

BAB II

TINJAUAN PUSTAKA

2.1 Retinopati Diabetes

Retinopati diabetes adalah gangguan penglihatan yang disebabkan oleh penyakit diabetes. Gejala yang ditunjukkan oleh penderita retinopati diabetes antara lain *microneurisma*, *hemorrhages*, *hard exudates*, *soft exudates*, dan *neovascularis* (pembuluh darah baru). (Dillak, Rocky, 2012)

2.1.1 *Microneurisma*

Mikroneurisma merupakan titik merah kecil diantara pembuluh darah retina. Hal ini terjadi karena dinding pembuluh darah terkecil melemah kemudian pecah. Dalam beberapa kasus *microneurisma* ini meledak menyebabkan *hemorrhages*. Seiring kebocoran darah, lemak dan protein juga ikut keluar dari pembuluh darah titik terang kecil yang dinamakan *exudate*. (Fazri, Ahmad, 2013)

2.1.2 *Hemorrhages*

Hemorrhages merupakan kerusakan akibat retinopati diabetes berupa bercak-bercak merah darah akibat pecahnya *microneurisma*. (Fazri, Ahmad, 2013)

2.1.3 *Soft Exudates*

Soft exudates merupakan kerusakan akibat retinopati diabetes berupa bercak-bercak putih kecil kekuning-kuningan, kerusakan ini akan terus berlanjut dan akan meluas apabila tidak diangani dengan segera. (Fazri, Ahmad, 2013)

2.1.4 *Hard Exudates*

Hard exudates merupakan kerusakan akibat retinopati diabetes yang terlihat melebar dan membesar, kerusakan ini terus berlanjut bila tidak segera ditangani dengan baik dan mengakibatkan kondisi mata semakin parah. (Fazri, Ahmad, 2013)

2.2 Dataset Structured Analysis of the Retina (*STARE*)

STARE adalah proyek yang disusun oleh Michele Goldbaum, M.D., di University of California, San Diego. Proyek ini dimulai pada tahun 1975 dan didanai oleh A.S. National Institutes of Health. Dalam sejarahnya, lebih dari tiga puluh orang yang berkontribusi pada proyek ini, dengan latar belakang mulai dari kedokteran hingga *engineering*. Gambar dan *clinical data* disediakan oleh Shiley Eye Center di University of California, San Diego, dan Veterans Administration Medical Center di San Diego. (<http://cecas.clemson.edu/~ahoover/stare/>)

STARE merupakan database tentang retinopati diabetes yang banyak digunakan dalam penelitian ilmiah hingga sekarang. Contoh tentang penelitian retinopati diabetes yang menggunakan database *STARE* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Contoh Penelitian Menggunakan database *STARE*

NO	Peneliti	Tahun	Metode
1	E. Daniel, J. Anitha	2015	<i>OPGM+Adaptive GA</i>
2	Ramasubramanian.B, S.Selvaperumal	2016	<i>BOTHAT transform,</i> <i>CLAHE, Median filter,</i> <i>CLAHE</i>

NO	Peneliti	Tahun	Metode
3	Yu Qian Zhao, dkk	2014	<i>CLAHE, GABOR, Level Set, Region Growing</i>
4	Shuangling Wang, dkk	2014	<i>Gaussian Filtering, CNN</i>
5	Vahid Mohammadi Saffarzadeh, Alireza Osareh, Bitra Shadgar	2014	<i>Perceptive transform, Multiscale line operator, K-Means</i>
6	Miao Liao, dkk	2014	<i>Multiscale top hat transform, histogram fitting stretching</i>

Dimana database *STARE* berisikan 400 citra retina mata yang terdiri dari 41 cira retina normal dan 359 citra retinopati diabetes. Berikut adalah contoh citra retinopati diabetes yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 Retinopati Diabetes

