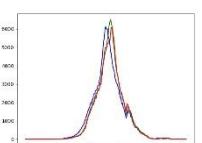
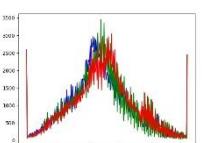
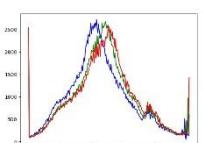
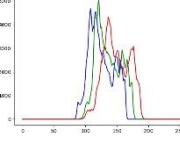
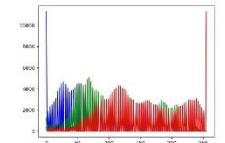
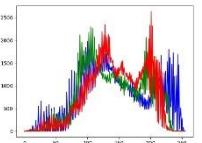
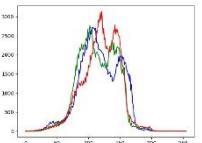
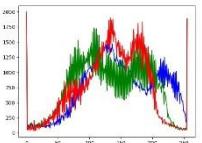
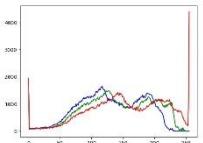
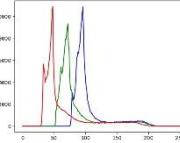
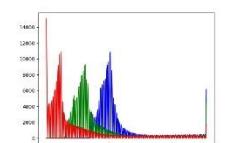
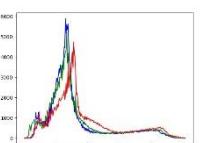
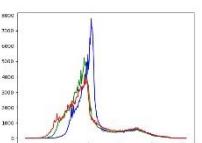
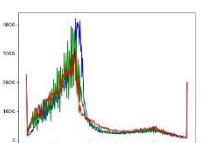
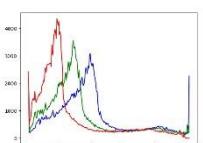
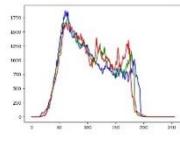
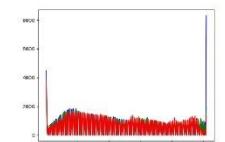
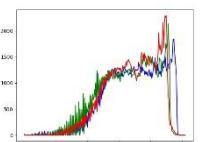
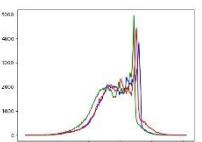
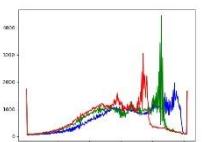
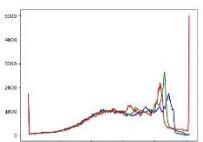


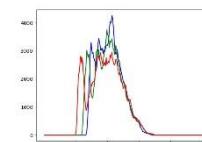
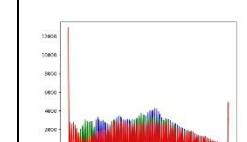
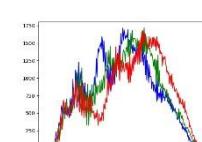
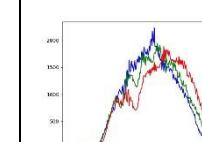
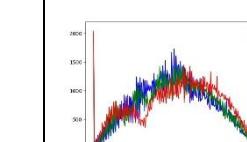
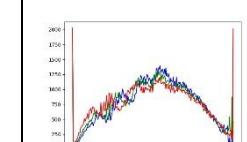
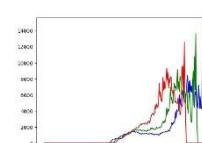
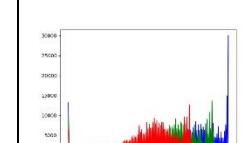
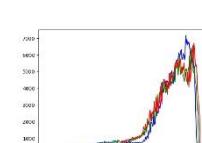
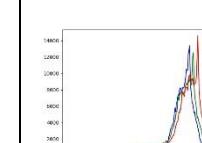
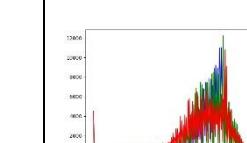
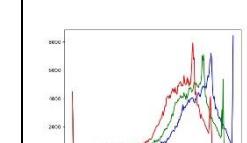
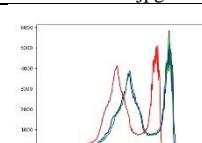
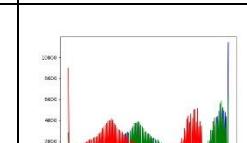
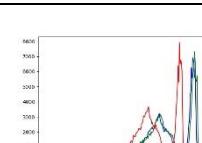
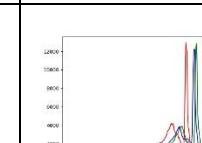
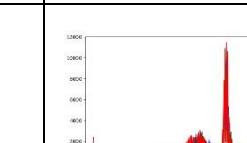
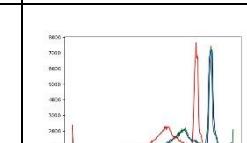
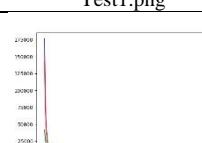
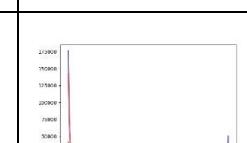
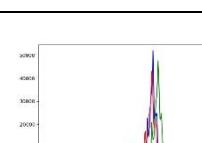
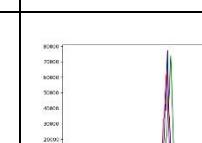
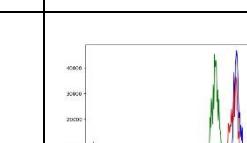
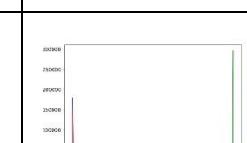
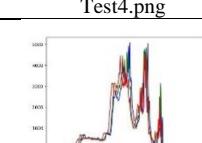
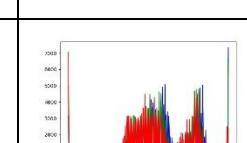
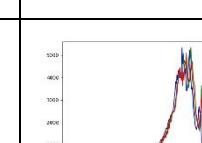
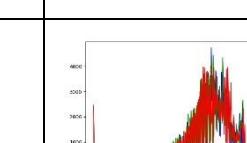
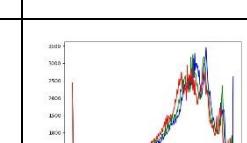
No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
10.						
11.						
12.						
13.						
14.						

