# PEMBUATAN APIKASI RESTORASI CITRA DIGITAL MENGGUNAKAN MATLAB

**DR.Wendi Zarman,M.Si[1], John Adler,M.Si[2], Nuraeni Ratnawati[3]**
[1,2,3] Jurusan Teknik Komputer Unikom, Bandung
[3]nuraeniratnawati@gmail.com

***ABSTRAK***

Masalah yang umum terjadi pada citra digital adanya derau, muncul bintik-bintik, pengambilan gambar yang tidak fokus atau objek yang bergerak, sehingga menyebabkan hasil citra yang kabur. Pada citra yang kabur, akan direstorasi melalui proses *deblurring* menggunakan metode algoritma Lucy-Richardson. Algoritma Lucy-Richardson bergantung dari jumlah iterasi yang digunakan. Proses akhir dari restorasi citra akan dibersihkan dari derau melalui proses *denoising* menggunakan metode *Median Filter.* Proses restorasi citra dibuat menggunakan perangkat lunak Matlab. Algoritma Lucy-Richardson mampu memperbaiki kualitas citra yang terdegradasi *motion blur*, dengan iterasi ke-5 hingga ke-10. Untuk restorasi citra digital yang optimal dibutuhkan penelitian lebih lanjut mengenai hubungan antara jumlah iterasi dengan proses perbaikan citra.

Kata kunci : *restorasi citra, deblurring, Lucy-Richardson, denoising, Median Filter*

## PENDAHULUAN

Citra dapat berupa analog maupun digital, agar memudahkan pengiriman ataupun penerimaan informasi, citra yang digunakan biasanya bersifat digital. Namun tampaknya tidak semua citra digital dapat langsung digunakan oleh pengguna karena hasil pada citra yang kurang baik kualitasnya dan perlu diolah terlebih dahulu sehingga siap untuk digunakan. Masalah yang umum terjadi adalah karena adanya derau (*noise)* yang mengakibatkan citra terlihat kotor, muncul bintik-bintik, dan hilangnya informasi. Masalah lainnya terjadi karena pengambilan gambar yang tidak fokus atau objek yang bergerak, sehingga menyebabkan hasil citra yang kabur (*blur*).

Untuk menjawab permasalahan-permasalahan pada citra digital diperlukan pemrosesan citra (*Image Processing*). Diantara bentuk pemrosesan citra digital adalah *Image Restoration*, yang bertujuan memperbaiki kualitas citra dengan menghilangkan cacat pada citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Masukan dan keluarannya berupa citra. Bentuk restorasi citra ada dua yaitu, proses *deblurring* atau penghilangan kesamaran citra yang kabur dan proses *denoising* atau penghilangan derau.

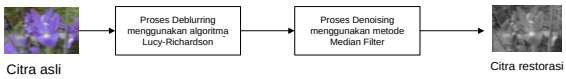
## PERANCANGAN

Pertama akan ditampilkan diagram aktivitas untuk menggambarkan alur dari langkah-langkah yang akan dipakai dalam menjalankan program. Kemudian, rancangan untuk tampilan layar program aplikasi dan rancangan untuk modul-modul dalam program.

Aplikasi yang akan dibuat secara garis besar memiliki alur penggunaan. Pengguna program akan langsung memasuki layar utama saat membuka program. Dilayar utama, pengguna akan memasukkan file citra digital yang mengalami *blur* atau kabur dan berderau dengan menekan tombol *Open File*. Citra yang dipilih kemudian akan ditampilkan

### 1. Diagram Blok Aplikasi

Aplikasi yang akan dibuat secara garis besar akan memiliki alur sebagai berikut.