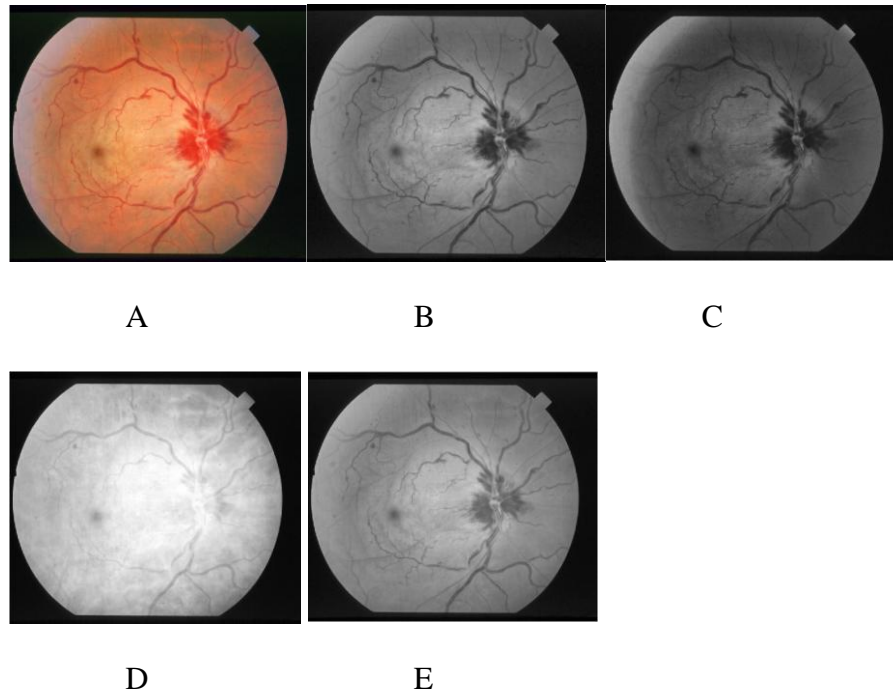
2.3 *Grayscale*

Grayscale atau abu-abu pada sebuah image digital adalah image yang pada setiap pikselnya hanya berisikan informasi intensitas warna putih dan hitam. Image grayscale memiliki banyak variasi nuansa abu-abu sehingga berbeda dengan image hitam putih. Grayscale juga disebut monokromatik karena tidak memiliki warna lain selain variasi intensitas putih dan hitam. Sebuah image yang dijadikan grayscale akan terkesan berbeda bila dibandingkan dengan image berwarna. (Nurullah, M, 2014)

. Pembentukan grayscale dapat dilakukan dengan dua cara yaitu dengan mengambil salah satu warna dalam citra atau mencari rata-rata warna dari sebuah citra tersebut. Pada kasus retinopati diabetes untuk mengambil warna keabuan dari citra menggunakan cara mengambil salah satu warna yang terdapat pada citra tersebut. Warna yang diambil adalah warna *green (hijau)* karena menghasilkan citra keabuan yang lebih bagus dibandingkan warna rata-rata, *red* ataupun *blue*. Pada grayscale menggunakan warna biru menghasilkan citra grayscale yang terlalu gelap, sedangkan pada grayscale menggunakan warna merah menghasilkan citra grayscale terlalu terang, begitupun dengan grayscale yang dihasilkan dengan menggunakan nilai rata-rata warna (RGB) menghasilkan citra grayscale yang kurang tajam atau terlalu terang, Hal ini dapat dilihat pada Gambar 2.2.

Dimana A merupakan citra asli retinopati diabetes, B merupakan citra grayscale yang didapatkan dengan cara mengambil warna hijau dari citra original, C merupakan citra grayscale yang didapatkan dengan cara mengambil warna biru dari citra original, D merupakan grayscale citra yang didapatkan dengan cara

mengambil warna merah dari citra original, dan E merupakan citra grayscale yang didapatkan melalui rata-rata dari citra original ($red+blue+green / 3$).