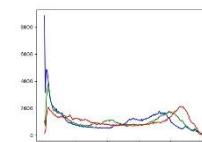
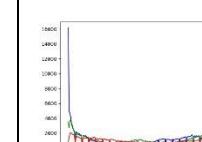
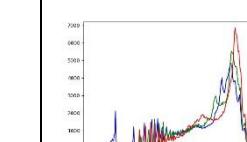
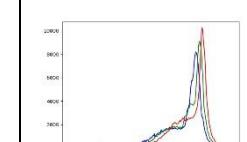
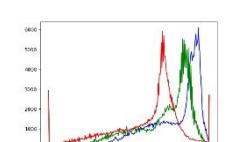
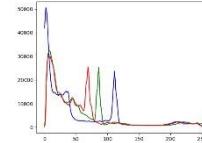
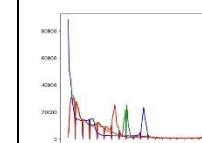
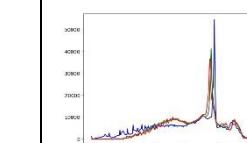
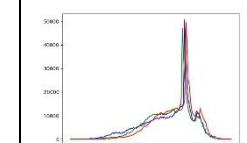
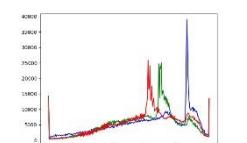
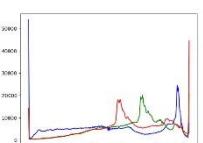
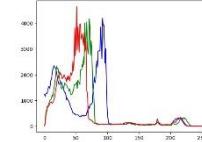
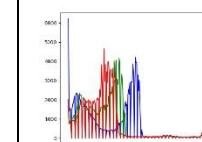
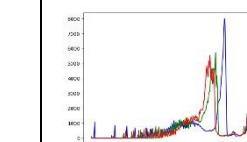
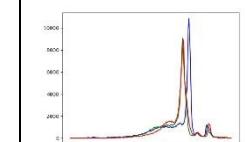
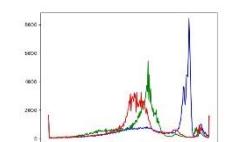
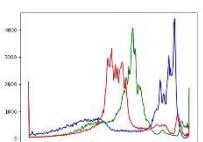
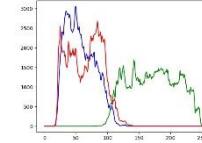
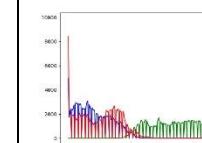
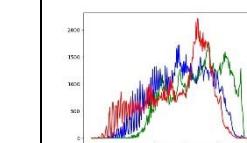
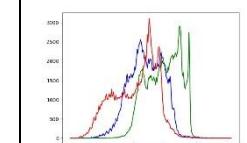
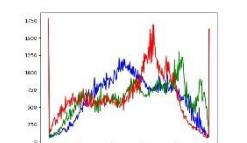
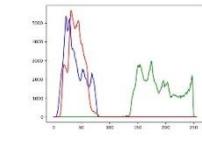
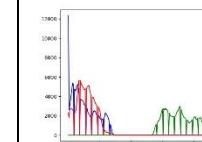
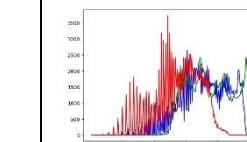
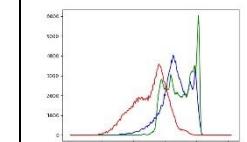
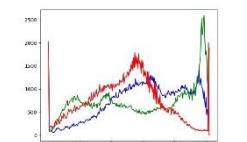
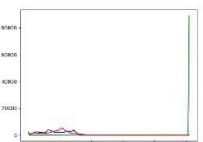
No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
15.						
16.						
17.						
18.						
19.						

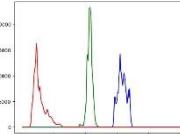
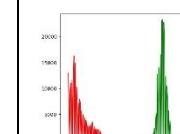
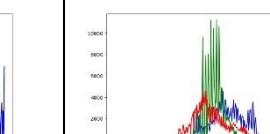
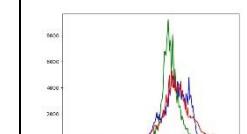
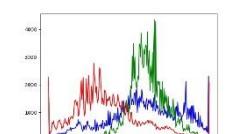
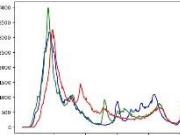
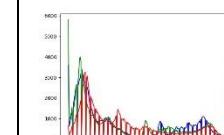
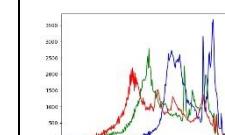
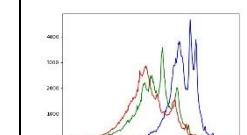
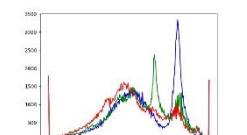
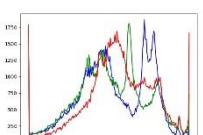
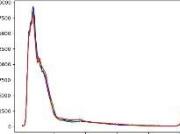
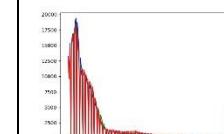
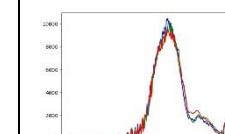
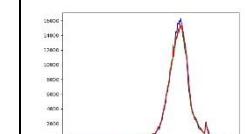
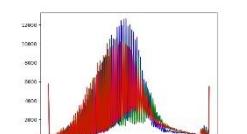
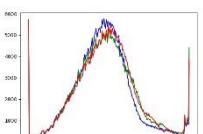
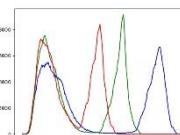
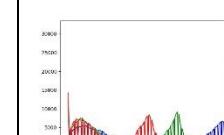
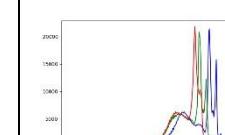
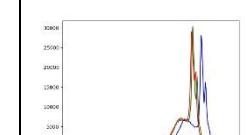
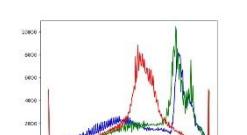
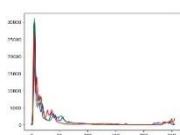
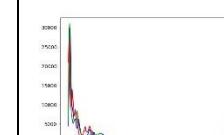
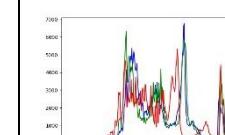
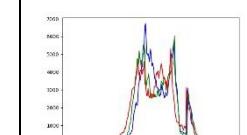
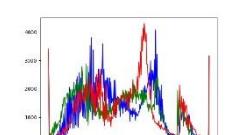
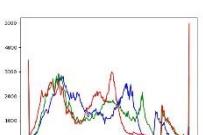
No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
20.						
21.						
22.						
23.						
24.						

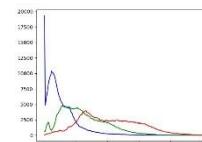
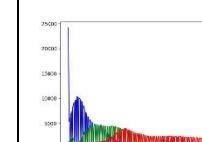
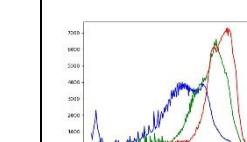
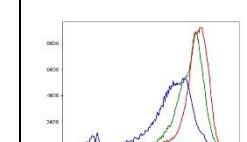
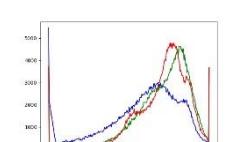
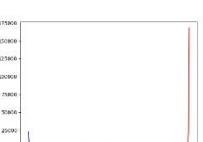
No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
25.						
26.						
27.						
28.						
29.						