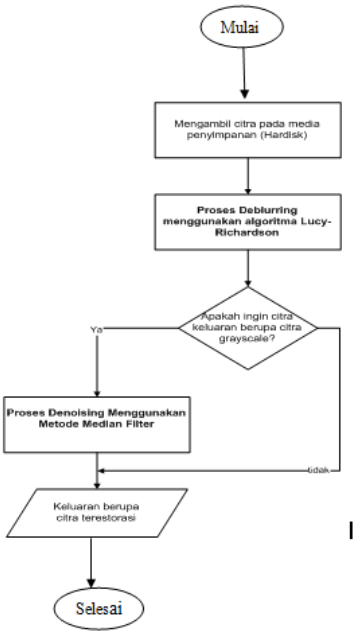



*Gambar 1 Diagram Blok*

Berikut ini masing-masing peran atau fungsi yang dicitrakan dalam perancang diatas:

1. Citra asli, merupakan masukan citra asli yang bersifat kabur (*blur*) dan berderau.
2. Proses *deblurring*, merupakan proses memperbaiki citra yang kabur dengan menggunakan metode *Blind Deconvolution Lucy-Richardson(L-R).*
3. Proses *denoising*, merupakan proses penghilangan derau pada citra hasil proses *deblurring*.. Metode yang digunakan adalah metode *Median Filter.*
4. Citra restorasi, merupakan keluaran atau *ouput* dari proses restorasi citra.

**2. *Flowchart***

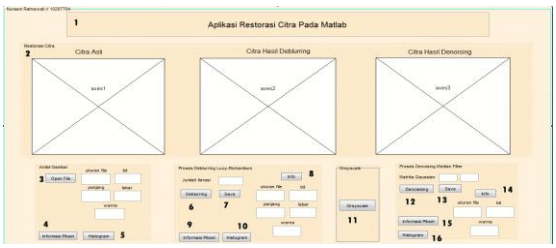



*Gambar 2 Flowchart*

Keterangan *flowchart:*

1. Mulai
2. Mengambil citra pada media penyimpanan, yaitu pengambilan citra digital dari penyimpanan *hardisk* yang akan diproses dan berupa citra terdegradasi.
3. Proses *deblurring*, yaitu proses menghilangkan kesamaran menggunakan metode *blind* dekonvolusi L-R yang bergantung pada nilai iterasi. Nilai iterasi yang digunakan merupakan inputan. Untuk melanjutkan pada proses selanjutnya, maka disimpan terlebih dahulu.
4. *Decision*, merupakan pilihan apakah akan melalui proses restorasi lain yaitu proses *denoising* atau tidak.
5. Proses *denoising*, merupakan proses menghilangkan derau dengan menggunakan metoda *median filter* dan citra masukan diambil dari hasil proses *deblurring*.
6. Citra hasil restorasi, keluaran citra dengan hasil akhir.
7. Selesai

**3. Antarmuka**




*Gambar 3. Antarmuka program*

Keterangan antarmuka program*:*

1. Panel judul aplikasi.
2. Panel untuk menampilkan citra.
3. Tombol *Open File*, untuk mengambil citra asli dari tempat penyimpanan.
4. Tombol informasi piksel, untuk menampilkan informasi piksel citra asli.
5. Tombol histogram untuk menampilkan histogram citra asli
6. Tombol *deblurring* untuk memproses dekonvolusi dari citra asli yang sebelumnya telah ada masukkan nilai iterasi.
7. Tombol *save* untuk menyimpan citra hasil *deblurring.*
8. Tombol info untuk menampilkan informasi citra berupa ukuran file, tinggi dan lebar citra, serta nilai bit.
9. Tombol informasi piksel, untuk menampilkan informasi piksel pada citra yang telah terrestorasi *deblurring.*
10. Menampilkan histogram.
11. Tombol *grayscale*, untuk mengubah citra hasil *deblurring* menjadi citra *grayscale* sebelum kemudian diproses *denoising.*
12. *Tombol denoising*, untuk memproses pengurangan derau.
13. Tombol *save* untuk menyimpan citra hasil *denoising*
14. Tombol info untuk menampilkan informasi citra berupa ukuran file, warna, dan nilai bit.
15. Tombol informasi piksel, untuk menampilkan informasi piksel pada citra yang telah terrestorasi *denoising.*
16. Menampilkan histogram.

