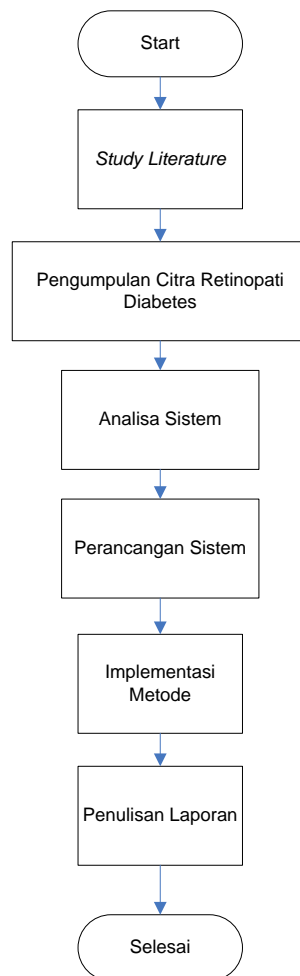


BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Tahapan Penelitian

Untuk menyelesaikan penelitian ini ada beberapa tahapan yang harus dikerjakan. Tahapan yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Metodologi Penelitian

3.2 *Study Literatur*

Penelitian selalu diawali dengan proses pengkajian yang berkaitan dengan topik penelitian yang diambil. Proses ini meliputi pencarian referensi pendukung yang sesuai dan dapat dipertanggung jawabkan. Referensi yang digunakan dapat berupa buku jurnal maupun artikel seperti, peningkatan kontras, retinopati diabetes, transformasi spasial domain, *harmony search* dan metode pengujian untuk peningkatan kontras. Informasi yang didapat nantinya berguna untuk mendesain algoritma dalam memecahkan permasalahan yang sedang diteliti. Setelah study literature dilakukan, pendefinisian masalah dapat diketahui lebih detail.

3.3 **Pengumpulan Dataset**

Pada tahap ini dilakukan pengumpulan dataset. Dataset yang digunakan dalam penelitian ini adalah citra retinopati diabetes yang diambil dari database *STARE*.

3.4 **Analisa Sistem**

Tahap ini membahas tentang analisa kebutuhan fungsionalitas sistem, batasan sistem serta masukan dan keluaran sistem. Sistem yang dibangun harus mampu membaca citra masukan dengan format ppm, dan meningkatkan kontras citra.

3.5 **Perancangan Sistem**

Dalam penelitian ini akan dilakukan beberapa tahapan meliputi tahap implementasi metode *GLCE*, serta metode yang diusulkan yaitu *GLCE* berbasis *harmony search* (kombinasi *MLSD+HS* dengan *GLCE*). Tahapan yang diperlukan untuk mengimplementasi metode *GLCE*, dan *GLCE+HS* yaitu *input image*, mengubah citra RGB menjadi *grayscale*, *local mean*, *global mean*, *standard*

deviation, harmony search. Dimana ilustrasi rancangan sistem yang akan dikerjakan ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Flowchart Rancangan Sistem

3.5.1 Input Image

Data masukan yang digunakan dalam sistem adalah citra RGB retinopati diabetes yang berukuran 605 x 700 piksel dengan format PPM. Setelah didapat citra retinopati diabetes yang akan diolah tahap selanjutnya adalah merubah citra RGB ke citra grayscale.

3.5.2 Grayscale

Pada tahap ini citra RGB retinopati diabetes dikonversi menjadi citra grayscale. Hal ini dilakukan untuk menyederhanakan model citra. Cara mendapatkan citra grayscale retinopati diabetes ini dengan mengambil warna green dari citra RGB retinopati diabetes yang telah dijelaskan pada bab 2.3. Setelah didapatkan citra grayscale tahap selanjutnya adalah mencari *local mean* dan *global mean*.

3.5.3 Local Mean dan Global Mean

Local mean / filter mean pada kasus ini digunakan untuk mencari informasi lokal dari sebuah citra, yang bergantung pada karakteristik *neighborhood*. Berikut adalah contoh perhitungan manual dari *filter mean* pada citra keabuan 5x5 dengan ketetanggaan (n) 3x3 yang direpresentasikan pada Tabel 3.2. Dimana matriks grayscale dari citra retinopati diabetes diumpamakan dengan Tabel 3.1.