No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
30						
31.						
32.						
33.						
34.						

No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
35.						
36.						
37.						
38.						
39.						

No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
40.						
41.						
42.						
43.						
44.						

No	Citra Awal	Citra Contrast Stretching	Citra Retinex (SSR)	Citra Retinex (MSR)	Citra Retinex (MSRCR)	Citra Retinex (MSRCP)
45.						
46.						
47.						
48.						
49.						

No	Citra Awal	Citra <i>Contrast Stretching</i>	Citra <i>Retinex</i> (SSR)	Citra <i>Retinex</i> (MSR)	Citra <i>Retinex</i> (MSRCR)	Citra <i>Retinex</i> (MSRCP)
50.	 Test18.png					

3. Perbandingan MSE dan PSNR Citra Hasil

No	Citra Awal	<i>Contrast Stretching</i>		<i>Retinex</i> (SSR)		<i>Retinex</i> (MSR)		<i>Retinex</i> (MSRCR)		<i>Retinex</i> (MSRCP)	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
1.	IMG-WA014.jpg	0.0158	18.004	0.0039	24.049	0.0272	15.654	0.0279	15.532	0.0350	14.546
2.	IMG-WA001.jpg	0.0002	37.864	0.1148	9.399	0.1245	9.047	0.1089	9.629	0.0873	10.585
3.	IMG-WA005.jpg	0.0003	36.330	0.2451	6.106	0.2622	5.812	0.1879	7.260	0.1693	7.711
4.	IMG-WA008.jpg	0.0034	24.561	0.1067	9.717	0.1274	8.947	0.1095	9.605	0.0917	10.372
5.	Cloud.jpg	0.0236	16.259	0.0869	10.605	0.1059	9.749	0.1012	9.947	0.0980	10.086
6.	IMG-WA009.jpg	0.0035	24.451	0.1424	8.463	0.1350	8.695	0.0666	11.761	0.0854	10.685
7.	IMG-WA029.jpg	0.0129	18.864	0.0854	10.682	0.0895	10.477	0.0208	16.816	0.0168	17.724
8.	IMG-WA025.jpg	0.0054	22.654	0.1789	7.472	0.2158	6.657	0.1882	7.252	0.1393	8.560

9.	IMG-WA030.jpg	0.0010	29.777	0.0706	11.509	0.0712	11.474	0.0643	11.915	0.0626	12.032
10.	IMG-WA028.jpg	0.0172	17.639	0.1724	7.633	0.1933	7.136	0.2283	6.413	0.2057	6.866
11.	IMG-WA012.jpg	0.0315	15.003	0.0054	22.608	0.0326	14.856	0.0362	14.404	0.0390	14.086
12.	IMG-WA023.jpg	0.0007	30.969	0.1882	7.251	0.2380	6.232	0.1752	7.563	0.1591	7.981
13.	IMG-WA024.jpg	0.0210	16.774	0.0159	17.969	0.0526	12.788	0.0514	12.886	0.0502	12.984
14.	IMG-WA026.jpg	0.0120	19.178	0.0594	12.261	0.0556	12.542	0.0313	15.032	0.0296	15.278
15.	IMG-WA006.jpg	0.0036	24.358	0.1225	9.115	0.1044	9.810	0.0721	11.418	0.0693	11.589
16.	IMG-WA015.jpg	0.0093	20.280	0.0257	15.886	0.0257	15.891	0.0240	16.187	0.0175	17.559
17.	IMG-WA017.jpg	0.0166	17.789	0.0160	17.941	0.0097	20.126	0.0230	16.380	0.0223	16.506
18.	IMG-WA018.jpg	0.0098	20.086	0.1207	9.182	0.1409	8.508	0.0469	13.288	0.0290	15.364
19.	IMG-WA019.jpg	0.0216	16.654	0.0929	10.318	0.0694	11.584	0.0580	12.363	0.0630	12.001
20.	IMG-WA020.jpg	0.0085	20.673	0.1116	9.522	0.1094	9.607	0.0827	10.823	0.0832	10.798
21.	IMG-WA033.jpg	0.0025	25.897	0.1952	7.093	0.1762	7.538	0.1410	8.505	0.1050	9.786
22.	IMG-WA021.jpg	0.0140	18.511	0.0913	10.390	0.0889	10.506	0.0740	11.307	0.0728	11.373
23.	IMG-WA031.jpg	0.0017	27.658	0.1926	7.152	0.1326	8.771	0.1421	8.471	0.1202	9.200
24.	IMG-WA027.jpg	0.0047	23.271	0.1520	8.181	0.1477	8.305	0.0552	12.576	0.0478	13.200
25.	Iris.jpg	0.0003	34.513	0.1198	9.213	0.1496	8.250	0.1412	8.501	0.1424	8.464

26.	For.jpg	0.0591	12.279	0.0192	17.160	0.0136	18.637	0.0283	15.482	0.0359	14.448
27.	19.jpg	0.0424	13.720	0.0220	16.558	0.0359	14.441	0.0221	16.541	0.0207	16.829
28.	Lena.png	0.2118	6.740	0.4134	3.835	0.3868	4.124	0.2728	5.641	0.2940	5.316
29.	25.jpg	0.0526	12.787	0.0909	10.409	0.0227	16.430	0.0811	10.908	0.0803	10.947
30.	10.jpg	0.0075	21.216	0.1787	7.477	0.2140	6.695	0.0578	12.374	0.0375	14.253
31.	22.jpg	0.0237	16.248	0.0011	29.540	0.0018	27.391	0.0125	19.003	0.0128	18.922
32.	Tes13.png	0.0373	14.282	0.0154	18.116	0.0101	19.946	0.0254	15.951	0.0216	16.642
33.	14.jpg	0.0088	20.513	0.0131	18.794	0.0116	19.355	0.0179	17.462	0.0144	18.413
34.	City.jpg	0.0174	17.588	0.0707	11.500	0.0304	15.168	0.0446	13.497	0.0433	13.629
35.	Clouds.jpg	0.0298	15.254	0.0255	15.926	0.0270	15.674	0.0399	13.985	0.0391	14.070
36.	Alone.jpg	0.0539	12.683	0.0096	20.150	0.0164	17.851	0.0264	15.770	0.0244	16.123
37.	Test1.png	0.0240	16.196	0.0238	16.226	0.0138	18.575	0.0184	17.346	0.0173	17.610
38.	Test4.png	0.0099	20.038	0.0679	11.677	0.0931	10.308	0.1863	7.297	0.1766	7.529
39.	Tree.jpg	0.0139	18.542	0.0489	13.101	0.0290	15.365	0.0396	14.017	0.0406	13.908
40.	Origin.jpg	0.0018	27.250	0.1271	8.955	0.1298	8.866	0.1059	9.750	0.0739	11.309
41.	Demo2.png	0.0006	32.007	0.1809	7.423	0.1959	7.079	0.1836	7.359	0.1788	7.475
42.	Test9.png	0.0021	26.639	0.2218	6.540	0.1986	7.020	0.1581	8.009	0.1525	8.165
43.	Test6.jpg	0.0036	24.383	0.0707	11.501	0.0337	14.714	0.0765	11.159	0.0178	17.475
44.	Test14.png	0.0018	27.396	0.1591	7.982	0.1265	8.976	0.1463	8.347	0.0178	17.477
45.	Test15.png	0.0404	13.929	0.1034	9.852	0.1199	9.209	0.0696	11.573	0.0421	13.754
46.	Test12.png	0.0040	23.963	0.0817	10.874	0.0745	11.274	0.0498	13.023	0.0408	13.887
47.	Test16.png	0.0014	28.449	0.2388	6.218	0.2861	5.433	0.124	9.040	0.1205	9.189
48.	Test17.png	0.0033	24.740	0.1837	7.357	0.1887	7.240	0.1078	9.672	0.1012	9.945
49.	Test5.png	0.0001	38.514	0.1771	7.517	0.1755	7.554	0.1165	9.335	0.0917	10.373
50.	Test18.png	0.0144	18.404	0.2035	6.912	0.2271	6.436	0.2006	6.975	0.1132	9.459