Gambar 2.2. (A) Citra original dan (B-E) Citra *grayscale*

2.4 Transformasi *Spatial Domain*

Transformasi spatial domain merupakan teknik peningkatan mutu citra yang melakukan manipulasi langsung pixel (x,y) suatu citra dengan menggunakan fungsi transformasi: $g(x, y) = T[f(x, y)]$, dimana $f(x, y)$ sebagai citra input, $g(x, y)$ hasil citra yang sudah diproses dan T adalah operator pada f yang didefinisikan berdasarkan beberapa lingkungan di (x, y) . (Russ, John C, 2007)

Masking / Filter suatu pixel (x,y) ditentukan berdasarkan pixel tetangganya yang didefinisikan sebagai bentuk bujur sangkar (sering digunakan) ataupun circular sebagai sub-citra yang berpusat di titik (x,y) dengan ukuran lebih dari 1×1 .

Pusat sub-citra berpindah dari satu pixel ke pixel lainnya dimulai dari pojok atas. Nilai koefisien masking ditentukan berdasarkan prosesnya. Teknik *masking* digunakan untuk penajaman citra dan penghalusan citra. (Russ, John C, 2007)

2.5 *Hamony Search*

Harmony search adalah salah satu algoritma metaheuristik yang dikemukakan oleh Geem et al pada tahun 2001, algoritma ini dikonseptualisasikan menggunakan proses *musikal* untuk mencari *perfect satae of harmony*. Para pemusik mencari harmony yang tepat berdasarkan *aesthetic satandard*, sama seperti proses optimasi yang mencari global solution seperti yang ditentukan berdasarkan *objective function*. (Geem, Zong Woo, 2009)

Pitch dari setiap instrument musik menentukan *aesthetic quality* dari masing-masing instrument, sama seperti *object function value* ditentukan melalui suatu *set value* yang diatur kedalam *decision variable* masing-masing. Algoritma *harmony search* ini diturunkan berdasarkan kinerja proses alami yang terjadi ketika seorang musisi mencari *harmony* yang lebih baik, seperti ketika improvisasi musik jazz. (Geem, Zong Woo, 2009)

Dalam improvisasi musik, masing-masing musisi menyembunyikan sebuah nada dalam tingkat nada yang mungkin dihasilkan. Nada-nada tersebut membuat sebuah vector *harmony*. Jika semua nada menghasilkan *harmony* yang bagus, maka pengalaman membuat *harmony* tersebut akan tersimpan diingatan para musisi, sehingga kemungkinan untuk membuat *harmony* bagus lainnya meningkat. Hal ini diadaptasi dalam teknik optimasi, dengan masing-masing variabel keputusan

memilih nilai awal (initial value) dalam batasan yang mungkin membentuk sebuah vector solusi. Jika masing-masing variabel keputusan membentuk *harmony* yang baik, pengalaman tersebut akan disimpan dalam variabel memori, yang nantinya akan memperbesar kemungkinan untuk mendapatkan vector solusi yang lebih baik. (Geem, Zong Woo, 2009)

Ada beberapa strategi yang bisa digunakan untuk mendapatkan harmonisasi yang baik. Pada pertunjukan musik yang dilakukan oleh para musisi jazz, biasanya mereka memiliki tiga pilihan untuk melakukan improvisasi. (Geem, Zong Woo, 2009)

1. Mengambil nilai yang sudah terdapat pada memori *harmony*, atau yang sering disebut dengan *Memory Considerations*.
2. Mengambil nilai yang berdekatan dari suatu nilai pada memori *harmony* (*pitch adjustments*).
3. Mengambil nilai acak dari rentang nilai yang mungkin, biasanya didefinisikan sebagai *randomization*.

Tiga aturan ini nantinya akan dibutuhkan dalam menyusun algoritma *harmony search* yang diwakilkan oleh beberapa parameter. Secara umum parameter-parameter ini akan sangat mempengaruhi kinerja dari *harmony search* dalam mendapatkan solusi yang optimal.