**4. Analisis Perhitungan Kualitas Citra**

Analisa yang akan dilakukan terhadap restorasi citra, adalah sebagai berikut:

**a. Analisis visual**, yaitu menganalisa hasil citra dengan berbagai nilai iterasi, dalam Tugas Akhir iterasi dibatasi 5, 10, 15, 20, dan 25

**b. Analisis statistika**, yaitu menghitung nilai kedalaman kapasitas citra (*Depth Resolution*) dan menghitung *Euclidean distance*

**1. Menghitung DR (*Depth Resolution*)**

*Depth Resolution* bertujuan untuk menguji perhitungan besar kapasitas citra. Namun, sebelum masuk dalam perhitungan agar mengubah terlebih dahulu citra dari *jpg menjadi *bmp untuk menguji perhitungan yang telah

dibuat benar atau masih belum benar. Berikut rumus

$$DR = \frac{File\ Size\ yang\ telah\ diketahui}{1000 + nilai\ bit} \quad (1)$$

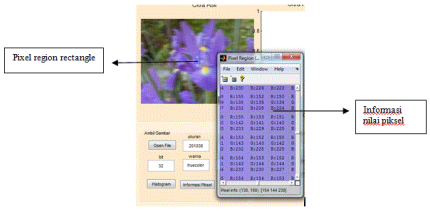
2. **Menghitung *Euclidean Distance***

*Euclidean distance* adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini biasanya diterapkan pada 2 dimensi dan 3 dimensi citra.

$$d(i,j) = \sqrt{(xi_1 - xj_1)^2} + \sqrt{(xi_2 - xj_2)^2} \quad (2)$$

## HASIL PENGUJIAN

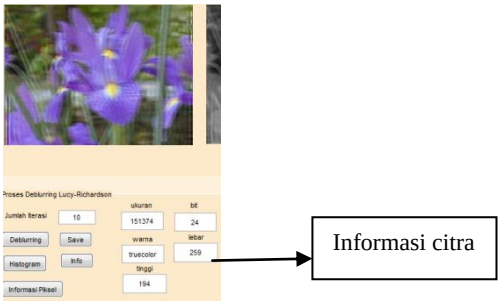
Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan sebagaimana mestinya dengan nilai iterasi yang telah ditentukan serta dilakukan sesuai dengan skenario uji coba yang dimulai dengan pengambilan citra, proses *deblurring*, proses *denoising* hingga proses u*nsharp*. Untuk memudahkan dalam pengujian, maka menampilkan terlebih dahulu matriks piksel.




*Gambar 4 menampilkan nilai piksel*

1. **Proses *Deblurring***

Pada tahap ini, citra asli yang telah ditampilkan beserta informasi data citra akan diproses *deblurring* untuk mengurangi tingkat kekaburan menggunakan algoritma Lucy-Richardson (L-R). Namun, sesuai teori bahwa algoritma L-R menggunakan estimasi PSF (*Point Spread Function*) melalui nilai iterasi, maka sebelum diproses pengguna memasukkan nilai iterasi terlebih dahulu.




*Gambar 5 Proses deblurring*

Untuk mendapatkan informasi perubahan nilai citra dan menampilkan histogram, maka ada proses *save* yang berfungsi menyimpan citra hasil *deblurring* sementara. Kemudian untuk menampilkan informasi perubahan nilai citra dengan menekan tombol info dan citra diambil dari media penyimpanan hasil proses *deblurring*. Untuk menampilkan nilai piksel RGB, tekan tombol informasi piksel, dan tampilan sama dengan proses pengambilan citra.

2. **Proses *Denoising***

Sesuai teori, bahwa semakin besar nilai iterasi, maka akan semakin tebal derau yang ada pada citra. Oleh karena itu, citra yang telah di*deblur*kan akan dikurangi derau dengan proses *denoising* mengunakan metode *Median Filter*. Namun, untuk proses ini citra terlebih dahulu diubah menjadi citra *grayscale* agar mempermudah pengurangan derau.




*Gambar 6 Pengubahan citra true color menjadi citra grayscale*

Proses perubahan citra *true color* menjadi citra *grayscale* dengan menekan tombol *Grayscale*. Proses *grayscale* yang digunakan adalah mengubah nilai piksel dengan cara memberi konstanta pengali. Berikut merupakan persamaan yang digunakan untuk mengubah derajat keabuan.

$$Gray = 0.3 * R + 0.59 * G + 0.11 * B \quad (3)$$




*Gambar 7 Proses denoising*

### 3. Analisa Visual

Analisa visual, merupakan bentuk analisa hasil percobaan dengan menampilkan citra-citra. Berikut hasil analisa proses *deblurring* dan *denoising* dengan berbagai tingkatan iterasi. Berikut infrmasi mengenai data citra yang digunakan sebagai bahan uji