4. Perbandingan Size Citra Awal dan Citra Hasil (.JPG dan .PNG)

No	Citra Awal	Size Citra Awal	Size Contrast Stretching		Size Retinex							
			SSR		MSR		MSRCR		MSRCP			
			.JPG	.PNG	.JPG	.PNG	.JPG	.PNG	.JPG	.PNG	.JPG	.PNG
1.	IMG-WA014.jpg	64,2kb	170kb	669kb	112kb	534kb	113kb	554kb	181kb	712kb	168kb	799kb
2.	IMG-WA001.jpg	70,9kb	142kb	950kb	192kb	917kb	176kb	901kb	283kb	1,15 mb	264kb	1,17 mb
3.	IMG-WA005.jpg	42,2kb	83,9kb	525kb	126kb	637kb	121kb	688kb	213kb	860kb	192kb	954kb
4.	IMG-WA008.jpg	73,3kb	152kb	813kb	194kb	854kb	182kb	858kb	248kb	992kb	234kb	1,02 mb
5.	Cloud.jpg	15,8kb	32,4kb	127kb	27,9kb	143kb	34kb	184kb	42,7kb	205kb	39,3 kb	192kb
6.	IMG-WA009.jpg	96,1kb	216kb	1,32 mb	157kb	929kb	155kb	931kb	269kb	1,39 mb	239kb	1,37 mb
7.	IMG-WA029.jpg	16,2kb	42,7kb	309kb	42,1kb	296kb	41,5 kb	301kb	62,4kb	383kb	58,5 kb	394kb
8.	IMG-WA025.jpg	54,6kb	118kb	782kb	159kb	919kb	153kb	952kb	243kb	1,13 mb	221kb	1,27 mb
9.	IMG-WA030.jpg	84kb	183kb	1,25 mb	212kb	1,4mb	231kb	1,54 mb	343kb	1,84 mb	318kb	1,87 mb
10.	IMG-WA028.jpg	67,8kb	159kb	1,03 mb	156kb	952kb	162kb	1,02 mb	258kb	1,3 mb	242kb	1,41 mb
11.	IMG-WA012.jpg	125kb	84kb	316kb	73,2kb	312kb	70,1 kb	321kb	85,6kb	351kb	82,1 kb	348kb
12.	IMG-WA023.jpg	61kb	121kb	687kb	135kb	747kb	136kb	777kb	199kb	929kb	180kb	919kb

13.	IMG-WA024.jpg	108kb	242kb	1,37 mb	203kb	1,2mb	203kb	1,24 mb	256kb	1,43 mb	245kb	1,43 mb
14.	IMG-WA026.jpg	93kb	208kb	1,18 mb	182kb	1,01 mb	183kb	1,05 mb	348kb	1,42 mb	313kb	1,6mb
15.	IMG-WA006.jpg	55kb	123kb	775kb	90,5kb	550kb	89,7 kb	560kb	150kb	765kb	131kb	813kb
16.	IMG-WA015.jpg	149kb	322kb	1,71 mb	203kb	1,17 mb	203kb	1,17 mb	320kb	1,62 mb	315kb	1,67 mb
17.	IMG-WA017.jpg	18,5kb	71,9kb	289kb	70,3kb	352kb	80,5 kb	471kb	114kb	498kb	98kb	573kb
18.	IMG-WA018.jpg	32kb	83,5kb	418kb	77,2kb	380kb	75kb	398kb	167kb	544kb	138kb	643kb
19.	IMG-WA019.jpg	23,8kb	62,4kb	282kb	46,4kb	212kb	46kb	211kb	101kb	272kb	80,1 kb	400kb
20.	IMG-WA020.jpg	27,8kb	70,2kb	390kb	58,8kb	294kb	59,1 kb	313kb	91,6kb	394kb	80,8 kb	469kb
21.	IMG-WA033.jpg	88,1kb	192kb	1,32 mb	213kb	1,35 mb	215kb	1,41 mb	337kb	1,76 mb	339kb	1,88 mb
22.	IMG-WA021.jpg	41,7kb	94,7kb	519kb	85,1kb	397kb	81,8kb	410kb	117kb	491kb	111kb	561kb
23.	IMG-WA031.jpg	60,5kb	128kb	820kb	150kb	883kb	154kb	926kb	256kb	1,17 mb	235kb	1,24 mb
24.	IMG-WA027.jpg	153kb	305kb	1,18 mb	237kb	1,44 mb	237kb	1,46 mb	394kb	1,97 mb	373kb	2,08 mb
25.	Iris.jpg	19,4kb	25,9kb	115kb	33,5kb	132kb	32,8kb	139kb	37,4kb	150kb	38,4kb	151kb
26.	For.jpg	14,7kb	33,6kb	149kb	29,6kb	175kb	29,9kb	185kb	57,5kb	246kb	51,3kb	253kb
27.	19.jpg	19,7kb	40,7kb	193kb	34,8kb	203kb	38,9kb	229kb	54,2kb	263kb	51,6kb	271kb
28.	Lena.png	766kb	119kb	335kb	99,1kb	456kb	95,9kb	472kb	144kb	607kb	136kb	541kb
29.	25.jpg	16kb	37,9kb	161kb	36,1kb	207kb	42,8kb	262kb	60,8kb	303kb	56,6kb	285kb
30.	10.jpg	90,9kb	140kb	599kb	98,1kb	482kb	98,1kb	482kb	181kb	673kb	172kb	668kb
31.	22.jpg	50,1kb	93,3kb	408kb	73,7kb	389kb	71,5kb	379kb	100kb	465kb	97,1kb	479kb