Berikut adalah parameter yang digunakan secara umum dalam *harmony search*. (Geem, Zong Woo, 2009)

1. *HMS* (*Harmony Memory size*) adalah jumlah dari kumpulan *harmony memory*, atau disebut juga sebagai jumlah populasi di algoritma genetika.

2. *HMCR (Harmony Memory Consideration Rate)* adalah probabilitas dari *harmony memory* untuk digunakan kembali sebagai hasil dari vektor solusi. Parameter ini memiliki rentang nilai dari 0 sampai dengan 0,99. Parameter ini tidak menggunakan nilai 1, untuk mencegah terjadinya stagnansi (range nilai tidak berubah dan perbaikan minimal).
3. *PAR (Pitch Adjustment Rate)* adalah parameter yang mempunyai peran signifikan dalam menentukan jumlah nilai yang harus diubah, disesuaikan, atau ditukar dengan nilai yang lain. Parameter ini memiliki rentang nilai dari 0 sampai dengan satu dan dibawah nilai *HMCR*.
4. *NI (Number of Improvisations)* adalah parameter yang digunakan sebagai kriteria pemberhenti perulangan, biasanya berupa jumlah iterasi untuk mendapatkan nilai optimal.

Langkah-langkah yang digunakan dalam melakukan algoritma *harmony search* adalah. (Geem, Zong Woo, 2009)

1. Inisialisasi *Harmony search Algorithm (HSA)* dan parameter-parameter permasalahan optimasi.
2. Inisialisasi *Harmony Memory (HM)*.

$$x(j) = lb(j) + (ub(j) - lb(j)) * rand(0,1) \quad (i)$$

dimana $lb(j)$ merepresentasikan batas bawah dan $ub(j)$ merepresentasikan batas atas dari sebuah parameter. Sedangkan $rand(0,1)$ adalah nilai acak dari 0 sampai 1.

3. Improvisasi *harmony* baru.
4. Memperbarui *Harmony Memory (HM)*.

5. Melakukan pengecekan kondisi berhenti.

Dalam langkah inisialisasi *HSA*, perlu diperhatikan fungsi objektif yang akan digunakan untuk menghitung nilai perbaikan dari *HM*. Fungsi objektif bisa dispesifikasikan pada notasi nomor 1. (Geem, Zong Woo, 2009)

$$\mathbf{Min} \{ f(x) \mid x \in X \} \text{ atau } \mathbf{Max} \{ f(x) \mid x \in X \} \quad (1)$$

Keterangan notasi nomor 1 diatas adalah sebagai berikut.

1. $F(x)$ adalah fungsi objektif.
2. x adalah himpunan dari variabel keputusan x_i .
3. X adalah himpunan dari rentang nilai yang mungkin untuk nilai variabel keputusan.

Dalam langah ke dua, *HM* adalah vektor solusi dengan banyak sejumlah *HMS*. Pada langkah ini solusi-solusi yang dibentuk merupakan nilai random. Bentuk *HM* bila digambarkan dalam matriks bisa berbentuk seperti pada Gambar 2.3. (Geem, Zong Woo, 2009)

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} \end{bmatrix} \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HMS}) \end{bmatrix}$$

Gambar 2.3 ilustrasi *harmony memory* (Geem, Zong Woo, 2009)

Untuk setiap X_i dengan $i \in \{1 \dots HMS\}$, perlu ditentukan nilai fungsi objektifnya $f(x_i)$, agar bisa diketahui mana himpunan yang memiliki nilai terbaik dan nilai terburuk. Dalam prosesnya, himpunan dengan nilai terbaik akan menjadi

solusi yang dipakai, sedangkan himpunan dengan nilai yang lebih baik sekana perulangan berlangsung. (Geem, Zong Woo, 2009)

