

# Deblurring Citra Dengan Metode Lucy Richardson Deconvolution

Mawaddah Harahap<sup>1</sup>, William<sup>2</sup>, Josua Parulian Sihombing<sup>3</sup>, Abdul Rahman Lubis<sup>4</sup>

Address: Universitas Prima Indonesia, Jl. Sekip Simpang Sekambing, Medan, Indonesia<sup>1,2,3,4</sup>,

E-mail: mawaddah@unprimdn.ac.id

## Abstrak

Adanya *file* gambar hasil *scan* tidak memiliki kualitas yang bagus(blur). Citra blur ini menyebabkan objek yang terdapat didalamnya menjadi tidak terlihat jelas. Tujuan dari penelitian ini adalah untuk meningkatkan kualitas citra pada dokumen dari hasil scan dengan melakukan *deblurring* dengan menggunakan algoritma *Lucy Richardson Deconvolution*. Teknik yang digunakan untuk menghilangkan kekaburan citra (*deblurring*), penapisan dengan merestorasi citra dengan menggunakan metode iteratif, yaitu algoritma Lucy-Richardson. Metode *Lucy-Richardson* dapat melakukan *deblurring* pada citra dengan nilai *threshold* 192. Maksimal *blur* yang bisa di *deblurring* adalah sampai dengan 50% *blur* pada citra menggunakan pengujian PSNR dengan nilai minimum 30dB. Penerapan metode *Lucy-Richardson* menggunakan 1 parameter (*Threshold*), sehingga lebih memudahkan untuk mendapatkan hasil citra yang paling optimal

**Keywords** *Deblurring Citra*, Lucy Richardson Deconvolution, gambar hasil scan

## PENDAHULUAN

Citra merupakan suatu gambaran benda yang mampu ditafsirkan oleh indra penglihatan. Citra yang dapat diolah oleh komputer adalah citra yang bersifat digital. Citra digital tidak dapat langsung digunakan karena kualitas citra yang kurang baik atau perlu diolah terlebih dahulu sehingga siap digunakan[1]. Pada proses *capture* (pengambilan gambar), seringkali citra yang dihasilkan mengalami penurunan mutu (degradasi) seperti, citra mengandung derau (*noise*), kabur (*blurring*) dan sebagainya. Hal tersebut dapat mengurangi tingkat kualitas pada citra sehingga menyebabkan citra sulit diinterpretasi karena informasi yang terdapat di dalam citra berkurang[3]. Citra kabur(*blur*) disebabkan karena pengambilan gambar yang tidak fokus atau objek yang bergerak. Untuk mengatasi permasalahan-permasalahan tersebut maka diperlukan pengolahan citra (*image processing*). Salah satu teknik pengolahan citra adalah *Image Restoration* atau restorasi citra[1].

Penelitian mengenai restorasi citra yang pernah dilakukan sebelumnya seperti Restorasi Citra *Optical Character Recognition* Dengan Algoritma *Recurrent Hopfield*. Dari hasil percobaan perbaikan pola karakter menggunakan algoritma *hopfield* di atas diperoleh bahwa restorasi dapat berjalan dengan optimal pada vektor berdimensi 3x3, namun ada kemungkinan hasil restorasi dapat mengalami kesalahan pada vektor berdimensi lebih besar[10]. Penelitian lainnya seperti Analisis Restorasi Citra Kabur Algoritma *Wiener* Menggunakan Indeks Kualitas Citra. Hasil penelitian menunjukkan bahwa dengan memberikan model kekaburan yang berbeda pada citra yang sama, maka tingkat kabur yang terjadi juga akan berbeda. Indeks kualitas citra hasil restorasi untuk citra yang terdegradasi *Gaussian blur* lebih tinggi dibandingkan citra hasil restorasi untuk citra yang terdegradasi *motion blur*. Indeks kualitas citra hasil restorasi untuk citra yang terdegradasi *motion blur* berkisar antara 0,84229 sampai dengan 0,87146. Indeks kualitas citra hasil restorasi untuk citra yang terdegradasi

*Gaussian blur* berkisar antara 0,86969 sampai dengan 0,90025[5].

Untuk memulihkan citra yang *blur* dilakukan dengan metode *deblurring*. *Deblurring* adalah proses pemulihan citra yang terdapat *blur* menjadi citra yang lebih baik. Salah satu metode *deblurring* yang dapat digunakan adalah Metode *Lucy-Richardson*. Metode *Lucy-Richardson* adalah sebuah metode *deblurring* yang menggunakan proses iterasi. Metode ini melakukan perkiraan iterasi secara manual kemudian dikali dengan sebuah citra koreksi. Iterasi dapat dilakukan hingga citra yang diperoleh mencapai kriteria yang lebih baik. Semakin banyak jumlah iterasi maka hasil citra yang diperoleh akan semakin membaik, namun *noise* akan bertambah seiring dengan bertambahnya jumlah iterasi.

## LANDASAN TEORI

### a. Citra

Citra merupakan representasi (gambaran), kemiripan, atau bisa dikatakan imitasi dari suatu objek. Citra memiliki 3 sifat keluaran dari suatu sistem perekaman data yang berbeda yaitu, bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan [1].

Dengan transformasi *wavelet* kita dapat merubah ukuran citra, menggosok citra, memutar citra, ataupun melakukan pencerminan terhadap citra tersebut. Citra juga merupakan proyeksi *Scene* tiga dimensi ke dalam permukaan dua dimensi. *Scene* didefinisikan sebagai kumpulan objek tiga dimensi dengan pengaturan geometris dan biasanya diatur secara fisik oleh hukum alam. Dengan kata lain citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi [1].

Resolusi citra (*Image resolution*) adalah jumlah *pixel* per inci (kepadatan *pixel* per inci) yang dinyatakan dengan *pixel x pixel*. Semakin tinggi resolusi citra, maka semakin baik kualitas citra tersebut, dalam arti bahwa dalam ukuran fisik yang sama, citra dengan resolusi tinggi akan lebih detil serta jika citra diperbesar maka detil citra masih jelas. Namun, resolusi yang tinggi akan mengakibatkan jumlah *bit* yang diperlukan untuk menyimpan atau mentransmisikannya meningkat [2].

Citra dapat diubah dari domain spatial menjadi domain yang lain, dengan tujuan untuk mempermudah

pengkodean. Proses perubahan ini dinamakan transformasi.

Kedalaman *bit* menyatakan jumlah *bit* yang diperlukan untuk merepresentasikan tiap piksel citra pada sebuah *frame*. Kedalaman bit biasanya dinyatakan dalam satuan *bit / piksel*. Semakin banyak jumlah *bit* yang digunakan untuk merepresentasikan sebuah citra, maka semakin baik kualitas citra tersebut. *Bit depth* merupakan jumlah *bit* yang digunakan untuk merepresentasikan tiap *pixel*. *Bit depth* adalah jumlah *bit* untuk tiap *pixel*. Semakin banyak jumlah *bit* yang digunakan untuk merepresentasikan sebuah *pixel*, yang berarti semakin tinggi kedalaman *pixel*-nya, maka semakin tinggi pula kualitasnya, dengan resiko jumlah *bit* yang diperlukan menjadi lebih tinggi. Dengan 1 *byte* (8 *bit*) untuk tiap *pixel*, diperoleh  $2^8$  atau 256 *level* intensitas. Dengan *level* intensitas sebanyak itu, umumnya mata manusia sudah dapat dipuaskan. Kedalaman *pixel* paling rendah terdapat pada *binary-value image* yang hanya menggunakan 1 *bit* untuk tiap *pixel*, sehingga hanya ada dua kemungkinan bagi tiap *pixel*, yaitu 0 (hitam) atau 1 (putih). *Color resolution* merupakan jumlah warna yang dapat ditampilkan pada sebuah citra. Sedangkan *color depth* adalah jumlah maksimum warna pada citra berdasarkan *bit depth* dari citra dan layar monitor komputer. Tabel 2.1 berikut menunjukkan hubungan antara *bit depth* dan *color resolution* [2].

Tabel 1. Hubungan antara *Bit Depth* dan *Color Resolution*

| Bit Depth | Color Resolution | Kalkulasi                  |
|-----------|------------------|----------------------------|
| 1 bit     | 2 warna          | $2^1$ (2)                  |
| 2 bit     | 4 warna          | $2^2$ (2x2)                |
| 3 bit     | 8 warna          | $2^3$ (2x2x2)              |
| 4 bit     | 16 warna         | $2^4$ (2x2x2x2)            |
| 5 bit     | 32 warna         | $2^5$ (2x2x2x2x2)          |
| 6 bit     | 64 warna         | $2^6$ (2x2x2x2x2x2)        |
| 7 bit     | 128 warna        | $2^7$ (2x2x2x2x2x2x2)      |
| 8 bit     | 256 warna        | $2^8$ (2x2x2 x2 x2 x2x2x2) |
| 16 bit    | 65.536 warna     | $2^{16}$                   |
| 24 bit    | 16.777.215 warna | $2^{24}$                   |

Nilai suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0-255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra *integer* [2].

1. Citra Biner

Citra biner merupakan citra digital yang hanya memiliki dua kemungkinan nilai *pixel* yaitu hitam dan putih. Citra biner juga disebut sebagai citra B & W (*Black and White*) atau monokrom. Hanya dibutuhkan 1 *bit* untuk mewakili nilai setiap *pixel* dari citra biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengembangan, morfologi, ataupun *dithering* [3].

2. Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai pada setiap pikselnya, dengan kata lain nilai bagian *red = green = blue*. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra *grayscale* memiliki 8 *bit* (256 kombinasi warna keabuan) dimana terdiri dari rentang nilai 0 sampai 1 (putih hingga mendekati kehitaman) [3].

3. Citra Warna (8 bit)

Setiap piksel dari citra warna (8 *bit*) hanya diwakili oleh 8 *bit* dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada jenis citra warna 8 *bit*. Citra warna 8 *bit* dengan menggunakan palet warna 256 dengan paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Model ini lebih sering digunakan. Bentuk kedua dinamakan 8 bit *truecolor* [4].

4. Citra Warna (16 bit)

Citra warna 16 *bit* (biasanya disebut sebagai citra *highcolor*) dengan setiap pikselnya diwakili dengan 2*byte memory* 916 *bit*). Warna 16 *bit* memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5*bit* ditambah 1*bit* ekstra. Pemilihan komponen hijau dengan deret 6*bit* dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau [4].

5. Citra Warna (24 bit)

Setiap piksel dari citra warna 24*bit* diwakili dengan 24*bit* sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat

dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja. Setiap poin informasi piksel (RGB) disimpan ke dalam 1*byte* data. 8*bit* pertama penyimpanan nilai biru, kemudian diikuti dengan nilai hijau pada *bit* kedua dan pada 8*bit* terakhir merupakan warna merah [4].

b. Konversi Ruang Warna RGB ke HSV

Model warna RGB adalah suatu cara atau teknik pemodelan warna berdasarkan konsep penggabungan cahaya primer yaitu *Red*, *Green* dan *Blue* untuk membentuk suatu warna baru. Pada model pewarnaan ini RGB adalah sebuah ruang warna yang sifatnya tergantung kepada perangkat. Perangkat yang berbeda akan mendeteksi atau mereproduksi nilai *RGB* secara berbeda. Untuk membentuk warna dengan RGB, tiga warna (satu merah, satu hijau, dan satu biru) harus ditumpangkan (misalnya dengan emisi dari layar hitam, atau dengan refleksi dari layar putih) [19].

Sementara itu, model warna HSV mendefinisikan warna dalam terminologi *Hue*, *Saturation* dan *value*. Definisi untuk *Hue* dan *Saturation* sama dengan definisi pada model HSL, sedangkan untuk *value* memiliki arti kecerahan dari warna yang ada variasi dengan warna *saturation*. Nilainya berkisar antara 0 sampai 100%. Apabila nilainya 0 maka warnanya akan menjadi hitam dan apabila dinaikan maka kecerahan akan menaik dan akan muncul variasi-variasi baru dari warna tersebut [18]

Variasi dari roda HSV digunakan untuk memilih warna yang diinginkan. *Hue* diwakili oleh lingkaran / keliling dalam roda. Sumbu *horizontal* menunjukkan *saturation* dan sumbu vertikal menunjukkan *value*. Untuk mengambil suatu warna tertentu kita perlu menentukan dahulu *hue* dan kemudian kita baru memilih nilai *saturation* dan untuk *brightness* kita bisa memilih dari nilai *value*.

c. Restorasi Citra

Restorasi citra berbeda dengan peningkatan kualitas citra (*image enhancement*) meskipun keduanya sama-sama bertujuan untuk memperbaiki kualitas citra. *Image enhancement* lebih banyak berhubungan dengan penajaman dari fitur tertentu dalam citra, sedangkan restorasi citra memanfaatkan pengetahuan tentang proses terjadinya degradasi untuk memperoleh kembali citra asal [5].

Ketika sebuah citra ditangkap oleh kamera, sering kali terdapat beberapa gangguan yang mungkin terjadi, seperti kamera tidak fokus, muncul bintik-bintik yang disebabkan oleh proses *capture* tidak sempurna, pencahayaan yang tidak merata mengakibatkan intensitas tidak seragam, kontras citra terlalu rendah sehingga objek sulit untuk dipisahkan dari latar belakangnya atau gangguan yang disebabkan oleh kotoran-kotoran yang menempel pada citra, dan lain sebagainya. Setiap gangguan pada citra dinamakan *noise*. Citra yang mengandung *noise* seperti ini memerlukan langkah-langkah perbaikan. Hal ini dilakukan untuk memfasilitasi proses analisis citra [6].

Restorasi citra (*Image Restoration*) yaitu mencari terlebih dahulu penyebab kerusakan citra setelah itu baru mengaplikasikan teknik-teknik yang ada untuk memperbaikinya. Teknik restorasi berorientasi pada pemodelan distorsi dan melakukan proses kebalikan dari distorsi dalam *re-cover* Citra aslinya. Beberapa contoh kerusakan yang bisa di restorasi seperti blur/ kabur, bintik-bintik, *dual image*, *over saturated color*, dan piksel *error* [6].

Restorasi citra berkaitan dengan penghilangan atau pengurangan degradasi pada citra yang terjadi karena proses akuisisi citra. Degradasi yang dimaksud termasuk derau (yang merupakan *error* dalam nilai piksel) atau efek optis misalnya *blur* (kabur) akibat kamera yang tidak fokus atau karena gerakan kamera [7].

Tujuan terakhir dari teknik pemulihan gambar ini untuk memperbaiki gambar itu sendiri sesuai dengan beberapa pendefinisian rasa (*sense*). Teknik restorasi ini berorientasi pada memodelkan pendistorsian dan penggunaan proses kebalikan (*inverse*) dengan tujuan untuk memperoleh gambar aslinya. Pendekatan ini umumnya berkaitan dengan memformulasikan kriteria kebaikan yang akan menghasilkan perkiraan optimal dari hasil yang diinginkan. Perbedaannya, teknik mempertinggi sebenarnya adalah prosedur *heuristic* yang di desain untuk memanipulasi sebuah gambar dengan tujuan untuk mendapatkan keuntungan dari aspek *psychophysical* dari sistem penglihatan manusia [6].

Restorasi juga menjelaskan tentang memperbaiki kualitas sebuah citra dengan proses matematis dan probabilitas. Berbeda dengan *image enhancement* yang bersifat subjektif, karena proses perbaikan citranya berdasar pada perasaan manusia. Restorasi citra digital adalah suatu teknik yang memperhatikan bagaimana

mengurangi perubahan bentuk dan penurunan kualitas citra yang diawali selama pembentukan citra tersebut [2]. Gonzales dan Woodmendefinisikan restorasi citra sebagai proses yang berusaha merekonstruksi atau mengembalikan suatu citra yang mengalami degradasi.

Jadi, restorasi merupakan teknik yang berorientasi pada pemodelan degradasi dan menerapkan proses invers dalam rangka merekonstruksi pada citra yang original [6].

Kadang-kadang restorasi citra digital sering dipertukarkan dengan pengertian peningkatan kualitas citra. Kedua teknik ini memang sering bertukar metode dalam menyelesaikan masalah, tetapi titik berat restorasi adalah pada perbaikan citra yang mengalami kerusakan, baik selama proses digitalisasi maupun cacat akibat usia, jamur, goresan, pelabelan teks pada citra yang dilakukan baik sengaja maupun tidak sengaja. Sedangkan, *image enhancement* cenderung memperhatikan perbaikan kualitas citra yang mengalami penurunan kualitas selama pembentukan citra atau justru memberi efek berlebihan pada citra yang sudah ada [6].

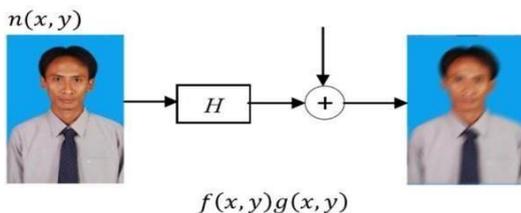
Dalam peningkatan citra (*image enhancement*), tujuan utama dari teknik restorasi adalah untuk meningkatkan citra dalam arti yang telah ditetapkan, seperti mempertajam (*sharpening*), menilangkan derau (*noise removing*), pengaturan cahaya (*brightness*) dan lainnya. Meskipun ada bidang yang tumpang tindih, peningkatan citra sebagian besar merupakan proses yang subjektif, sedangkan restorasi citra adalah untuk sebagian besar proses obyektif. Restorasi mencoba untuk memulihkan gambar yang telah terdegradasi (menurun kualitasnya) dengan menggunakan pengetahuan atau metode-metode yang telah diketahui dari fenomena degradasi seperti bintik *noise*, *blur* (kabur) dan seterusnya [6]. Dengan demikian, teknik restorasi berorientasi ke arah gambar asli. Dalam hal ini menghilangkan *noise* (derau) adalah salah satu teknik restorasi citra yang masih banyak diteliti. Pendekatan ini digunakan untuk merumuskan kriteria perbaikan yang akan menghasilkan perkiraan yang optimal untuk hasil yang diharapkan. Sebaliknya, teknik peningkatan pada dasarnya adalah prosedur *heuristic* (coba coba) yang dirancang untuk memanipulasi gambar untuk mendapatkan peningkatan dari gambar yang terdegradasi secara pengamatan langsung, sedangkan penghilangan efek gambar blur (kabur) dengan menerapkan fungsi *deblurring* dianggap sebagai teknik restorasi. Salah satu pendekatan yang banyak digunakan, termasuk juga untuk teknik restorasi citra yang terkena

noise adalah dengan menggunakan filter spasial. Filter ini bekerja dengan membentuk jendela spasial (*spatial window*) yang akan mengevaluasi setiap nilai piksel pada citra digital. Titik pusat piksel akan dikoreksi berdasarkan nilai piksel tetangganya [6].