32.	Tes13.png	311kb	76,9kb	246kb	68,4kb	277kb	71,6kb	294kb	101kb	348kb	92,7kb	369kb
33.	14.jpg	31,9kb	51,8kb	279kb	56,9kb	298kb	59,3kb	315kb	72,3kb	350kb	70kb	357kb
34.	City.jpg	41,7kb	68,8kb	300kb	57,6kb	271kb	61,7kb	287kb	80,7kb	333kb	77,3kb	335kb
35.	Clouds.jpg	99,5kb	76,7kb	277kb	70,8kb	313kb	73kb	325kb	93,2kb	362kb	88,7kb	358kb
36.	Alone.jpg	55,4kb	96,1kb	386kb	90,2kb	485kb	90,7kb	510kb	124kb	588kb	109kb	592kb
37.	Test1.png	224kb	79,3kb	301kb	60,6kb	285kb	60,7kb	292kb	79,9kb	337kb	79,3kb	347kb
38.	Test4.png	729kb	175kb	781kb	276kb	804kb	270kb	861kb	384kb	1,04 mb	356kb	953kb
39.	Tree.jpg	32,8kb	46,4kb	226kb	41,3kb	223kb	42,6kb	242kb	53,6kb	282kb	52,1kb	285kb
40.	Origin.jpg	86,6kb	138kb	575kb	126kb	507kb	123kb	501kb	156kb	569kb	156kb	581kb
41.	Demo2.png	1,92 mb	397kb	1,96 mb	478kb	2,12 mb	470kb	2,11 mb	633kb	2,49 mb	634kb	2,51 mb
42.	Test9.png	224kb	40,3kb	192kb	42,5kb	194kb	42,5kb	200kb	56,9kb	235kb	55,1kb	240kb
43.	Test6.jpg	25,9kb	31kb	172kb	35,1kb	201kb	37,1kb	219kb	53,9kb	265kb	52,3kb	221kb
44.	Test14.png	357kb	34,6kb	206kb	49,3kb	269kb	52,8kb	285kb	88,7kb	362kb	73,1kb	275kb
45.	Test15.png	275kb	35,5kb	202kb	57,5kb	236kb	58kb	256kb	100kb	337kb	83,3kb	334kb
46.	Test12.png	370kb	62,1kb	280kb	57,3kb	260kb	58,3kb	264kb	77,8kb	319kb	75,8kb	320kb
47.	Test16.png	499kb	142kb	744kb	155kb	783kb	148kb	781kb	260kb	995kb	253kb	1mb
48.	Test17.png	633kb	159kb	745kb	127kb	634kb	128kb	640kb	244kb	884kb	217kb	857kb
49.	Test5.png	275kb	58,6kb	317kb	92kb	400kb	91,1kb	420kb	142kb	512kb	134kb	519kb
50.	Test18.png	678kb	184kb	728kb	180kb	699kb	176kb	700kb	224kb	793kb	207kb	730kb

LAMPIRAN B

Listing Program

```
import sys
import platform
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtCore import (pyqtSlot, QCoreApplication, QPropertyAnimation,
QDate, QDateTime, QMetaObject, QObject, QPoint, QRect, QSize, QTime, QUrl,
Qt, QEvent, pyqtSlot)
from PyQt5.QtGui import (QImage, QPixmap, QBrush, QColor, QConicalGradient,
QCursor, QFont, QFontDatabase, QIcon, QKeySequence, QLinearGradient,
QPalette, QPainter, QPixmap, QRadialGradient)
from PyQt5.QtWidgets import *
from PyQt5.QtWidgets import QMainWindow, QApplication, QFileDialog,
 QMessageBox
from PyQt5.QtPrintSupport import QPrintDialog, QPrinter
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math
from matplotlib.figure import Figure
from matplotlib.backends.backend_qt5agg import (NavigationToolbar2QT as
NavigationToolbar)
from splash_screen import Ui_SplashScreen
from ui_main import Ui_MainWindow
from ui_functions import UIFunctions

global fname
global image

counter = 0
fname = ""

class MainWindow(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.Btn_Toggle.clicked.connect(lambda:
UIFunctions.toggleMenu(self, 250, True))
        # # PAGE 1
        self.ui.btn_page_1.clicked.connect(lambda:
self.ui.stackedWidget.setCurrentWidget(self.ui.page_1))
        # # PAGE 2
        self.ui.btn_page_2.clicked.connect(lambda:
self.ui.stackedWidget.setCurrentWidget(self.ui.page_2))
        self.ui.btn_buka_gambar.clicked.connect(self.open_img)
        self.ui.btn_proses.clicked.connect(self.proses)
        self.ui.btn_hitung.clicked.connect(self.indikator)
        # # PAGE 3
        self.ui.btn_page_3.clicked.connect(lambda:
self.ui.stackedWidget.setCurrentWidget(self.ui.page_3))
```

```

        self.image = None
        self.fname = None
        self.tmp = None

    @pyqtSlot()
    def open_img(self):
        global fname
        fname, filter = QFileDialog.getOpenFileName(self, 'Open File',
'D:\\\\', "Image Files (*.png *.jpg)")
        if fname:
            self.loadImage(fname)
        else:
            print("Invalid Image")
        self.histogram()

    def loadImage(self, fname):
        self.image = cv2.imread(fname)
        self.tmp = self.image
        self.displayImage()

    def displayImage(self, window=1):
        qformat = QImage.Format_Indexed8
        if len(self.image.shape) == 3:
            if(self.image.shape[2]) == 4:
                qformat = QImage.Format_RGBA8888
            else:
                qformat = QImage.Format_RGB888
        img = QImage(self.image, self.image.shape[1], self.image.shape[0],
self.image.strides[0], qformat)
        img = img.rgbSwapped()
        if window == 1:
            self.ui.label_gambar_awal.setPixmap(QPixmap.fromImage(img))
            self.ui.label_gambar_awal.setAlignment(QtCore.Qt.AlignHCenter |
QtCore.Qt.AlignVCenter)
        if window == 2:
            self.ui.label_contras.setPixmap(QPixmap.fromImage(img))
            self.ui.label_contras.setAlignment(QtCore.Qt.AlignHCenter |
QtCore.Qt.AlignVCenter)
        if window == 3:
            self.ui.label_ssr.setPixmap(QPixmap.fromImage(img))
            self.ui.label_ssr.setAlignment(QtCore.Qt.AlignHCenter |
QtCore.Qt.AlignVCenter)
        if window == 4:
            self.ui.label_msr.setPixmap(QPixmap.fromImage(img))
            self.ui.label_msr.setAlignment(QtCore.Qt.AlignHCenter |
QtCore.Qt.AlignVCenter)
        if window == 5:
            self.ui.label_msrr.setPixmap(QPixmap.fromImage(img))
            self.ui.label_msrr.setAlignment(QtCore.Qt.AlignHCenter |
QtCore.Qt.AlignVCenter)
        if window == 6:
            self.ui.label_msrrc.setPixmap(QPixmap.fromImage(img))
            self.ui.label_msrrc.setAlignment(QtCore.Qt.AlignHCenter |
QtCore.Qt.AlignVCenter)

#####
#~FUNCTIONMETHOD~#####
def proses(self):