Dalam langkah ke tiga, *HSA* akan secara otomatis membuat vektor solusi yang baru, berdasarkan pada tiga operator yaitu (Geem, Zong Woo, 2009)

a Memory Consideration

Pada operator ini , pencarian variabel keputusan akan berdasarkan pada memori yang tersimpan sebelumnya di posisi yang sama dengan kemungkinan pemilihan operator ini sebesar parameter *HMCR*, dengan kemungkinan ($0 \leq HMCR \leq 1$).

b Random Consideration

Operator ini akan terlaksana dengan kemungkinan ($1-HMCR$), yang merupakan alternatif apabila operator *Memory Consideration* tidak digunakan dalam mencari vektor solusi.

c Pitch Adjustment

Operator ini akan terlaksana dengan prasyarat, terlaksananya operator *Memory Consideration*. Operator ini akan melakukan improvisasi nilai dari variabel keputusan secara lokal. Operator ini memiliki kemungkinan terjadi ($PAR * HMCR$).

Dalam langkah ke 4, jika hasil vektor *harmony* memiliki nilai fungsi objektif yang lebih baik dari vektor *harmony* dengan nilai fungsi objektif terburuk di *HM*, maka vektor *harmony* yang baru akan dimasukkan ke dalam *HM*, sedangkan vektor *harmony* dengan nilai fungsi objektif terburuk di *HM* akan dikeluarkan.

Dalam langkah ke 5, akan dilakukan pengecekan sampai kondisi berhenti dari *harmony search* terpenuhi. Biasanya kondisi berhenti dari algoritma adalah nilai dari parameter *NI* (jumlah maksimal dari perbaikan *harmony*). Bila kondisi berhenti salah, maka ulangi langkah ke 3 dan ke 4.

2.6 Sobel Operator

Peningkatan kualitas citra (*image enhancement*) bertujuan menghasilkan citra dengan kualitas yang lebih baik dibandingkan dengan citra semula. Langkah selanjutnya dalam pengolahan citra adalah analisis citra (*image analysis*). Analisis citra bertujuan mengidentifikasi parameter-parameter yang diasosiasikan dengan ciri (*feature*) dari objek di dalam citra, untuk selanjutnya parameter tersebut digunakan dalam menginterpretasi citra. Analisis citra pada dasarnya terdiri dari tiga tahapan yaitu ekstraksi fitur (*feature extraction*), segmentasi, dan klasifikasi. (Sonka, Milan, 2015)

Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi (*edge*) dari objek di dalam citra. Setelah setiap tepi objek diketahui, langkah selanjutnya dalam analisis citra adalah segmentasi, yaitu mereduksi citra menjadi objek atau region. Langkah terakhir dari analisis citra adalah klasifikasi, yaitu memetakan segmen-segmen yang berada ke dalam kelas objek yang berbeda pula. Salah satunya operator deteksi tepi adalah *sobel operator*. (Sonka, Milan, 2015)

Sobel operator adalah salah satu deteksi tepi yang sering digunakan karena kompleksitas matematika yang sederhana serta hasil yang didapatkan lebih bagus

di bandingkan operator deteksi tepi yang lainnya. (Suresh, Shilpa & Shyam Lal, 2017) Perumusan sobel operator dapat dilihat pada persamaan nomor 2, 3, dan 4.

$$sml(i, j) = g(i + 1, j - 1) + 2g(i + 1, j) + g(i + 1, j + 1) - g(i - 1, j - 1) - 2g(i - 1, j) - g(i - 1, j + 1) \quad (2)$$

$$snl(i, j) = g(i - 1, j + 1) + 2g(i, j + 1) + g(i + 1, j + 1) - g(i - 1, j - 1) - 2g(i, j - 1) - g(i + 1, j - 1) \quad (3)$$