*Tabel 1 Spesifikasi citra uji*

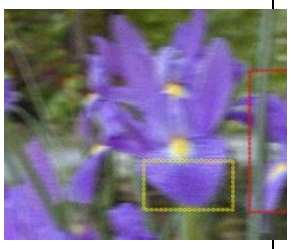
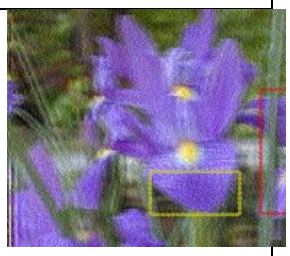
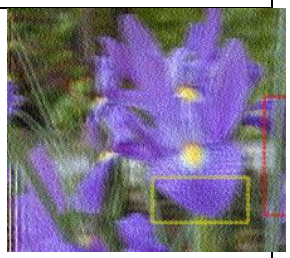
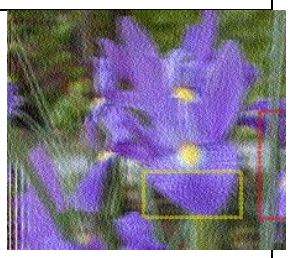
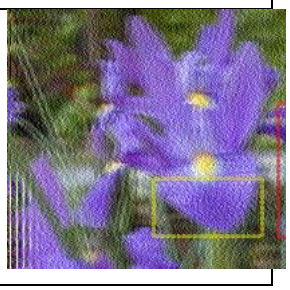
| **Citra** | **Nama** | **Kriteria Uji** | **Keterangan** |
|---|---|---|---|
|  | **Bunga** | 1. Bersolusi 259x194<br>2. Terdegradasi *motion blur* dan *gaussian noise*<br>***3.*** *True color*<br>4. 32 bit | **Dapat diproses** |

### a. Citra Hasil *Deblurring*

Citra dengan nama file bunga, resolusi 259x194 piksel, dan pengolahan *deblurring* dengan jumlah matriks *motion* sebesar 5x5

*Tabel 2. Citra hasil restorasi deblurring dan citra melalui proses unsharp untuk iterasi 5 sampai 25*

.

| *Citra rusak* | *Iterasi* | *Citra Deblurring* |
|---|---|---|
|  | *5* |  |
| | *10* |  |
| | *15* |  |
| | *20* |  |
| | *25* |  |

Pada tabel di atas menampilkan hasil restorasi *deblurring*. Secara visual terlihat bahwa algoritma Lucy-Richardson dapat memperbaiki efek kabur yang dibangkitkan. Namun, jika diperhatikan citra asli mengalami perbaikan yang signifikan hanya sampai iterasi ke-10, karena pada iterasi ke-20 dan iterasi ke-25 derau semakin tebal (*Artifact*), sehingga citra terlihat rusak kembali

dengan degradasi yang lain. Dilihat dari ketajaman warna ungu pada kelopak bunga, semakin tinggi nilai iterasi, maka warna semakin tajam dan jelas, begitu pula dengan batang daun pada sisi kanan citra tampak semakin tegas garis citranya.

Penebalan derau dan timbulnya garis-garis baru yang merupakan efek dari algoritma Lucy-Richardson mengakibatkan derau terlihat jelas. Oleh karena itu, pada program Tugas Akhir ini, membutuhkan proses *denoising*.

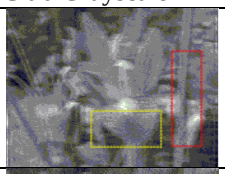
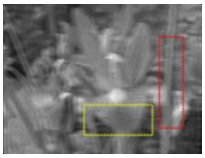
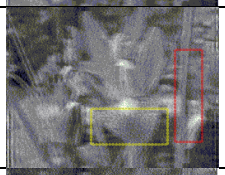
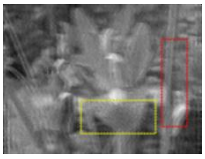
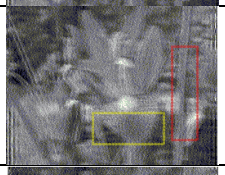
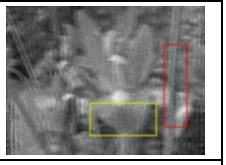
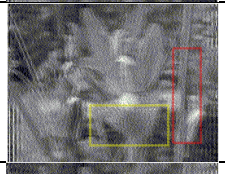
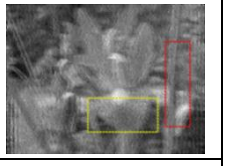
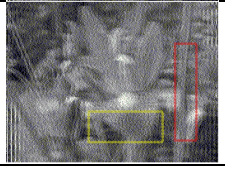
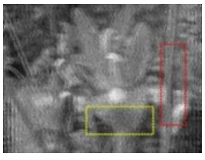
Jika diperhatikan dan dibandingkan dengan citra asli, maka proses *deblurring* dengan iterasi ke-10 lebih baik, sehingga dapat dikatakan nilai iterasi sebesar 10 merupakan iterasi yang cocok untuk citra diatas.