```

```

self.contrasstretching()
    self.ssr()
    self.msr()
    self.msrr()
    self.msrrcp()

def contrasstretching(self):
    self.image=self.constrech()
    cv2.imwrite('out/kontras.jpg', self.image)
    cv2.imwrite('out/kontras.png', self.image)
    self.displayImage(2)
    self.histogram2()

def constrech(self):
    self.image = self.tmp
    img = self.image
    min_percent = 2
    max_percent = 98
    low, high = np.percentile(img, (min_percent, max_percent))
    res_img = (img.astype(float) - low) / (high-low)
    return np.maximum(np.minimum(res_img*255, 255),
0).astype(np.uint8)

def ssr(self):
    self.image=self.retinex_SSR()
    cv2.imwrite('out/ssr.jpg', self.image)
    cv2.imwrite('out/ssr.png', self.image)
    self.displayImage(3)
    self.histogram3()

def retinex_SSR(self):
    self.image = self.tmp
    img = self.image
    sigma = 250
    if len(img.shape)==2:
        img=img[...,None]
    ret=np.zeros(img.shape,dtype='uint8')
    for i in range(img.shape[-1]):
        channel=img[...,i].astype('double')
        S_log=np.log(channel+1)
        #kernel gaussian
        ksize=int(np.floor(sigma*6)/2)*2+1
        k_1D=np.arange(ksize)-ksize//2
        k_1D=np.exp(-k_1D**2/(2*sigma**2))
        k_1D=k_1D/np.sum(k_1D)
        qr = k_1D
        #gaussian blur
        row_filter= qr
        t=cv2.filter2D(channel,-1,row_filter[...,None])
        tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
        gaussian= tr
        L_log=np.log(gaussian+1)
        r=S_log-L_log
        R=r
        mmin=np.min(R)
        mmax=np.max(R)

```

```

    self.contrasstretching()
        self.ssr()
        self.msr()
        self.msrrc()
        self.msrrcp()

    def contrasstretching(self):
        self.image=self.constrech()
        cv2.imwrite('out/kontras.jpg', self.image)
        cv2.imwrite('out/kontras.png', self.image)
        self.displayImage(2)
        self.histogram2()

    def constrech(self):
        self.image = self.tmp
        img = self.image
        min_percent = 2
        max_percent = 98
        low, high = np.percentile(img, (min_percent, max_percent))
        res_img = (img.astype(float) - low) / (high-low)
        return np.maximum(np.minimum(res_img*255, 255), 0).astype(np.uint8)

    def ssr(self):
        self.image=self.retinex_SSR()
        cv2.imwrite('out/ssr.jpg', self.image)
        cv2.imwrite('out/ssr.png', self.image)
        self.displayImage(3)
        self.histogram3()

    def retinex_SSR(self):
        self.image = self.tmp
        img = self.image
        sigma = 250
        if len(img.shape)==2:
            img=img[...,None]
        ret=np.zeros(img.shape,dtype='uint8')
        for i in range(img.shape[-1]):
            channel=img[...,i].astype('double')
            S_log=np.log(channel+1)
            #kernel gaussian
            ksize=int(np.floor(sigma*6)/2)*2+1
            k_1D=np.arange(ksize)-ksize//2
            k_1D=np.exp(-k_1D**2/(2*sigma**2))
            k_1D=k_1D/np.sum(k_1D)
            qr = k_1D
            #gaussian blur
            row_filter= qr
            t=cv2.filter2D(channel,-1,row_filter[...,None])
            tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
            gaussian= tr
            L_log=np.log(gaussian+1)
            r=S_log-L_log
            R=r
            mmin=np.min(R)
            mmax=np.max(R)
            stretch=(R-mmin) / (mmax-mmin)*255
            ret[...,i]=stretch

```

```

    return ret.squeeze()

def msr(self):
    self.image=self.MultiScaleRetinex()
    cv2.imwrite('out/msr.jpg', self.image)
    cv2.imwrite('out/msr.png', self.image)
    self.displayImage(4)
    self.histogram4()

def MultiScaleRetinex(self):
    self.image = self.tmp
    img = self.image
    flag=True
    weights=None
    sigmas=[15,80,250]
    if weights==None:
        weights=np.ones(len(sigmas))/len(sigmas)
    elif not abs(sum(weights)-1)<0.00001:
        raise ValueError('sum of weights must be 1!')
    r=np.zeros(img.shape,dtype='double')
    img=img.astype('double')
    for i,sigma in enumerate(sigmas):
        #kernel gaussian
        ksize=int(np.floor(sigma*6)/2)*2+1
        k_1D=np.arange(ksize)-ksize//2
        k_1D=np.exp(-k_1D**2/(2*sigma**2))
        k_1D=k_1D/np.sum(k_1D)
        qr = k_1D
        #gaussian blur
        row_filter= qr
        t=cv2.filter2D(img,-1,row_filter[...,None])
        tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
        r+=(np.log(img+1)-np.log(tr+1))*weights[i]
    if flag:
        mmin=np.min(r,axis=(0,1),keepdims=True)
        mmax=np.max(r,axis=(0,1),keepdims=True)
        r=(r-mmin)/(mmax-mmin)*255
        r=r.astype('uint8')
    return r

def msr_cr(self):
    self.image=self.retinex_MSRCR()
    cv2.imwrite('out/msr_cr.jpg', self.image)
    cv2.imwrite('out/msr_cr.png', self.image)
    self.displayImage(5)
    self.histogram5()

def retinex_MSRCR(self):
    self.image = self.tmp
    img = self.image
    s1=0.01
    s2=0.01
    sigmas=[12,80,250]
    alpha=125
    img=img.astype('double')+1
    csum_log=np.log(np.sum(img, axis=2))
    self.image=self.MultiScaleRetinex()

```

```

pq = self.image
msr=pq-1
r=(np.log(alpha*img)-csum_log[...,None])*msr
for i in range(r.shape[-1]):
    sort_img=np.sort(r[...,i],None)
    N=r[...,i].size
    Vmin=sort_img[int(N*s1)]
    Vmax=sort_img[int(N*(1-s2))-1]
    r[...,i][r[...,i]<Vmin]=Vmin
    r[...,i][r[...,i]>Vmax]=Vmax
    ab= (r[...,i]-Vmin)*255/(Vmax-Vmin)
    r[...,i]=ab
return r.astype('uint8')

def msrcp(self):
    self.image=self.retinex_MSRCP()
    cv2.imwrite('out/msrcp.jpg', self.image)
    cv2.imwrite('out/msrcp.png', self.image)
    self.displayImage(6)
    self.histogram6()

def retinex_MSRCP(self):
    self.image = self.tmp
    img = self.image
    eps=np.finfo(np.double).eps
    sigmas=[12,80,250]
    s1=0.01
    s2=0.01
    Int=np.sum(img, axis=2) / 3
    Diffs=[]
    for sigma in sigmas:
        #kernel gaussian
        ksize=int(np.floor(sigma*6)/2)*2+1
        k_1D=np.arange(ksize)-ksize//2
        k_1D=np.exp(-k_1D**2/(2*sigma**2))
        k_1D=k_1D/np.sum(k_1D)
        qr = k_1D
        #gaussian blur
        row_filter= qr
        t=cv2.filter2D(Int,-1,row_filter[...,None])
        tr = cv2.filter2D(t,-1,row_filter.reshape(1,-1))
        Diffs.append(np.log(Int+1)-np.log(tr+1))
    MSR=sum(Diffs)/3
    sort_img=np.sort(MSR,None)
    N=MSR.size
    Vmin=sort_img[int(N*s1)]
    Vmax=sort_img[int(N*(1-s2))-1]
    MSR[MSR<Vmin]=Vmin
    MSR[MSR>Vmax]=Vmax
    ab= (MSR-Vmin)*255/(Vmax-Vmin)
    Int1= ab
    B=np.max(img, axis=2)
    A=np.min(np.stack((255/(B+eps),Int1/(Int+eps)),axis=2),axis=-1)
    return (A[...,:,None]*img).astype('uint8')

#####HISTOGRAMIMAGE#####
def histogram(self):