$$Sobel = \sqrt{sml(i, j)^2 + snl(i, j)^2} \quad (4)$$

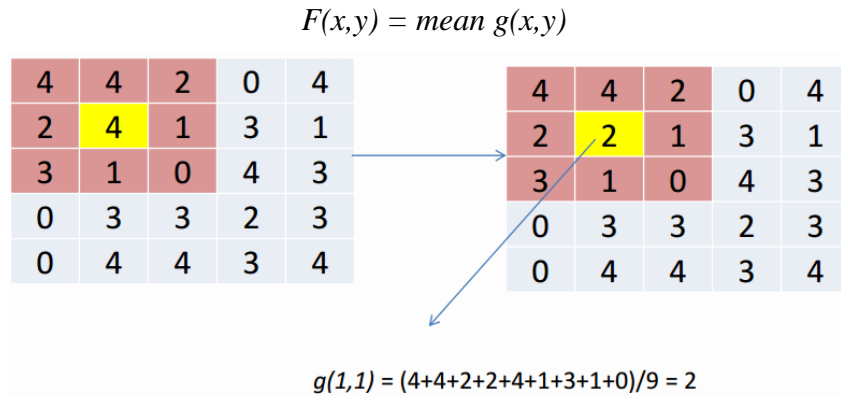
2.7 Filter Mean

Noise (derau) merupakan suatu kondisi cacat pada citra yang muncul dalam citra saat proses perubahan atau kompresi citra. Intensitas *noise* tinggi maupun rendah dapat menurunkan kualitas citra dan menyebabkan hilangnya beberapa detail informasi citra. Untuk menghilangkan *noise*, ada beberapa teknik *filtering*. Sebelumnya dikenal *filter* linier, dimana filter ini menyebabkan citra menjadi kabur. Kemudian dikembangkan *filter nonlinier* yang bisa memperbaiki kekurangan *filter* linier. Salah satunya adakah *filter mean* yang menggunakan informasi intensitas piksel dalam jendela *filtering* dan mengganti piksel yang pada pusat jendela dengan nilai *mean* (rata-rata). (Russ, John C, 2007)

Filter mean bekerja dengan mengganti nilai suatu piksel pada citra asal (pusat citra) dengan nilai rata-rata dari piksel asal citra tersebut berdasarkan suatu

lingkungan tetangga. Ilustrasi perhitungan filter mean dapat dilihat pada Gambar

2.4. (Russ, John C, 2007)



Gambar 2.4 Ilustrasi Filter mean (Naf'an, Muhammad Zidny, Lc., S.Kom., M.Kom., 2015)

2.8 Local Enhancement

Local Enhancement (LE) adalah peningkatan kontras dari citra yang didasarkan oleh distribusi tingkat keabuan pada sekitar piksel tetangga (*neighborhood*). (Braik, malik, 2007) *Local enhancement* dapat dirumuskan pada persamaan nomor 5.

$$g(i,j) = \frac{D}{\sigma(i,j)} [f(i,j) - m(i,j)] \quad (5)$$

Dimana $m(i,j)$ adalah *local mean* dan D adalah *global mean* direpresentasikan pada persamaan 6, serta 7.

$$m(i,j) = \frac{1}{n * n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x,y) \quad (6)$$

$$D = \frac{1}{M * N} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x,y) \quad (7)$$

Dan *standard deviasi* direpresentasikan pada persamaan 8.

$$\sigma(i, j) = \sqrt{\frac{1}{(n*n)-1} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} (f(x, y) - m(i, j))^2} \quad (8)$$

Dimana M adalah mendefinisikan kolom suatu matriks piksel citra dan N mendefinisikan baris suatu matriks piksel citra. Sedangkan $n \times n$ mendefinisikan banyaknya ketetanggan (*neighborhood*) matiks citra serta $f(i, j)$ didapatkan dari matiks keabuan dari citra asal. (Al-betar, Mohammed dkk. 2016).