c. Degradasi Citra

Citra yang ditangkap oleh alat-alat optik seperti mata, kamera, dan sebagainya sebenarnya merupakan citra yang sudah mengalami degradasi. Gambar 2.6 memperlihatkan model degradasi yang dalam hal ini  $f(x,y)$  merupakan citra asli dan  $g(x,y)$  adalah citra terdegradasi. Citra  $g(x,y)$  adalah hasil perkalian citra  $f(x,y)$  dengan operator distorsi H ditambah denganderau aditif  $n(x,y)$  [5].

$$g(x,y) = Hf(x,y) + n(x,y) \tag{1}$$



Gambar 2. Model Degradasi

Derau  $n(x,y)$  adalah sinyal aditif yang muncul selama proses akuisisi citra dan menyebabkan citra menjadi rusak (mengalami degradasi). Perhatikan pada model ini mengasumsikan bahwa degradasi invarian secara spasial sehingga dapat dipandang sebagai penapis linier dan sinyal aditif. Secara ringkas, persamaan (3) dapat ditulis sebagai bentuk matriks vektor [5]:

$$G = H * f + n \tag{2}$$

Jika H adalah fungsi dwimatris (2 dimensi), maka persamaan (3) dapat dipandang sebagai konvolusi fungsi ini dengan citra  $f(x,y)$  menghasilkan distorsi. Dengan demikian, persamaan (3) dapat ditulis sebagai:

$$g(x,y) = H(x,y) * f(x,y) + n(x,y) \tag{3}$$

Yang dalam hal ini simbol \* menyatakan konvolusi. Dalam domain frekuensi, model degradasi ini dapat ditulis sebagai:

$$G(u,v) = H(u,v) F(u,v) + N(u,v) \tag{4}$$

Yang dalam hal ini  $G(u,v)$ ,  $H(u,v)$ ,  $F(u,v)$ , dan  $N(u,v)$  masing-masing adalah transformasi Fourier dari  $g(x,y)$ ,  $H(x,y)$ ,  $f(x,y)$ , dan  $n(x,y)$ .

Contoh operator distorsi adalah kernel rerata yang mengakibatkan citra menjadi kabur (*blur*), yaitu :

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Kernel ini mengganti pixel yang dikonvolusikan dengan nilai rata-rata dari 8 pixel tetangganya (plus pixel itu sendiri) [5].

Permasalahan restorasi citra adalah bagaimana melakukan pendekatan hasil pengukuran  $f(x,y)$  jika diberikan  $g(x,y)$ . Kita mengasumsikan bahwa pengetahuan degradasi diketahui dalam bentuk operator H dan karakteristik derau  $n(x,y)$  diketahui (berdasarkan informasi statistik) atau dapat diestimasi. Karena permasalahan ini dipandang sebagai permasalahan pendekatan hasil pengukuran, maka kita menentukan  $f(x,y)$  sebagai estimasi terhadap  $f(x,y)$  sedemikian sehingga  $f(x,y) - f(x,y)$  menjadi minimal [5].

d. Blurring (Pengaburan)

Deblurring adalah proses menghilangkan artefak yang kabur dari sebuah citra, misalkan blur yang disebabkan oleh penyimpangan defocus atau blur. Blur tersebut biasanya dimodelkan sebagai konvolusi dari (kadang-kadang ruang angkasa atau waktu-bervariasi) fungsi penyebaran titik (*Point Spread Function*) dengan citra input yang tajam secara hipotetis, di mana kedua input citra tajam (yang akan dipulihkan) dan fungsi penyebaran titik (PSF) yang tidak diketahui. Ini adalah contoh dari masalah invers. Dalam hampir semua kasus, ada informasi yang hilang atau terkorupsi dalam citra kabur untuk menentukan secara unik citra asli yang masuk akal, inilah hal yang sulit dilakukan. Hal ini umumnya diatasi dengan penggunaan istilah regularisasi untuk mencoba menghilangkan solusi yang tidak masuk akal [17].

Efek *blur* adalah distorsi gambar yang umum terjadi di bidang fotografi. Gambar yang blur (kabur) bisa muncul karena berbagai alasan, seperti lensa kamera yang tidak fokus, intensitas cahaya yang sangat ekstrim, ketidaksempurnaan fisik lensa yang mengakibatkan deviasi optis, dan gerakan relatif objek terhadap lensa kamera. Gerakan relatif objek terhadap lensa, misalnya, diketahui menyebabkan pergeseran posisi rincian objek pada gambar sehingga tampilan tekstur dan tepian dari objek pada gambar menjadi tidak jelas. Efek motion blur bisa juga terjadi secara tidak sengaja sebagai akibat proses

pengambilan gambar digital yang tidak sempurna (misalnya karena adanya pergerakansekecil apapun [8].

Sementara itu efek blur dapat disebabkan adanya gerakan yang disengaja yang ditambahkan pada gambar untuk menimbulkan sensasi gerak cepat dari objek gambar. Fotografer sering menggunakan efek ini untuk menghasilkan gambar yang dramatis sehingga gambar yang dihasilkan akan dianggap memiliki daya tarik lebih [8].

Citra yang diperoleh melalui sensor modern sering terganggu dengan oleh berbagai sumber *noise*. Dengan *noise* kita lihat variasi stokastik sebagai lawan distorsi deterministik seperti bayangan atau kurangnya fokus. Secara khusus kita akan menganggap penggunaan perangkat modern, charge-coupled digital (CCD) kamera dimana foton menghasilkan elektron yang biasa disebut sebagai foto electron[8].

Perbaikan kualitas citra diperlukan karena seringkali citra yang dijadikan objek mempunyai kualitas yang buruk, misalnya citra mengalami derau (*noise*), citra terlalu gelap/terang, citra kurang tajam, kabur, dan sebagainya. Restorasi citra mengambil peranyang sangat penting, terlebih pada pada era citra digital seperti saat ini. Alat-alat optik digital seperti kamera memiliki keterbatasan dalam menangkap citra, sehingga menyebabkan citra yang dihasilkan kabur atau mengalami *noise*. Adapun *noise* ini bisa disebabkan oleh keterbatasan perangkat maupun manusia. Citra yang mengandung *noise* seringkali membatasi informasi berharga yang dibutuhkan ketika analisa citra dilakukan. Itulah sebabnya *noise* tersebut harus dihilangkan agar memberikan hasil yang optimal [5].

Degradasi yang dibahas pada penelitian ini adalah pengaburan(blurring). Citra yang kabur dapat disebabkan oleh berbagai sebab, misalnya pergerakan selama pengambilan gambar oleh alatoptik seperti kamera, penggunaan alat optik yang tidak fokus, penggunaan lensa dengan sudut yang lebar, gangguan atmosfer, pencahayaan yang singkat sehingga mengurangi jumlah foton yang ditangkap oleh alat optik, dan juga sebab lainnya [5].

#### e. Algoritma Lucy-Richardson Deconvolution

Orang mungkinberanggapanbahwa bila transformasi dari citra asli menjadi citra terdegradasi diketahui (yaitu PSF), maka dengan melakukan transformasi kebalikan (invers)

akan mendapatkan kembali citra terestorasi yang sempurna. Tetapi hal ini tidak benar karena pengaruh distorsi pada citra akan memberi tekanan secara kuat pada transformasi invers. Karena alasan inilah maka algoritma restorasi iteratif dikembangkan. Pada setiap iterasi, algoritma ini cenderung memperbaiki distorsi yang disebabkan oleh PSF. Iterasi dihentikan jika kompromi terbaik antara kualitas citra dengan derau telah dicapai [3].

Untuk membuat citra blur biasanya dimodelkandalam konvolusi antara citra asli dengan point spread function dimana spektrum titik atau objek akan menyebar sehingga objek terlihat memudar. Blur alami biasanya terjadi akibat adanya bayangan contoh kabut, gerakan objek saat pengambilan gambar dengan kamera, ataupun citra yang menyebar ataupun optik yang tidak fokus Citra terblur dapat ditulis seperti pada persamaan(5) [5].

Pada restorasi citra, PSF merupakan faktor penting karena PSF menggambarkandistorsi. Pada persamaan (5), citra terdegradasi dapat kita tulis sebagai [5]:

Citra terdegradasi = citra asli \* PSF + derauaditif  
Sehingga berdasarkan model degradasi pada persamaan (5), pekerjaan utama pada deblurring adalahdekonvolusi citra kabur (blur) dengan PSF. Dekonvolusi adalah proses yangmembalikkan efek konvolusi. Dengan kata lain, kualitas citra hasil deblurring terutama ditentukan oleh pengetahuanPSF [5].