```

```

#histogram gambar input
    self.ui.label_histo_awal.canvas.axes1.clear()
    read_img = cv2.imread(fname, cv2.IMREAD_COLOR)
    color = ('b','g','r')
    for i,col in enumerate(color):
        histr = cv2.calcHist([read_img],[i],None,[256],[0,256])
        plt.plot(histr, color=col)
        plt.savefig('out/histo.jpg')
        self.ui.label_histo_awal.canvas.axes1.plot(histr,color =
col,linewidth=1.0)
        self.ui.label_histo_awal.canvas.axes1.set_ylabel('Y',
color='blue')
        self.ui.label_histo_awal.canvas.axes1.set_xlabel('X',
color='blue')
        self.ui.label_histo_awal.canvas.axes1.set_title('Histogram')
        self.ui.label_histo_awal.canvas.axes1.set_facecolor('xkcd:wheat')
        self.ui.label_histo_awal.canvas.axes1.grid()
    self.ui.label_histo_awal.canvas.draw()
    plt.close()

def histogram2(self):
    # histogram gambar enhanc
    self.ui.label_histo_contrast.canvas.axes1.clear()
    color = ('b','g','r')
    for i,col in enumerate(color):
        histr = cv2.calcHist([self.constrech()], [i], None, [256], [0,256])
        plt.plot(histr, color=col)
        plt.savefig('out/histocontrast.jpg')
        self.ui.label_histo_contrast.canvas.axes1.plot(histr,color =
col,linewidth=1.0)
        self.ui.label_histo_contrast.canvas.axes1.set_ylabel('Y',
color='blue')
        self.ui.label_histo_contrast.canvas.axes1.set_xlabel('X',
color='blue')
        self.ui.label_histo_contrast.canvas.axes1.set_title('Histogram
Contrast Stretching')

    self.ui.label_histo_contrast.canvas.axes1.set_facecolor('xkcd:wheat')
    self.ui.label_histo_contrast.canvas.axes1.grid()
    self.ui.label_histo_contrast.canvas.draw()
    plt.close()

def histogram3(self):
    # histogram gambar enhanc
    self.ui.label_histo_ssr.canvas.axes1.clear()
    color = ('b','g','r')
    for i,col in enumerate(color):
        histr = cv2.calcHist([self.retinex_SSR()], [i], None, [256], [0,256])
        plt.plot(histr, color=col)
        plt.savefig('out/histoSSR.jpg')
        self.ui.label_histo_ssr.canvas.axes1.plot(histr,color =
col,linewidth=1.0)
        self.ui.label_histo_ssr.canvas.axes1.set_ylabel('Y',
color='blue')
        self.ui.label_histo_ssr.canvas.axes1.set_xlabel('X',
color='blue')

```

```

        self.ui.label_histo_ssr.canvas.axes1.set_title('Histogram Retinex (SSR)')
        self.ui.label_histo_ssr.canvas.axes1.set_facecolor('xkcd:wheat')
        self.ui.label_histo_ssr.canvas.axes1.grid()
        self.ui.label_histo_ssr.canvas.draw()
        plt.close()

    def histogram4(self):
        # histogram gambar enhanc
        self.ui.label_histo_msr.canvas.axes1.clear()
        color = ('b','g','r')
        for i,col in enumerate(color):
            histr =
cv2.calcHist([self.MultiScaleRetinex()], [i], None, [256], [0,256])
            plt.plot(histr, color=col)
            plt.savefig('out/histoMSR.jpg')
            self.ui.label_histo_msr.canvas.axes1.plot(histr,color =
col,linewidth=1.0)
            self.ui.label_histo_msr.canvas.axes1.set_ylabel('Y',
color='blue')
            self.ui.label_histo_msr.canvas.axes1.set_xlabel('X',
color='blue')
            self.ui.label_histo_msr.canvas.axes1.set_title('Histogram Retinex
(MSR)')
            self.ui.label_histo_msr.canvas.axes1.set_facecolor('xkcd:wheat')
            self.ui.label_histo_msr.canvas.axes1.grid()
            self.ui.label_histo_msr.canvas.draw()
            plt.close()

    def histogram5(self):
        # histogram gambar enhanc
        self.ui.label_histo_msrrc.canvas.axes1.clear()
        color = ('b','g','r')
        for i,col in enumerate(color):
            histr =
cv2.calcHist([self.retinex_MSRCR()], [i], None, [256], [0,256])
            plt.plot(histr, color=col)
            plt.savefig('out/histoMSRCR.jpg')
            self.ui.label_histo_msrrc.canvas.axes1.plot(histr,color =
col,linewidth=1.0)
            self.ui.label_histo_msrrc.canvas.axes1.set_ylabel('Y',
color='blue')
            self.ui.label_histo_msrrc.canvas.axes1.set_xlabel('X',
color='blue')
            self.ui.label_histo_msrrc.canvas.axes1.set_title('Histogram
Retinex (MSRCR)')

            self.ui.label_histo_msrrc.canvas.axes1.set_facecolor('xkcd:wheat')
            self.ui.label_histo_msrrc.canvas.axes1.grid()
            self.ui.label_histo_msrrc.canvas.draw()
            plt.close()

    def histogram6(self):
        # histogram gambar enhanc
        self.ui.label_histo_msrrcp.canvas.axes1.clear()
        color = ('b','g','r')
        for i,col in enumerate(color):