### b. Citra Hasil *Denoising*

Citra dengan nama file bunga dan resolusi 259x194 piksel serta pengolahan *denoising* menggunakan metode *median filter* dengan jumlah matriks *gaussian* sebesar 3x3.

Sebelum melakukan proses denoising, citra hasil deblurring terlebih dahulu dirubah menjadi grayscale, untuk mempermudah pengurangan derau.

*Tabel 2. Citra deblurring yang telah dirubah menjadi grayscale dan citra hasil restorasi denoising*

| Citra Grayscale | Iterasi | Citra Denoising |
|---|---|---|
|  | 5 |  |
|  | 10 |  |
|  | 15 |  |
|  | 20 |  |
|  | 25 |  |

Pada tabel diatas merupakan proses restorasi *denoising* yaitu mengurangi atau bahkan menghilangkan derau pada citra hasil *deblurring*. Hasil citra *denoising* terlihat berubah, sesuai dengan prinsip kerja metode *median filter* yaitu melembutkan atau menyamarkan derau. Sama halnya dengan proses deblurring, hasil citra yang mendekati baik terlihat ada pada iterasi ke-10.

## 4. Analisa Statistik

### a. Menghitung DR (*Depth Resolution*)

DR *(Depth Resolution)*, bertujuan untuk menghitung kapasitas citra (ukuran file) pada media penyimpanan (*Hardisk*) sebelum dan sesudah direstorasi. Jika perhitungan manual ini ukuran file sama dengan informasi yang ditampilkan komputer, maka proses restorasi dapat dikatakan berhasil. Ukuran file bersifat tetap jika nilai bit-nya tidak berubah.

**DR pada proses *deblurring***

Ukuran file yang diketahui dari citra bunga asli yang terdegradasi (rusak) dengan resolusi 259x194 adalah 201038 dengan kedalaman bit sebesar 32 bit, maka nilai DR adalah:

$$DR = \frac{201038}{1032} = 194{,}80\ atau\ 194\ \text{KB}$$

Setelah direstorasi, ukuran file berubah menjadi 151374 dengan kedalaman bit 24, maka ukuran DR adalah:

$$DR = \frac{151374}{1024} = 147{,}826\ atau\ 147KB$$

Dalam hal ini untuk menguji perhitungan besar kapasitas citra pada hasil proses *deblurring* berhasil dilakukan. Kapasitas dapat dibuktikan dengan melihat informasi ukuran pada citra.

**DR pada proses *denoising***

Ukuran file yang diketahui dari citra *deblurring* yang diubah menjadi citra *grayscale* adalah 119798 dengan kedalaman bit sebesar 8 bit.

$$DR = \frac{119798}{1008} = 118{,}847\ atau\ 118KB$$

Setelah direstorasi, ukuran file menjadi 51518.

$$DR = \frac{51518}{1008} = 51{,}10\ atau\ 51KB$$

Dalam hal ini untuk menguji perhitungan besar kapasitas citra pada hasil proses *denoising* sama dengan informasi ukuran file yang ditampilkan oleh komputer.