Algoritma Lucy-Richardson (L-R), yang dikenal juga dengan dekonvolusi Lucy-Richardson, dikembangkan secara independen oleh Richardson (1972) dan Lucy (1974). Algoritma ini efektif jika kita mengetahui PSF tetapi hanya mengetahui sedikit mengenai derau aditif pada citra. Algoritma ini pada mulanya digunakan untuk merestorasi citra astronomi, sebelum akhirnya digunakan juga secara luas untuk merestorasi sembarang citra yang mengalami kekaburan. Algoritma ini memaksimalkan kemungkinan (*maximum likelihood*) bahwa sebuah citra bila dikonvolusi dengan PSF hasilnya adalah instansiasi dari citra kabur, dengan mengasumsikan derau tersebut dengan distribusi *Poisson*. Distribusi *Poisson* adalah [3]:

$$p(x) = \frac{e^{-a} a^x}{x!} \tag{6}$$

dengan x adalah peubah acak dan a adalah konstanta. Esensi dari iterasi adalah sebagai berikut: estimasi ke-(n+1) dari citra restorasi adalah estimasi ke-n citra

restorasi dikali dengan citra koreksi. Persamaan iterasinya adalah [3]:

$$\hat{f}_{n+1} = \hat{f}_n \left( \frac{g}{\hat{f}_n * PSF} \right) * reflect(PSF) \tag{7}$$

Keterangan:

- $f^{\wedge} \Rightarrow f^{\wedge}_{(x,y)}$  = estimasi citra restorasi
- $g \Rightarrow g_{(x,y)}$  = nilai citra yang terdistorsi
- $*reflect(PSF)$  = kebalikan nilai  $PSF_{(x,y)} = PSF_{(x,y)}$
- $f^{\wedge}n \Rightarrow f^{\wedge}_0 = f^{\wedge}_{(x,y)}$  = nilai iterasi awal
- $f_0$  = nilai awal iterasi =  $g * PSF$

yang dalam hal ini, operator  $*$  menyatakan konvolusi,  $f = f(x, y)$  menyatakan estimasi citra restorasi,  $g = g(x, y)$  menyatakan citra masukan (yang mengalami degradasi),  $reflect(PSF)$  menyatakan pencerminan PSF, yaitu  $reflect((PSF(x, y)) = PSF(-x, -y)$ , dan

$\left( \frac{g}{\hat{f}_n * PSF} \right) * reflect(PSF)$  menyatakan citra koreksi [3].

Nilai awal iterasi adalah  $f_0 = g * PSF$ . Kekonvergenan algoritma Lucy-Richardson berarti citra koreksi mendekati satu (*unity*) ketika iterasi bertambah. Jika algoritma konvergen (yang telah dibuktikan oleh Shepp dan Vardi) maka iterasinya itu konvergen ke maksimum kemungkinan statistik Poisson di dalam data [3].

Restorasi citra merupakan lahan penelitian yang terus berkembang. Algoritma Lucy-Richardson (LR) digunakan untuk restorasi dengan cara iterasi. Persamaan LR dinyatakan sebagai berikut.

$$f_{k+1}(x, y) = f_k(x, y) + w(x, y) [g(x, y) - f_k(x, y) ** h(x, y)] \tag{15}$$

Dimana:

- $*$  adalah operasi konvolusi
- $f$  adalah estimasi dari citra tidak terdegradasi

*Lucy-Richardson Deconvolution* adalah algoritma yang berdasarkan pada proses iterasi. Performansi algoritma ini untuk mendeteksi keberadaan *noise* jauh lebih baik daripada algoritma *deconvolution* lainnya. Algoritma *Lucy-Richardson Deconvolution* populer di dalam bidang astronomi dan citra medis. Algoritma ini diturunkan dari teorema Bayes pada awal tahun 1970 oleh Lucy dan Richardson. Pada awal tahun 1980, algoritma ini diperkenalkan kembali oleh Shepp dan Vardi sebagai algoritma untuk menyelesaikan permasalahan citra *positron emission tomography*, dimana statistik poisoning sangat dominan. Metode ini menggunakan sebuah solusi

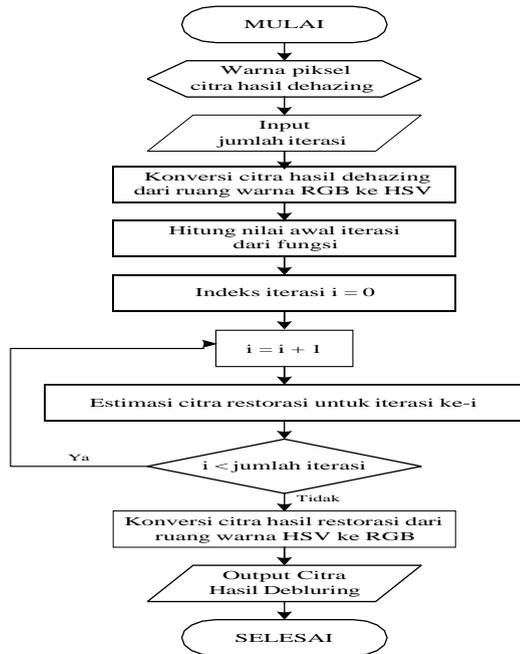
kemiripan maksimum, yang juga ditemukan penerapannya pada algoritma maksimisasi ekspektasi dari Dempster, et. al. Alasan popularitas dari algoritma *Lucy-Richardson Deconvolution* adalah implementasi dari kemiripan maksimum dan kemampuannya untuk menghasilkan citra rekonstruksi dengan kualitas yang bagus pada kondisi dimana terdapat *noise* dengan *level* tinggi [3].

## METODOLOGI

Langkah kerja dari proses *deblurring* dengan menggunakan algoritma *Lucy-Richardson Deconvolution* ini dapat dirincikan sebagai berikut:

1. *Input* citra yang akan diperbaiki.
2. Tentukan jumlah iterasi (perulangan).
3. Konversikan citra RGB yang dimasukkan ke bentuk ruang warna HSV.
4. Hitung nilai awal iterasi dari fungsi, yaitu  $F_0 = g * PSF$ .
5. Set  $i = 0$ .
6.  $i = i + 1$ .
7. Estimasi citra restorasi untuk iterasi ke- $i$ , dengan menggunakan rumusan berikut:
 
$$F_1 = F_0 * \frac{g}{F_0 * PSF} * PSF'$$
8. Ulangi langkah (6) dan (7) hingga  $i =$  jumlah iterasi.
9. Konversi citra hasil restorasi dari ruang warna HSV ke RGB.
10. Output citra hasil.

Proses kerja dari metode *Lucy-Richardson Deconvolution* dalam melakukan proses *deblurring* dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 1.



Gambar 3. Flowchart dari Proses Deblurring

Langkah kerja dari proses *deblurring* dengan menggunakan metode *Lucy-Richardson Deconvolution* dapat dirincikan sebagai berikut:

1. Warna piksel citra input, seperti terlihat pada tabel 1.

Tabel 1. Warna RGB dari Piksel Citra Input

|     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 156 | 186 | 256 | 276 | 235 | 220 | 189 | 158 | 128 |
| 156 | 176 | 266 | 276 | 286 | 237 | 223 | 194 | 109 |
| 25  | 67  | 199 | 224 | 224 | 193 | 182 | 139 | 69  |
| 226 | 246 | 246 | 266 | 266 | 227 | 212 | 181 | 135 |
| 196 | 246 | 246 | 266 | 276 | 230 | 208 | 180 | 109 |
| 100 | 158 | 191 | 216 | 216 | 188 | 166 | 144 | 58  |
| 206 | 226 | 236 | 246 | 236 | 220 | 189 | 174 | 120 |
| 176 | 206 | 236 | 246 | 246 | 223 | 201 | 166 | 95  |
| 42  | 100 | 166 | 199 | 191 | 182 | 166 | 128 | 53  |
| 206 | 216 | 216 | 236 | 226 | 204 | 189 | 166 | 89  |
| 176 | 206 | 226 | 236 | 226 | 208 | 194 | 166 | 74  |
| 50  | 108 | 158 | 183 | 183 | 171 | 161 | 118 | 37  |
| 176 | 206 | 216 | 206 | 196 | 181 | 166 | 151 | 58  |
| 146 | 196 | 216 | 216 | 206 | 187 | 166 | 145 | 53  |
| 34  | 108 | 166 | 166 | 158 | 155 | 134 | 96  | 15  |
| 146 | 206 | 196 | 186 | 176 | 166 | 157 | 140 | 96  |
| 122 | 166 | 201 | 193 | 175 | 166 | 153 | 133 | 91  |
| 38  | 94  | 149 | 155 | 149 | 150 | 133 | 100 | 67  |
| 156 | 176 | 186 | 166 | 156 | 157 | 144 | 131 | 74  |
| 113 | 148 | 166 | 175 | 148 | 149 | 133 | 120 | 61  |
| 44  | 77  | 122 | 144 | 127 | 126 | 107 | 90  | 34  |
| 136 | 166 | 156 | 146 | 126 | 140 | 118 | 109 | 30  |
| 95  | 139 | 148 | 148 | 131 | 137 | 112 | 99  | 28  |
| 33  | 94  | 127 | 127 | 105 | 113 | 87  | 74  | 8   |
| 106 | 146 | 156 | 126 | 106 | 131 | 109 | 96  | 47  |
| 77  | 122 | 131 | 122 | 104 | 128 | 103 | 78  | 49  |
| 16  | 71  | 99  | 105 | 83  | 113 | 84  | 61  | 28  |
| 106 | 126 | 126 | 106 | 106 | 135 | 109 | 91  | 65  |
| 69  | 95  | 122 | 113 | 104 | 128 | 103 | 78  | 53  |
| 16  | 55  | 83  | 99  | 99  | 117 | 87  | 54  | 31  |