2.9 Local Standard Deviation

Local Standard Deviation (LSD) adalah peningkatan kontras yang menggunakan teknik local enhancement. Metode ini dapat digunakan dalam banyak kasus seperti *medical images*, *real time images*, dan *sebagainya*. Namun dalam beberapa image metode ini dapat berubah menjadi 0 sehingga dapat menghasilkan gambar yang tidak diinginkan. (Singh, S, 2012) Berikut adalah persamaansi dari metode *LSD* yang direpresentasikan pada persamaan nomor 9.

$$g(i, j) = \frac{D}{\sigma(i, j)} ([f(i, j) - m(i, j)] + m(i, j)) \quad (9)$$

2.10 Modification Local Standard Deviation

Modification Local Standard Deviation (MLSD) peningkatan citra yang merupakan turunan dari metode *LSD*. Berikut adalah persamaansi dari metode *MLSD* yang direpresentasikan pada persamaan nomor 10

$$g(i, j) = \frac{k*D}{\sigma(i, j)+b} ([f(i, j) - c * m(i, j)] + m(i, j)^a) \quad (10)$$

Perbedaan antara perumusan *LE*, *LSD* dan *MLSD* adalah terletak pada parameter. Pada *LE* $m(i, j)^a$ didefinisikan sebagai 0, b didefinisikan sebagai 0, c dan k didefinisikan sebagai 1. Pada *LSD* a didefinisikan sebagai 1, b didefinisikan sebagai 0, c dan k didefinisikan sebagai 1. Sedangkan pada perumusan *MLSD* $b \neq 0$, dan $c \neq 0$ melainkan $a \in [0, 1, 5]$, $b \in \left[0, \frac{D}{2}\right]$, $c \in [0, 1]$, $k \in [0, 5, 1, 5]$.(Braik, Malik, 2007) (Singh, Nirmal, 2013)

2.11 Global Local Contrast Enhancement

Global Local Contrast Enhancement (GLCE) adalah teknik pemerataan kontras yang dihasilkan dari kombinasi *local information* dan *global information* dari image.(Singh, Archana, 2016) Berikut adalah perumusan dari *GLCE* yang direpresentasikan pada persamaan nomor 11.

$$g(i, j) = f(i, j) + [m(i, j) - \text{global Mean}] \quad (11)$$

Dimana $f(i, j)$ merupakan nilai piksel dari image original, $m(i, j)$ adalah *filter mean* dengan ordo 3×3 , serta *global mean* adalah rata-rata dari keseluruhan intensitas image. (Singh, Archana, 2016)

2.12 AMBE (Absolute Mean Brighness Error)

AMBE adalah algoritma yang digunakan untuk mengukur *error brighness* dari citra. Perumusan *AMBE* dapat dilihat pada persamaan nomor 12.

$$AMBE = |I - I'| \quad (12)$$

Dimana I mendefinisikan rata-rata dari citra original sedangkan I' mendefinisikan rata-rata dari citra *enhancement*. Semakin kecil hasil yang dihasilkan dari perumusan *AMBE* maka semakin baik kualitas citra yang didapatkan dalam mempertahankan pencahayaannya. (Daniel, Ebenezer dan Anita, J , 2015)

2.13 Entropy

Entropy adalah suatu metode yang digunakan untuk mengambil informasi pada suatu citra. Semakin tinggi nilai *entropy* dari sebuah citra maka semakin baik pula pemerataan intensitas histogramnya. Perumusan *entropy* dapat dilihat pada persamaan nomor 13. (Suresh, Shilpa & Shyam Lal, 2017)

$$H(e) = - \sum_{i=0}^{Lmax-1} p_e(i) \log_2(p_e(i)) \quad (13)$$

Dimana $Lmax$ nilai maksimum intensitas dari citra, sedangkan $p_e(i)$ merupakan probabilitas dari kemunculan piksel. (Suresh, Shilpa & Shyam Lal, 2017)

2.14 PSNR (Peak Signal to Noise Ratio)

PSNR adalah nilai perbandingan antara dua objek dalam satuan desibel. Semakin besar nilai *PSNR* berarti perbedaan antara file asli dengan file hasil modifikasi semakin kecil, sebaliknya semakin kecil nilai *PSNR*, maka perbedaan antara file asli dengan file hasil modifikasi semakin besar. Perumusan *PSNR* dapat dilihat pada persamaan nomor 14.