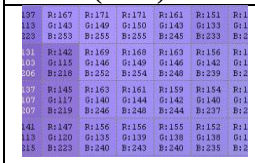
### b. Menghitung *Euclidean Distance*

Diketahui rumus Menghitung *Euclidean Distance* adalah menghitung jarak antara titik-titik piksel pada citra asli dengan citra hasil restorasi. Rumus *Euclidean Distance* yaitu $d(i,j) = \sqrt{(xi_1 - xj_1)^2} + \sqrt{(xi_2 - xj_2)^2} + \ldots dst$, yang dalam hal ini $X_i$ mewakili citra restorasi dan $X_j$ mewakili citra asli. Citra yang baik adalah citra yang nilai *Euclidean Distance* mendekati atau paling kecil. Untuk

menjadi bahan pembanding, maka nilai piksel citra asli diambil dari citra yang belum terdistorsi.

### 1. *Ecludean Distance* (ED) pada proses *deblurring*

*Tabel 3 Nilai piksel citra asli dengan matriks 5x4 pada koordinat [128 96] hingga koordinat [132 99]*

| Matriks Citra Asli (RGB) | Matriks Citra Asli [X] |
|---|---|
|  | 187 191 192 183 172<br>158 189 187 185 179<br>160 183 184 181 177<br>163 177 179 177 175 |

Berikut adalah nilai matriks citra asli yang rusak dan citra hasil *deblurring* dengan iterasi 5, 10, 15, 20 dan 25, sekaligus nilai *Ecludean Distance* (ED). Pengambilan matriks di koordinat yang sama yaitu mulai koordinat [128 96] hingga koordinat [132 99] dengan jumlah matriks 5x4.

*Tabel 4 Nilai Ecludean Distance pada citra rusak dengan matriks 5x4 pada koordinat [128 96] hingga koordinat [132 99]*

| Matriks Citra Rusak [X'] | $[X' - X] = [Y]$ | *ED* |
|---|---|---|
| 181 179 169 167 162<br>180 172 181 167 167<br>181 174 180 176 165<br>182 171 182 168 177 | -6 -12 -23 -16 -10<br>22 -12 -6 -22 -26<br>19 -9 -4 -15 -13<br>19 -6 3 -9 2 | 65 |

Tabel 5 *Nilai Ecludean Distance pada citra deblurring dengan matriks 5x4 pada koordinat [128 96] hingga koordinat [132 99]*

| Iterasi | Matriks *Deblurring* [X'] | Akar $\sum(X'-X)^2$ |
|---|---|---|
| 5 | 184 181 168 164 161<br>188 174 177 169 173<br>182 183 182 168 169<br>181 170 170 162 177 | 61.22 |
| 10 | 185 184 167 159 162<br>200 177 170 169 180<br>177 198 184 153 177<br>174 167 175 156 177 | 76 |
| 15 | 180 187 169 153 164<br>209 178 165 168 183<br>170 197 190 136 186<br>166 165 182 155 175 | 90.3 |
| 20 | 171 191 173 144 168<br>216 177 165 167 182<br>161 213 198 121 193<br>160 162 187 156 171 | 110.4 |
| 25 | 162 197 175 135 173<br>220 176 168 166 176<br>152 215 206 109 200<br>157 161 189 158 168 | 126.8 |

Sesuai teori bahwa prinsip dari *Ecludean Distance*, adalah mencari nilai jarak terkecil dari citra restorasi ke citra rusak atau lebih kecil jaraknya dibandingkan dengan jarak citra rusak dan citra asli sebagai analisis.

Melihat data pada tabel 4 bahwa citra rusak memiliki nilai ED sebesar 65, maka nilai ED pada citra restorasi harus lebih kecil dari nilai ED pada citra rusak. Pada tabel diatas, bahwa jarak yang lebih kecil dari jarak citra rusak adalah citra dengan iterasi ke-5. Pada iterasi ke-20 dan iterasi ke-25 nilai jarak ED terlampau lebih besar dari jarak ED citra rusak.

.

### 2. *Ecludean Distance* (ED) pada proses *denoising*

Untuk menjadi bahan pembanding, maka nilai piksel citra asli diambil dari citra yang belum terdistorsi.

Citra asli diubah menjadi citra grayscale terlebih dahulu, berikut adalah perhitungan ED proses *denoising* dari citra rusak dan citra asli sebagai pembanding.