2. Input jumlah iterasi. Misalkan jumlah iterasi = 1.
3. Konversi citra input dari ruang warna RGB ke HSV.

Perhitungan untuk *hue* adalah sebagai berikut:  
 $H(0,0)$

$$= 60^\circ * \frac{(G - B)}{(MAX - MIN)} + 0^\circ$$

$$= 60^\circ * \frac{(156 - 25)}{(156 - 25)} + 0^\circ$$

$$= 60^\circ * \frac{131}{131} + 0^\circ$$

$$= 60^\circ$$

Perhitungan untuk *saturation* adalah sebagai berikut:  
 $S(0,0)$

$$= 1 - \frac{MIN}{MAX}$$

$$= 1 - \frac{25}{156}$$

$$= 0.84$$

Perhitungan untuk *value* adalah sebagai berikut:

$$V(0,0) = MAX = 156.$$

Proses konversi diatas akan dilakukan untuk semua piksel lainnya. Hasil konversi dari ruang warna RGB ke HSV dapat dirincikan sebagai berikut:

Tabel 2. Nilai H

|       |        |       |        |        |        |        |         |        |        |
|-------|--------|-------|--------|--------|--------|--------|---------|--------|--------|
| 60    | 54.96  | 68.96 | 60     | 69.68  | 62.73  | 64.39  | 65.45   | 61.03  | 40.68  |
| 45.71 | 60     | 60    | 60     | 70     | 64.29  | 54.78  | 58.38   | 48.36  | 39.74  |
| 49.02 | 50.48  | 60    | 60     | 70.91  | 64.39  | 80.57  | 49.57   | 48.57  | 37.61  |
| 39.79 | 36.52  | 95.66 | 23.21  | 20.31  | 98.24  | 104.3  | 25.71   | 35.11  | 38.28  |
| 38.18 | 25.63  | 13.89 | 108.89 | 108.93 | 111.66 | 11.57  | 19.47   | 25.18  | 39.31  |
| 41.65 | 31.08  | -1290 | 180    | -32.5  | -35.56 | -48    | 1980    | 91.43  | 53.33  |
| 73.93 | 236.67 | -528  | 152.2  | -21    | -44.52 | -70.91 | -138.46 | 106.67 | 101.25 |
| 39.57 | 40.91  | -126  | 142.35 | 141    | -34.29 | -31.25 | -78.95  | 130.91 | -52.17 |
| 65.36 | 80.53  | -960  | -17.89 | -16.36 | -22.5  | -21.92 | -46.36  | 87.69  | 25     |
| 44.17 | 133.33 | -58.5 | 127.92 | -5.77  | -33    | -38.4  | -288    | 50     | 26.4   |

Tabel 3. Nilai S

|       |       |        |        |        |        |        |        |       |        |
|-------|-------|--------|--------|--------|--------|--------|--------|-------|--------|
| 0.84  | 0.64  | 0.252  | 0.188  | 0.217  | 0.186  | 0.184  | 0.284  | 0.365 | 0.461  |
| 0.558 | 0.358 | 0.224  | 0.188  | 0.217  | 0.183  | 0.217  | 0.204  | 0.424 | 0.57   |
| 0.796 | 0.558 | 0.297  | 0.191  | 0.224  | 0.184  | 0.174  | 0.264  | 0.441 | 0.558  |
| 0.922 | 0.745 | 0.633  | 0.581  | 0.562  | 0.438  | 0.552  | 0.675  | 0.696 | 0.652  |
| 1     | 1     | 1      | 1      | 1      | 1      | 1      | 1      | 1     | 1      |
| 0.829 | 0.675 | 0.01   | -0.161 | -0.273 | -0.163 | -0.159 | 0.007  | 0.172 | 0.281  |
| 0.359 | 0.102 | -0.027 | -0.234 | -0.385 | -0.197 | -0.153 | -0.099 | 0.165 | 0.216  |
| 0.691 | 0.398 | -0.064 | -0.345 | -0.458 | -0.3   | -0.407 | -0.174 | 0.121 | -0.767 |
| 0.528 | 0.26  | -0.013 | -0.452 | -0.726 | -0.305 | -0.477 | -0.229 | 0.159 | 0.245  |
| 0.679 | 0.143 | -0.317 | -0.469 | -0.491 | -0.148 | -0.229 | -0.055 | 0.385 | 0.769  |
| 0.84  | 0.64  | 0.252  | 0.188  | 0.217  | 0.186  | 0.184  | 0.284  | 0.365 | 0.461  |
| 0.558 | 0.358 | 0.224  | 0.188  | 0.217  | 0.183  | 0.217  | 0.204  | 0.424 | 0.57   |
| 0.796 | 0.558 | 0.297  | 0.191  | 0.224  | 0.184  | 0.174  | 0.264  | 0.441 | 0.558  |
| 0.922 | 0.745 | 0.633  | 0.581  | 0.562  | 0.438  | 0.552  | 0.675  | 0.696 | 0.652  |
| 1     | 1     | 1      | 1      | 1      | 1      | 1      | 1      | 1     | 1      |
| 0.829 | 0.675 | 0.01   | -0.161 | -0.273 | -0.163 | -0.159 | 0.007  | 0.172 | 0.281  |
| 0.359 | 0.102 | -0.027 | -0.234 | -0.385 | -0.197 | -0.153 | -0.099 | 0.165 | 0.216  |
| 0.691 | 0.398 | -0.064 | -0.345 | -0.458 | -0.3   | -0.407 | -0.174 | 0.121 | -0.767 |
| 0.528 | 0.26  | -0.013 | -0.452 | -0.726 | -0.305 | -0.477 | -0.229 | 0.159 | 0.245  |
| 0.679 | 0.143 | -0.317 | -0.469 | -0.491 | -0.148 | -0.229 | -0.055 | 0.385 | 0.769  |

Tabel 4. Nilai V

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 156 | 186 | 266 | 276 | 286 | 237 | 223 | 194 | 159 | 128 |
| 226 | 246 | 246 | 266 | 276 | 230 | 212 | 181 | 158 | 135 |
| 206 | 226 | 236 | 246 | 246 | 223 | 201 | 174 | 143 | 120 |
| 206 | 216 | 226 | 236 | 226 | 208 | 194 | 166 | 135 | 89  |
| 176 | 206 | 216 | 216 | 206 | 187 | 166 | 151 | 112 | 58  |
| 146 | 206 | 201 | 193 | 176 | 166 | 157 | 140 | 122 | 96  |
| 156 | 176 | 186 | 175 | 156 | 157 | 144 | 131 | 109 | 74  |
| 136 | 166 | 156 | 148 | 131 | 140 | 118 | 109 | 91  | 30  |
| 106 | 146 | 156 | 126 | 106 | 131 | 109 | 96  | 82  | 49  |
| 106 | 126 | 126 | 113 | 106 | 135 | 109 | 91  | 78  | 65  |

4. Hitung nilai awal iterasi dari fungsi.

Terdapat beberapa jenis PSF (*Point Spread Function*) yang dapat muncul dalam citra. Dalam penelitian ini, digunakan PSF Gaussian blur yang memiliki filter sebagai berikut:

PSF:

$$\frac{1}{256} \begin{bmatrix} 01 & 04 & 06 & 04 & 01 \\ 04 & 16 & 24 & 16 & 04 \\ 06 & 24 & 36 & 24 & 06 \\ 04 & 16 & 24 & 16 & 04 \\ 01 & 04 & 06 & 04 & 01 \end{bmatrix}$$

g adalah citra input dimana proses hanyaakan dilakukan terhadap nilai V (*value*) yang diperoleh pada tabel 4 diatas.

Rumus yang digunakan untuk menghitung nilai awal iterasi dari fungsi adalah  $F_0 = g * PSF$ . Proses perhitungan konvolusi  $g * PSF$  dapat dijabarkan sebagai berikut:

Piksel (0, 0):

Tabel 5. Ilustrasi Blok 5 x 5 dari Piksel (0, 0)

|   |   |     |     |     |
|---|---|-----|-----|-----|
| 0 | 0 | 0   | 0   | 0   |
| 0 | 0 | 0   | 0   | 0   |
| 0 | 0 | 156 | 186 | 266 |
| 0 | 0 | 226 | 246 | 246 |
| 0 | 0 | 206 | 226 | 236 |

$$= 0 * 1/256 + 0 * 4/256 + 0 * 6/256 + \dots + 236 * 1/256$$

$$= 95.29688 = 95$$

Proses perhitungan konvolusi  $g * PSF$  diatas akan diulangi untuk setiap piksel lainnya pada citra digital. Hasil  $F_0$  yang diperoleh dapat dirincikan sebagai berikut:

Tabel 6. Hasil  $F_0 = g * PSF$

|     |     |     |     |     |     |     |     |     |    |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 95  | 142 | 169 | 181 | 179 | 165 | 147 | 128 | 104 | 69 |
| 123 | 181 | 203 | 213 | 209 | 196 | 174 | 152 | 123 | 79 |
| 127 | 187 | 212 | 217 | 212 | 200 | 178 | 155 | 123 | 76 |
| 123 | 180 | 201 | 204 | 197 | 184 | 166 | 143 | 112 | 67 |
| 112 | 169 | 188 | 187 | 179 | 167 | 151 | 132 | 102 | 62 |
| 105 | 156 | 173 | 170 | 161 | 151 | 139 | 122 | 97  | 61 |
| 98  | 142 | 157 | 152 | 144 | 135 | 125 | 110 | 85  | 50 |
| 86  | 128 | 139 | 133 | 126 | 119 | 111 | 97  | 74  | 43 |
| 72  | 107 | 117 | 111 | 107 | 102 | 96  | 83  | 63  | 41 |
| 53  | 78  | 85  | 84  | 83  | 79  | 75  | 64  | 50  | 32 |

5. Set indeks iterasi  $i = 0$ .

6.  $i = i + 1 = 0 + 1 = 1$ .