$$PSNR = 10 \log_{10} * (255^2 / MSE) \quad (14)$$

Sedangkan perhitungan MSE dapat dilihat pada persamaan nomor 15.

$$MSE = \frac{1}{M*N} * \sum_{i=0}^M \sum_{j=0}^N |f(i,j) - f'(i,j)|^2 \quad (15)$$

Dimana MSE merupakan perhitungan nilai selisih antara file asli dengan file modifikasi, $f(i,j)$ menyatakan file asli, $f'(i,j)$ menyatakan file modifikasi, dan $M * N$ menyatakan panjang dan lebar file. (Daniel, Ebenezer dan Anita, J , 2015)

2.15 Kombinasi MLSD dengan GLCE

Pada tugas akhir ini metode yang digunakan adalah kombinasi dari perumusan dari metode $MLSD$ dengan metode $GLCE$. Perumusan dari metode yang diusulkan dapat dilihat pada persamaan nomor 16.

$$g(i,j) = \frac{k*D}{\sigma(i,j)+b} ([f(i,j) - c * m(i,j)] + m(i,j)^a) + ((m(i,j) - D)/2) \quad (16)$$

Dimana $m(i,j)$ merepresentasikan *local mean / filter mean*, $f(i,j)$ merepresentasikan matriks grayscale, $\sigma(i,j)$ merepresntasikan *standard deviation*, D merepresentasikan *global mean*, nilai a terdiri dari rentang nilai (0 - 1,5), nilai b (0 – globalMean / 2), nilai c (0 - 1), dan k (0,5 – 1,5). (Singh, Nirma 2013). Untuk mencari nilai parameter a,b,c,k pada tugas akhir ini digunakan metode optimasi *harmony search*.

Harmony search dipilih karena mempunyai kelebihan dibandingkan metode – metode optimasi yang lain yaitu lebih sedikitnya persyaratan matematika yang

digunakan dalam proses pencarian awal, mempunyai jumlah iterasi yang lebih sedikit, selain itu metode ini dapat dikatakan sederhana, mudah beradaptasi, dan terukur. (woo Geem, Zong , 2009) Pemilihan kandidat terbaik dari parameter a, b, c, k yang dihasilkan oleh algoritma *harmony search* didasarkan dari nilai *fitness* (fungsi objektif) yang paling tinggi. Perhitungan nilai *fitness* didapatkan dengan mengkombinasikan nilai *entropy*, jumlah intensitas, serta informasi *edge* dari sebuah citra. (Albetar, dkk , 2016)

$$\max F(I_e) = \log(\log(E(I_s))) * \frac{n_{edges}}{M * N} * H(I_e) \quad (17)$$

Dimana $E(I_s)$ merepresentasikan penjumlahan intensitas *edge* yang terdeteksi oleh *sobel* dari hasil matriks dari persamaan nomor 10, n_{edges} merepresentasikan jumlah garis yang terdeteksi dalam proses threshold *sobel*, $H(I_e)$ merepresentasikan perhitungan *entropy* dari persamaan nomor 10, serta $M * N$ merepresentasikan tinggi dan lebar dari ukuran citra. (Albetar, dkk , 2016)

n_{edges} atau banyak garis didapatkan dari perumusan $nnz(\text{Number Non Zero})$, maksudnya adalah menghitung banyak *value* piksel yang lebih dari 0. Dimana pada kasus ini n_{edges} didapatkan dari matriks *threshold* dari *sobel operator* dengan persamaan nomor 18.

$$th = \frac{sobel}{ni_{maxSobel}} \quad (18)$$

Dimana *Sobel* merepresentasikan *value matriks sobel* sedangkan $ni_{maxSobel}$ merepresentasikan nilai maximum intensitas *sobel operator*. (Nurullah, M, 2014)