```

```

        histr = cv2.calcHist([self.retinex_MSRCP()], [i], None, [256], [0,256])
            plt.plot(histr, color=col)
            plt.savefig('out/histoMSRCP.jpg')
            self.ui.label_histo_msrcp.canvas.axes1.plot(histr,color =
col,linewidth=1.0)
                self.ui.label_histo_msrcp.canvas.axes1.set_ylabel('Y',
color='blue')
                self.ui.label_histo_msrcp.canvas.axes1.set_xlabel('X',
color='blue')
                self.ui.label_histo_msrcp.canvas.axes1.set_title('Histogram
Retinex (MSRCP)')

            self.ui.label_histo_msrcp.canvas.axes1.set_facecolor('xkcd:wheat')
            self.ui.label_histo_msrcp.canvas.axes1.grid()
            self.ui.label_histo_msrcp.canvas.draw()
            plt.close()

#####
#####NILAI INDIKATOR PEMBANDING#####
def indikator(self):
    self.indikatorkontras()
    self.indikatorssr()
    self.indikatormsr()
    self.indikatormsrcr()
    self.indikatormsrcp()

def indikatorkontras(self):
    self.image = self.tmp
    img = self.image
    actual = img
    self.image = self.constrech()
    pred = self.image
    actual = actual.astype(np.float64) / 255.
    pred = pred.astype(np.float64) / 255.
    A = np.mean((actual - pred) ** 2)
    B = 10 * math.log10(1. / A)
    self.ui.label_mse_kontras.setText("MSE = "+ str(A))
    self.ui.label_psnr_kontras.setText("PSNR = "+ str(B))

def indikatorssr(self):
    self.image = self.tmp
    img = self.image
    actual = img
    self.image = self.retinex_SSR()
    pred = self.image
    actual = actual.astype(np.float64) / 255.
    pred = pred.astype(np.float64) / 255.
    A = np.mean((actual - pred) ** 2)
    B = 10 * math.log10(1. / A)
    self.ui.label_mse_ssrr.setText("MSE = "+ str(A))
    self.ui.label_psnr_ssrr.setText("PSNR = "+ str(B))

def indikatormsr(self):
    self.image = self.tmp
    img = self.image
    actual = img
    self.image = self.MultiScaleRetinex()
    pred = self.image

```

```

actual = actual.astype(np.float64) / 255.
pred = pred.astype(np.float64) / 255.
A = np.mean((actual - pred) ** 2)
B = 10 * math.log10(1. / A)
self.ui.label_mse_msр.setText("MSE = "+ str(A))
self.ui.label_psnr_msр.setText("PSNR = "+ str(B))

def indikatormsrcr(self):
    self.image = self.tmp
    img = self.image
    actual = img
    self.image = self.retinex_MSRCR()
    pred = self.image
    actual = actual.astype(np.float64) / 255.
    pred = pred.astype(np.float64) / 255.
    A = np.mean((actual - pred) ** 2)
    B = 10 * math.log10(1. / A)
    self.ui.label_mse_msрcr.setText("MSE = "+ str(A))
    self.ui.label_psnr_msрcr.setText("PSNR = "+ str(B))

def indikatormsrcp(self):
    self.image = self.tmp
    img = self.image
    actual = img
    self.image = self.retinex_MSRCP()
    pred = self.image
    actual = actual.astype(np.float64) / 255.
    pred = pred.astype(np.float64) / 255.
    A = np.mean((actual - pred) ** 2)
    B = 10 * math.log10(1. / A)
    self.ui.label_mse_msрcp.setText("MSE = "+ str(A))
    self.ui.label_psnr_msрcp.setText("PSNR = "+ str(B))
    self.show()

# SPLASH SCREEN
class SplashScreen(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_SplashScreen()
        self.ui.setupUi(self)
        self.setWindowFlag(QtCore.Qt.FramelessWindowHint)
        self.setAttribute(QtCore.Qt.WA_TranslucentBackground)
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0, 0, 0, 60))
        self.ui.dropShadowFrame.setGraphicsEffect(self.shadow)
        self.timer = QtCore.QTimer()
        self.timer.timeout.connect(self.progress)
        self.timer.start(35)
        self.ui.label_description.setText("<strong>WELCOME</strong> TO MY
APPLICATION")
        QtCore.QTimer.singleShot(1500, lambda:
self.ui.label_description.setText("<strong>LOADING</strong> CONTRAST
STRETCHING"))

```

LAMPIRAN C
DAFTAR RIWAYAT HIDUP



DATA DIRI:

Nama	:	Della Darmawan Lubis
Nim	:	0701162041
Tempat, Tanggal Lahir	:	Medan, 25 Agustus 1996
Jenis Kelamin	:	Perempuan
Alamat	:	Dusun 1, Desa Dagang Kelambir, Tanjung Morawa, Deli Serdang, Sumatera Utara.
Kel/Desa	:	Desa Dagang Kelambir
Kecamatan	:	Tanjung Morawa
Kabupaten	:	Deli Serdang
Agama	:	Islam
Status Nikah	:	Belum Menikah
Nama Orang Tua		
Ayah	:	Dedy Darmawan Lubis
Ibu	:	Wan Latifah Hanum

PENDIDIKAN FORMAL:

2002 – 2008	:	SD SWASTA MUHAMMADIYAH TANJUNG MORAWA
2008 – 2011	:	SMP NEGERI 1 TANJUNG MORAWA
2011 – 2014	:	SMA SWASTA NUR AZIZI TANJUNG MORAWA
2016 – 2021	:	UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik 2019 / 2020

Nama : Della Darmawulan Lubis	Pembimbing I : Dr. Mhd. Fugan, S.Si., M.Comp.Sc.
NIM : 0701162041	Pembimbing II : Muhammad Ikhwan, ST, M.Kom.
Prog. Studi : ILMU KOMPUTER	SK Pembimbing :
Judul Skripsi :	Perbandingan Metode Contrast Stretching dan Metode Retinex untuk Peningkatan Kecerahan Citra Digital

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	13/8/2020	Revisi Bab I	Fugan	20/4/2020	Bimbingan Bab I (Latar Belakang)	
II	27/8/2020	Revisi Bab II	Fugan	25/4/2020	ACC BAB I	
III	14/10/2020	Revisi Bab III	Fugan	21/7/2020	Bimbingan Bab 2 dan Bab 3	
IV	15/10/2020	Pengecekan proposal skripsi	Fugan	8/8/2020	Pengiriman Ulang Bab 2 dan Bab 3	
V	18/10/2020	Acc proposal skripsi	Asst. Prof. Dr. Mhd. Fugan, S.Si., M.Comp.Sc. Dekan Fakultas Sains dan Teknologi Nurcahaya Lubis Della Darmawulan Lubis	12/8/2020	ACC Bab 2 dan Bab 3 (Acc proposal skripsi)	Arie Saniar, Propos. Skripsi Mhd. Zulkarnain

VI	18 maret 2021	Bimbingan bab 4	Fq.	21/3/ 2021	Bimbingan bab 4	f
VII	18 maret 2021	Revisi bab 4	Fq.	21/3/ 2021	Revisi bab 4	f
VIII	7 maret 2021	Bimbingan bab 5	Fq.	10/3/ 2021	Bimbingan bab 5	f
IX	10 maret 2021	Revisi bab 5	Fq.	17/3/ 2021	Revisi bab 5	f
X	22 maret 2021	Acc Skripsi (Sidang)	Xq.	19/3/ 2021	Acc Skripsi	f

Medan, 18 Maret 2021