7. Estimasi citra restorasi untuk iterasi ke-1.

$$F_1 = F_0 * \frac{g}{F_0 * PSF} * PSF'$$

Perhitungandari  $F_0 * PSF =$  Piksel (0, 0):

Tabel 7. Ilustrasi Blok 5 x 5 dari Piksel (0, 0) pada  $F_0$

|   |   |     |     |     |
|---|---|-----|-----|-----|
| 0 | 0 | 0   | 0   | 0   |
| 0 | 0 | 0   | 0   | 0   |
| 0 | 0 | 95  | 142 | 169 |
| 0 | 0 | 123 | 181 | 203 |
| 0 | 0 | 127 | 187 | 212 |

$$= 0 * 1/256 + 0 * 4/256 + 0 * 6/256 + \dots + 212 * 1/256$$

$$= 63$$

Proses perhitungankonvolusi  $F_0 * PSF$  diatas akan diulangi untuk setiap piksel lainnya pada citradigital. Hasil  $F_0 * PSF$  yang diperoleh dapat dirincikan sebagai berikut:

Tabel 8. Hasil  $F_0 * PSF$

|   |    |    |    |    |    |    |    |   |   |
|---|----|----|----|----|----|----|----|---|---|
| 6 | 10 | 12 | 13 | 12 | 12 | 10 |    | 7 | 4 |
| 3 | 0  | 2  | 0  | 9  | 1  | 9  | 94 | 2 | 4 |
| 8 | 13 | 16 | 16 | 16 | 15 | 14 | 12 | 9 | 5 |
| 4 | 3  | 0  | 9  | 7  | 8  | 2  | 1  | 3 | 6 |
| 9 | 14 | 17 | 18 | 18 | 16 | 15 | 13 | 9 | 5 |
| 2 | 5  | 4  | 3  | 0  | 9  | 2  | 0  | 9 | 9 |
| 9 | 14 | 16 | 17 | 17 | 16 | 14 | 12 | 9 | 5 |
| 0 | 1  | 9  | 6  | 2  | 1  | 5  | 4  | 3 | 5 |
| 8 | 13 | 15 | 16 | 15 | 14 | 13 | 11 | 8 | 5 |
| 5 | 3  | 8  | 3  | 8  | 8  | 4  | 4  | 7 | 2 |
| 7 | 12 | 14 | 14 | 14 | 13 | 12 | 10 | 7 | 4 |
| 9 | 2  | 5  | 9  | 3  | 4  | 2  | 4  | 9 | 7 |

|   |    |    |    |    |    |    |    |   |   |
|---|----|----|----|----|----|----|----|---|---|
| 7 | 11 | 13 | 13 | 12 | 12 | 10 |    | 7 | 4 |
| 1 | 1  | 0  | 3  | 8  | 0  | 9  | 94 | 1 | 2 |
| 6 |    | 11 | 11 | 11 | 10 |    |    | 6 | 3 |
| 2 | 97 | 4  | 6  | 1  | 5  | 96 | 82 | 2 | 7 |
| 4 |    |    |    |    |    |    |    | 5 | 3 |
| 9 | 77 | 90 | 92 | 89 | 84 | 77 | 66 | 0 | 0 |
| 3 |    |    |    |    |    |    |    | 3 | 2 |
| 3 | 52 | 62 | 65 | 63 | 60 | 55 | 47 | 6 | 1 |

Perhitungandari  $\frac{g}{F_0 + PSF} =$

Piksel (0, 0):

Tabel 9. Hasil  $\frac{g}{F_0 + PSF}$

|          |          |          |          |          |          |          |          |          |    |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----|
| 2.47     |          |          |          |          |          |          |          |          |    |
| 2.690476 | 1.84     |          |          |          |          |          |          |          |    |
| 2.23913  | 1.558621 | 1.35     |          |          |          |          |          |          |    |
| 2.288889 | 1.531915 | 1.337278 | 1.3      |          |          |          |          |          |    |
| 2.070588 | 1.548872 | 1.367089 | 1.325153 | 1.30     |          |          |          |          |    |
| 1.848101 | 1.688525 | 1.386207 | 1.295302 | 1.230769 | 1.2      |          |          |          |    |
| 2.197183 | 1.585586 | 1.430769 | 1.315789 | 1.21875  | 1.308333 | 1.3      |          |          |    |
| 2.193548 | 1.711134 | 1.368421 | 1.275862 | 1.18018  | 1.333333 | 1.229167 | 1.       |          |    |
| 2.163265 | 1.896104 | 1.733333 | 1.369565 | 1.191011 | 1.559524 | 1.415584 | 1.454545 |          |    |
| 3.212121 | 2.423077 | 2.032258 | 1.738462 | 1.68254  | 2.25     | 1.981818 | 1.93617  | 2.166667 | 3. |

Perhitungan  $F_1 = F_0 * \frac{g}{F_0 + PSF} * PSF'$ :

Tabel 10. Hasil  $F_1 = F_0 * \frac{g}{F_0 + PSF} * PSF'$

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 235 | 9   | 113 | 129 | 142 | 68  | 46  | 9   | 230 | 201 |
| 76  | 80  | 57  | 80  | 90  | 30  | 5   | 227 | 209 | 190 |
| 29  | 36  | 33  | 37  | 35  | 9   | 235 | 207 | 178 | 155 |
| 27  | 21  | 14  | 19  | 4   | 238 | 222 | 191 | 163 | 108 |
| 232 | 7   | 2   | 248 | 233 | 211 | 187 | 175 | 131 | 69  |
| 194 | 8   | 240 | 220 | 198 | 187 | 179 | 164 | 150 | 125 |
| 215 | 225 | 225 | 200 | 176 | 177 | 165 | 153 | 130 | 88  |
| 189 | 219 | 190 | 170 | 149 | 159 | 136 | 129 | 109 | 35  |
| 156 | 203 | 203 | 152 | 127 | 159 | 136 | 121 | 103 | 67  |
| 170 | 189 | 173 | 146 | 140 | 178 | 149 | 124 | 108 | 99  |

- Karena jumlah iterasi hanya 1 kali saja, maka proses selesai.
- Konversi citra hasil restorasi dari ruang warna HSV ke RGB.

Proses perhitungan HSV ke RGB dengan perincian sebagai berikut:

**Piksel (0, 0):**

$$C = H * S = 60 * 0.84 = 50.4$$

$$X = C * (1 - |(H/60) \bmod 2 - 1|)$$

$$X = 50.4 * (1 - |(60/60) \bmod 2 - 1|)$$

$$X = 50.4$$

$$m = V - C = 235 - 50.4 = 184.6$$

Karena  $0^\circ \leq H \leq 60^\circ$ , maka:

$$R = C + m = 50.4 + 184.6 = 235$$

$$G = X + m = 50.4 + 184.6 = 235$$

$$B = m = 184.6 = 185$$

Hasil konversi dari HSV ke RGB:

Tabel 11. Hasil Konversi Nilai RGB

|      |      |      |      |      |      |        |      |      |      |
|------|------|------|------|------|------|--------|------|------|------|
| 235, | 9,   | 110, | 129, | 140, | 67,  | 45,    | 7,   | 230, | 201, |
| 235, | 6,   | 113, | 129, | 142, | 68,  | 46,    | 9,   | 230, | 195, |
| 185  | 230  | 96   | 118  | 127  | 56   | 34     | 246  | 208  | 182  |
| 76,  | 80,  | 57,  | 80,  | 87,  | 29,  | 5,     | 227, | 209, | 190, |
| 70,  | 80,  | 57,  | 80,  | 90,  | 30,  | 4,     | 227, | 205, | 182, |
| 50   | 59   | 44   | 69   | 75   | 18   | 249    | 215  | 188  | 167  |
| 29,  | 36,  | 33,  | 37,  | 32,  | 8,   | 230,   | 207, | 178, | 155, |
| 22,  | 32,  | 33,  | 37,  | 35,  | 9,   | 235,   | 205, | 174, | 147, |
| 246  | 8    | 15   | 26   | 19   | 253  | 221    | 194  | 157  | 134  |
| 27,  | 21,  | 234, | 19,  | 4,   | 211, | 179,   | 191, | 163, | 108, |
| 15,  | 10,  | 14,  | 11,  | 252, | 238, | 222,   | 181, | 153, | 99,  |
| 246  | 250  | 209  | 6    | 249  | 195  | 164    | 174  | 139  | 83   |
| 232, | 7,   | 2,   | 159, | 144, | 115, | 187,   | 175, | 131, | 69,  |
| 218, | 248, | 247, | 248, | 233, | 211, | 178,   | 162, | 116, | 55,  |
| 194  | 237  | 244  | 139  | 124  | 99   | 175    | 156  | 106  | 30   |
| 194, | 8,   | 240, | 249, | 198, | 187, | 179,   | 164, | 142, | 125, |
| 183, | 254, | 246, | 220, | 194, | 185, | 177,   | 150, | 150, | 123, |
| 159  | 243  | 253  | 220  | 189  | 181  | 171    | 14   | 134  | 110  |
| 209, | 201, | 225, | 236, | 176, | 177, | 165,   | 153, | 116, | 73,  |
| 215, | 202, | 222, | 200, | 171, | 175, | 163, 1 | 144, | 130, | 88,  |
| 188  | 225  | 211  | 217  | 168  | 168  | 54     | 139  | 112  | 66   |
| 189, | 219, | 190, | 219, | 214, | 159, | 136,   | 129, | 93,  | 35,  |
| 180, | 214, | 183, | 170, | 149, | 155, | 130,   | 125, | 109, | 30,  |
| 162  | 203  | 182  | 201  | 191  | 149  | 123    | 115  | 96   | 251  |
| 153, | 196, | 203, | 152, | 127, | 159, | 136,   | 121, | 97,  | 67,  |
| 156, | 203, | 191, | 146, | 118, | 155, | 129,   | 119, | 103, | 63,  |
| 121  | 182  | 191  | 144  | 115  | 152  | 126    | 110  | 89   | 61   |
| 170, | 170, | 173, | 206, | 140, | 178, | 149,   | 124, | 108, | 99,  |
| 162, | 189, | 173, | 146, | 137, | 176, | 146,   | 121, | 105, | 88,  |
| 140  | 174  | 154  | 198  | 137  | 173  | 140    | 108  | 89   | 79   |

## HASIL

Pengujian Threshold akan menguji nilai *threshold* yang paling ideal yang akan diuji dari nilai Threshold 10-255 dengan kenaikan 10 per pengujian dan kemudian akan mengambil nilai PSNR yang paling bagus dan akan diuji kembali dengan kenaikan 1 per pengujian untuk mencari *threshold* yang paling ideal.

Hasil pengujian dapat dirincikan sebagai berikut:

Tabel 12. Hasil pengujian *threshold* dengan *blurring* 5%.

| Thresho Id | Nilai MSE | Nilai PSNR |
|------------|-----------|------------|
| 10         | 79.99     | 38.62      |
| 20         | 78.45     | 38.66      |
| 30         | 79.04     | 38.64      |
| 40         | 78.76     | 38.65      |
| 50         | 78.83     | 38.65      |
| 60         | 78.69     | 38.65      |
| 70         | 78.7      | 38.65      |
| 80         | 78.57     | 38.65      |
| 90         | 79.19     | 38.64      |
| 100        | 78.15     | 38.67      |
| 110        | 79.35     | 38.63      |
| 120        | 78.71     | 38.65      |
| 130        | 78.18     | 38.67      |
| 140        | 77.77     | 38.68      |
| 150        | 66.61     | 39.01      |
| 160        | 66.91     | 39         |
| 170        | 61.22     | 39.2       |
| 180        | 56.43     | 39.34      |
| 190        | 55.72     | 39.4       |
| 200        | 56.72     | 39.36      |
| 210        | 60.61     | 39.22      |
| 220        | 57.33     | 39.34      |
| 230        | 61.69     | 39.18      |
| 240        | 57.15     | 39.35      |
| 250        | 60.83     | 39.21      |
| 255        | 57.11     | 39.35      |

Hasil dari pengujian menunjukkan bahwa *threshold* yang memiliki PSNR paling tinggi adalah 190-200.

Tabel 13. Hasil pengujian *threshold* 190-200 dengan *blurring* 5%.

| Threshold | Nilai MSE | Nilai PSNR |
|-----------|-----------|------------|
| 190       | 55.72     | 39.4       |
| 191       | 61.08     | 39.2       |
| 192       | 60.78     | 39.21      |
| 193       | 56.05     | 39.39      |
| 194       | 60.5      | 39.22      |
| 195       | 61.35     | 39.19      |
| 196       | 60.28     | 39.23      |
| 197       | 60.51     | 39.22      |
| 198       | 60.55     | 39.22      |
| 199       | 61.81     | 39.18      |
| 200       | 56.72     | 39.36      |

Hasil dari pengujian dengan *threshold* 190-200 menunjukkan bahwa *threshold* yang memiliki PSNR paling tinggi adalah 192.

Tabel 14. Hasil pengujian *threshold* dengan *blurring* 10%.

| Nilai Threshold | Nilai MSE | Nilai PSNR |
|-----------------|-----------|------------|
| 10              | 77.71     | 38.68      |
| 20              | 77.77     | 38.68      |
| 30              | 78.06     | 38.66      |
| 40              | 77.95     | 38.67      |
| 50              | 78.18     | 38.66      |
| 60              | 77.77     | 38.68      |
| 70              | 77.42     | 38.69      |
| 80              | 77.95     | 38.67      |
| 90              | 77.64     | 38.68      |
| 100             | 77.88     | 38.67      |
| 110             | 78.21     | 38.66      |
| 120             | 77.73     | 38.68      |
| 130             | 77.87     | 38.67      |
| 140             | 77.94     | 38.67      |
| 150             | 69.20     | 38.93      |
| 160             | 65.39     | 39.05      |
| 170             | 52.99     | 39.51      |
| 180             | 52.92     | 39.50      |
| 190             | 52.75     | 39.52      |
| 200             | 53.07     | 39.51      |
| 210             | 53.14     | 39.50      |
| 220             | 52.88     | 39.51      |
| 230             | 53.18     | 39.50      |
| 240             | 53.16     | 39.50      |
| 250             | 53.06     | 39.51      |
| 255             | 52.93     | 39.51      |

Hasil dari pengujian menunjukkan bahwa *threshold* yang memiliki PSNR paling tinggi adalah 190-200.

Tabel 15. Hasil pengujian *threshold* 190-200 dengan *blurring* 10%.

| Threshold | Nilai MSE | Nilai PSNR |
|-----------|-----------|------------|
| 190       | 52.75     | 39.52      |
| 191       | 69.25     | 38.96      |
| 192       | 63.86     | 39.10      |
| 193       | 64.27     | 39.09      |
| 194       | 69.43     | 38.92      |
| 195       | 71.12     | 38.87      |
| 196       | 68.92     | 38.94      |
| 197       | 65.51     | 39.05      |
| 198       | 65.18     | 39.06      |
| 199       | 65.03     | 39.07      |
| 200       | 65.15     | 39.05      |

Hasil dari pengujian dengan *threshold* 190-200 menunjukkan bahwa *threshold* yang memiliki PSNR paling tinggi adalah 192.

Tabel 16. Hasil pengujian *threshold* dengan *blurring* 15%.

| Threshold | Nilai MSE | Nilai PSNR |
|-----------|-----------|------------|
| 10        | 86        | 38.46      |
| 20        | 83.86     | 38.51      |
| 30        | 85.67     | 38.47      |
| 40        | 85.17     | 38.48      |
| 50        | 87.65     | 38.42      |
| 60        | 84.08     | 38.51      |
| 70        | 83.5      | 38.52      |
| 80        | 85.15     | 38.48      |
| 90        | 86.87     | 38.44      |
| 100       | 84.18     | 38.5       |
| 110       | 83.13     | 38.53      |
| 120       | 86        | 38.46      |
| 130       | 85.1      | 38.48      |
| 140       | 85.24     | 38.48      |
| 150       | 76.73     | 38.71      |
| 160       | 76.57     | 38.71      |
| 170       | 77.42     | 38.69      |
| 180       | 78.49     | 38.66      |
| 190       | 74.57     | 38.77      |
| 200       | 75.61     | 38.74      |
| 210       | 75.41     | 38.74      |
| 220       | 76.21     | 38.72      |
| 230       | 77.11     | 38.7       |
| 240       | 76.1      | 38.72      |
| 250       | 77.38     | 38.69      |
| 255       | 229.36    | 36.33      |

Hasil dari pengujian menunjukkan bahwa *threshold* yang memiliki PSNR paling tinggi adalah 190-200.

Tabel 17. Hasil pengujian *threshold* 190-200 dengan *blurring* 15%

| Threshold | Nilai MSE | Nilai PSNR |
|-----------|-----------|------------|
| 190       | 74.57     | 38.77      |
| 191       | 76.44     | 38.71      |
| 192       | 74.43     | 38.77      |
| 193       | 75.58     | 38.74      |
| 194       | 77.4      | 38.69      |
| 195       | 77.14     | 38.69      |
| 196       | 78.03     | 38.67      |
| 197       | 76.19     | 38.72      |
| 198       | 78.73     | 38.65      |
| 199       | 78.96     | 38.64      |
| 200       | 75.61     | 38.74      |

Tabel diatas menunjukkan *threshold* 192 adalah *threshold* paling ideal.