*Tabel 6 Perhitungan ED pada citra rusak yang diubah menjadi citra grayscale dengan matriks 5x4 pada koordinat [128 96] hingga koordinat [132 99]*

| $[X' - X] = [Y]$ | $\sqrt{\sum Y^2}$ |
|---|---|
| -11 -15 -19 -12 -3<br>20 -6 -13 -15 -9<br>17 0 -9 -8 -5<br>15 4 -3 2 4 | 50 |

*Tabel 7 Nilai Ecludean Distance pada citra denoising dengan matriks 5x4 pada koordinat [128 96] hingga koordinat [132 99]*

| Iterasi | $[X' - X] = [Y]$ | $\sqrt{\sum Y^2}$ |
|---|---|---|
| 5 | -9 -14 -18 -11 -2<br>26 -3 -3 -13 -7<br>23 1 -10 -7 -4<br>19 6 -5 -3 -1 | 52.58 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | -6 | -14 | -18 | -10 | -8 | 52.04 |
| | 27 | -3 | -6 | -7 | -15 | |
| | 20 | -2 | -10 | -4 | -5 | |
| | 18 | -2 | -8 | 0 | -2 | |
| 15 | -6 | -11 | -19 | -11 | -6 | 52.10 |
| | 23 | -7 | -18 | -14 | -13 | |
| | 14 | -1 | -12 | -1 | -6 | |
| | 16 | 3 | -11 | 3 | 0 | |
| 20 | -6 | -15 | -11 | -13 | -3 | 53 |
| | 23 | -7 | -14 | -16 | -14 | |
| | 3 | -1 | -14 | -4 | -8 | |
| | 4 | -1 | -13 | 0 | -2 | |
| 25 | -7 | -13 | -18 | -8 | -4 | 53,27 |
| | 22 | -8 | -12 | -11 | -15 | |
| | 5 | -2 | -11 | -7 | -9 | |
| | 7 | -6 | -14 | -3 | -1 | |

Informasi yang didapat dari tabel diatas, seharusnya hasil ED proses *denoising* kurang dari nilai ED citra rusak. Namun, pada pengujian nilai ED proses denoising lebih besar dari nilai ED pada citra rusak. Begitu pula halnya dengan citra hasil *deblurring*, bahwa iterasi ke-20 dan ke-25 nilai jarak ED terlampau lebih besar dari jarak ED citra rusak, sehingga tidak cocok digunakan untuk proses restorasi citra bunga.

**KESIMPULAN**

Dari data yang diperoleh dalam pengujian tugas akhir ini dapat ditarik suatu kesimpulan bahwa :

a) Algoritma Lucy-Richardson mampu memperbaiki kualitas citra yang terdegradasi *motion blur* dengan iterasi 5 hingga 15.
b) Berdasarkan pengujian, proses *deblurring* menggunakan algoritma Lucy-Richardson dan proses *denoising* menggunakan metode *Median Filter* mampu memperlihatkan perbedaan secara visual dan statistik.
c) Matlab mampu mensimulasikan efek memperbaiki atau merestorasi citra tersebut dengan algoritma Lucy-Richardson secara sederhana, dengan mempergunakan image *processing toolbox* yang dimilikinya.
d) Jumlah iterasi yang digunakan pada algoritma Lucy Richardson dimulai dengan iterasi ke-5 hingga iterasi ke-25. Untuk perbaikan kualitas citra yang optimal dibutuhkan penelitian lebih lanjut mengenai hubungan antara jumlah iterasi dengan perbaikan kualitas citra tersebut.
e) Diharapkan kedepan, dapat menggunakan algoritma lain pada proses *deblurring*, misalnya dengan algoritma Wiener. Begitu pula dengan metode pada proses *denoising* dapat menggunakan metode lain, seperti metode *wavelet*.
f) Proses restorasi citra digital dapat diproses secara on-line.

**DAFTAR PUSTAKA**


1) Ahmad, U. (2005). *Pengolahan Citra Digital.* Bogor: Graha Ilmu

2) Gonzales, R. C., Woods, R. E., & Eddin, S. L. (2002). *Digital Image Processing Using Matlab.* New Jersey: Pearson Education.

3) Sigit, R., Basuki, A., & Ramadijanti, N. (2005). *Step by Step Pengolahan Citra Digital.* Surabaya: Penerbit Andi

4) Hendriyani, Y. (2012). *Restorasi Citra Kabur Menggunakan Lucy-Richardson , 05 No 2*, 14.

5) Ramayanti, D.(2010). Pengolaahan Citra. In *Tutorial Matlab.*