Tabel 18. Hasil pengujian *threshold* dengan *blurring* 20%.

| Nilai Threshold | Nilai MSE | Nilai PSNR |
|-----------------|-----------|------------|
| 10              | 78.06     | 38.66      |
| 20              | 77.91     | 38.67      |
| 30              | 77.83     | 38.67      |
| 40              | 77.7      | 38.67      |
| 50              | 77.98     | 38.67      |
| 60              | 77.78     | 38.67      |
| 70              | 77.36     | 38.68      |
| 80              | 77.93     | 38.67      |
| 90              | 77.84     | 38.67      |
| 100             | 77.91     | 38.6       |
| 110             | 77.89     | 38.67      |
| 120             | 77.61     | 38.68      |
| 130             | 77.80     | 38.67      |
| 140             | 77.86     | 38.67      |
| 150             | 69.45     | 38.92      |
| 160             | 65.6      | 39.04      |
| 170             | 59.61     | 39.25      |
| 180             | 54.19     | 39.46      |
| 190             | 53.34     | 39.49      |
| 200             | 53.38     | 39.49      |
| 210             | 53.33     | 39.48      |
| 220             | 53.5      | 39.48      |
| 230             | 53.83     | 39.47      |
| 240             | 53.57     | 39.48      |
| 250             | 53.82     | 39.47      |
| 255             | 54.18     | 39.46      |

Hasil dari pengujian menunjukkan bahwa *threshold* yang memiliki PSNR paling tinggi adalah 190-200.

Tabel 19. Hasil pengujian *threshold* 190-200 dengan *blurring* 20%.

| Threshold | Nilai MSE | Nilai PSNR |
|-----------|-----------|------------|
| 190       | 53.34     | 39.49      |
| 191       | 88.42     | 38.4       |
| 192       | 68.1      | 38.90      |
| 193       | 86.78     | 38.44      |
| 194       | 89.09     | 38.38      |
| 195       | 71.17     | 38.87      |
| 196       | 88.91     | 38.39      |
| 197       | 87.89     | 38.41      |
| 198       | 87.22     | 38.43      |
| 199       | 86.42     | 38.45      |
| 200       | 53.38     | 39.49      |

Hasil pengujian dengan menggunakan beberapa gambar input berbeda dapat dilihat pada perincian tabel berikut:

| Citra Input | Citra Hasil | Nilai MSE | Nilai PSNR |
|-------------|-------------|-----------|------------|
|             |             | 66.61     | 39.01      |
|             |             | 66.91     | 39         |
|             |             | 61.22     | 39.2       |
|             |             | 61.35     | 39.19      |
|             |             | 60.28     | 39.23      |
|             |             | 60.51     | 39.22      |
|             |             | 78.18     | 38.66      |
|             |             | 77.77     | 38.68      |
|             |             | 68.1      | 38.90      |
|             |             | 86.78     | 38.44      |

## PEMBAHASAN

Berdasarkan proses pengujian yang telah dilakukan, maka dapat diperoleh beberapa informasi berikut:

1. *Threshold* yang terlalu tinggi akan menyebabkan *deblurring* masih bersisa di citra sedangkan jika terlalu kecil maka akan melakukan iterasi terus menerus walaupun sudah tidak diperlukan. Hal ini berarti bahwa *threshold* tidak boleh terlalu tinggi untuk menghasilkan citra *deblurring* yang bagus.
2. Proses *deblurring* dengan nilai PSNR lebih dari 30dB dari hasil pengujian diperoleh untuk *threshold* 192. Hal ini berarti bahwa citra *deblurring* dengan kualitas terbagus akan diperoleh untuk nilai *threshold* sebesar 192.
3. Proses *deblurring* dengan nilai PSNR paling bagus adalah di *deblurring* 5%. Hal ini berarti citra *deblurring* akan memiliki kualitas terbagus jika

metode diterapkan pada citra dengan tingkat *blur* sebesar 5 % saja.

4. Nilai parameter yang mempengaruhi citra hanya *threshold* sehingga akan lebih sedikit mempengaruhi kualitas citra hasil.
5. Semakin besar persentase *deblurring* pada citra, maka jumlah iterasi akan semakin banyak dan PSNR yang dihasilkan juga akan semakin rendah. Hal ini berarti bahwa citra dengan tingkat *blur* yang tinggi akan sulit dilakukan proses *deblurring* dan citra hasil *deblurring* akan memiliki kualitas yang kurang bagus.

## KESIMPULAN

Berdasarkan hasil penerapan metode *Lucy-Richardson* dalam melakukan proses *deblurring* dan meningkatkan kualitas citra dokumen hasil *scan* adalah sebagai berikut:

1. Metode *Lucy-Richardson* dapat melakukan *deblurring* pada citra dengan nilai *threshold* 192.
2. Maksimal *blur* yang bisa di *deblurring* adalah sampai dengan 50% *blur* pada citra menggunakan pengujian PSNR dengan nilai minimum 30dB.
3. Penerapan metode *Lucy-Richardson* menggunakan 1 parameter (*Threshold*), sehingga lebih memudahkan untuk mendapatkan hasil citra yang paling optimal.

## DAFTAR PUSTAKA

- [1] Danoedoro, P., (2016), Pengolahancitra digital: Teori dan Aplikasinya dalam Bidang Penginderaan Jauh, Andi Offset, Yogyakarta.
- [2] Putra, D., (2014), Pengolahan Citra Digital. Yogyakarta : Penerbit Andi.
- [3] Munir, R., (2014), Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung : Informatika.
- [4] Sinaga, A. S. RM, (2017), Implementasi Teknik *Threshoding* pada Segmentasi Citra Digital, Jurnal Mantik Penusa, Volume 1 No 2, p-ISSN 2088-3943, e-ISSN 2580-9741.
- [5] Nur Afyat, (2017), Analisis Restorasi Citra Kabur Algoritma Wiener Menggunakan Indeks Kualitas Citra, Nusantara Journal of Computers and Its Applications Volume 2 No 1.
- [6] Gonzalez, R.C. dan Woods, R.E., (2016), *Digital Image Preprocessing*. New Jersey: Upper Saddle River.
- [7] Susilawati, I., (2019), Teknik Pengolahan Citra, Universitas Mercu Buana Yogyakarta.
- [8] Fitria, N. H., (2017), Penerapan Fade untuk Mengukur Citra Pasca Proses Haze Removal (Studi Kasus Citra Berkabut pada Kawah Gunung Kelud), Universitas Islam Negeri Maulana Malik Ibrahim.
- [9] [slideplayer.info/slide/14270454](http://slideplayer.info/slide/14270454), 2019.
- [10] Yudhistiro, K., (2018), Restorasi Citra Optical Character Recognition Dengan Algoritma Recurrent Hopfield, SMATIKA Jurnal Volume 08 Nomor 01, April Tahun 2018 ISSN: 2087-0256, e-ISSN: 2580-6939.
- [11] Apriliyani, D. E., (2017), Analisis Kinerja Metode Lucy-Richardson dan Blind Deconvolution, Jurnal Teknologi Rekayasa Volume 22 No.1, April 2017.
- [12] Doni, (2019), Implementasi Perbaikan Kualitas Citra dengan Menggunakan Metode Lucy-Richardson, Majalah Ilmiah INTI, Volume 14, Nomor 3, September 2019 ISSN 2339-210X.
- [13] Ganesh, R. S., Babu, J., Prasad, N. B., Jagadeeshwar reddy, M., Devi. T, Sumanth, A., (2019), Blurring an Image using Python, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-9S4, July 2019.
- [14] Jintao L., Cuifang K., & Liu X., (2019), Structured illumination microscopy with modified joint Richardson-Lucy deconvolution, Proc. SPIE 11186, Advanced Optical Imaging Technologies II, 111860T (19 November 2019); <https://doi.org/10.1117/12.2537687>.
- [15] Li, T. J., Yuan Y., Shuai, Y. & Tan H. P., (2019), Joint method for reconstructing three-dimensional temperature of flame using Lucy-Richardson and nearest neighbor filtering using light-field imaging, Sci. China Technol. Sci. 62, 1232–1243 (2019). <https://doi.org/10.1007/s11431-018-9416-x>.
- [16] Mamta, R. G., Dutta, M., (2019), New Cepstrum Based Image Restoration Algorithm for Grayscale Images, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-11, September 2019.

- [17] Iriananda, S.W., (2016), Pemulihan Citra Blur (Deblur) Menggunakan Metode Lucy-Richardson dan Filter Wiener, Universitas Brawijaya.
- [18] DH Sulystiawati, (2018), Analisa Citra Parasit Malaria Dalam Ruang Warna Hue Saturation Value [HSV], JHP17 : Jurnal Hasil Penelitian 3 (01)
- [19] Vina Chovan Epifania, Eko Sedyono, 2011. Pencarian File Gambar Berdasarkan Dominasi Warna. Jurnal Buana Informatika, Vol 2 No 1, Januari 2011

#### AUTHOR(S) BIOGRAPHY



Mawaddah Harahap Dosen di Universitas Prima Indonesia Fakultas Teknologi dan Ilmu Komputer Program Studi Teknik Informatika, mulai bekerja sebagai dosen tahun 2009, Artificial Intelligent, Deep Learning, Computer Vision. bidang ilmu

Teknik Informatika, S2 Teknik Informatika Universitas Sumatera Utara, S1 Sistem Informasi Universitas Harapan.