Tabel 3.1 Matriks *Grayscale*

4	4	2	0	4
2	4	1	3	1
3	1	0	4	3
0	3	3	2	3
0	4	4	3	4

Tabel 3.2 Perhitungan *Filter mean*

no	$g(x,y)$	Perhitungan	Hasil	Piksel Yang Diganti
1	1,1	$(4+4+2+2+4+1+3+1+0)/(3*3)$	$2,33 \Rightarrow 2$	4
2	1,2	$(4+2+0+2+1+3+1+0+4)/(3*3)$	$1,89 \Rightarrow 2$	1
3	1,3	$(2+0+4+2+3+1+0+4+3)/(3*3)$	$2,11 \Rightarrow 2$	3
4	2,1	$(2+2+2+3+1+0+0+3+3)/(3*3)$	$1,78 \Rightarrow 2$	1
5	2,2	$(2+2+2+2+0+4+3+3+2)/(3*3)$	$2,22 \Rightarrow 2$	0
6	2,3	$(2+2+1+2+4+3+3+2+3)/(3*3)$	$2,44 \Rightarrow 2$	4
7	3,1	$(3+2+2+0+3+3+0+4+4)/(3*3)$	$2,33 \Rightarrow 2$	3
8	3,2	$(2+2+2+2+3+2+4+4+3)/(3*3)$	$2,67 \Rightarrow 3$	3
9	3,3	$(2+2+3+3+2+3+4+3+4)/(3*3)$	$2,89 \Rightarrow 3$	2

Sedangkan hasil akhir dari perhitungan *filter mean* dapat dilihat pada Tabel

3.3.

Tabel 3.3 Matriks *Filter mean*

4	4	2	0	4
2	2	2	2	1
3	2	2	2	3
0	2	3	3	3
0	4	4	3	4

Sedangkan *global mean* pada kasus ini digunakan untuk mencari informasi keseluruhan/global tentang penyimpangan dari sebuah citra. Nilai *global mean* didapatkan dari jumlah keseluruhan nilai piksel yang dihasilkan oleh *matriks grayscale* dibagi luas daerah citra (nilai rata-rata dari sebuah citra). Berikut adalah contoh perhitungan manual dari *global mean*.

$$\text{Global Mean} = \frac{4+4+2+0+4+2+4+1+3+1+3+1+0+4+3+0+3+3+2+3+0+4+4+3+4}{5 \times 5} = 2,48$$

Setelah mendapatkan hasil matriks dari *filter mean* langkah selanjutnya adalah mencari nilai *standard deviation*.

3.5.4 Standard Deviation

Standard deviation pada kasus ini digunakan untuk mencari homogenitas antara matriks grayscale dengan *filter mean*. Berikut adalah perhitungan manual *Standard Deviation*.

Iterasi 1

$$\sqrt{\frac{(4-4)^2 + (4-4)^2 + (2-2)^2 + (2-2)^2 + (4-2)^2 + (1-2)^2 + (3-3)^2 + (1-2)^2 + (0-2)^2}{(3 \times 3) - 1}} = 1,118034$$

Iterasi 2

$$\sqrt{\frac{(4-4)^2+(2-2)^2+(0-0)^2+(4-2)^2+(1-2)^2+(3-2)^2+(1-2)^2+(0-2)^2+(4-2)^2}{(3*3)-1}} = 1,369306$$

Iterasi 3

$$\sqrt{\frac{(2-2)^2+(0-0)^2+(4-4)^2+(1-2)^2+(3-2)^2+(1-1)^2+(0-2)^2+(4-2)^2+(3-3)^2}{(3*3)-1}} = 1,118034$$

Iterasi 4

$$\sqrt{\frac{(2-2)^2+(4-2)^2+(1-2)^2+(3-3)^2+(1-2)^2+(0-2)^2+(0-0)^2+(3-2)^2+(3-3)^2}{(3*3)-1}} = 1,172604$$

Iterasi 5

$$\sqrt{\frac{(4-2)^2+(1-2)^2+(3-2)^2+(1-2)^2+(0-2)^2+(4-2)^2+(3-2)^2+(2-3)^2+(4-4)^2}{(3*3)-1}} = 1,1457738$$

Iterasi 6

$$\sqrt{\frac{(1-2)^2+(3-2)^2+(1-1)^2+(0-2)^2+(4-2)^2+(3-3)^2+(3-3)^2+(2-3)^2+(3-3)^2}{(3*3)-1}} = 1,172604$$

Iterasi 7

$$\sqrt{\frac{(3-3)^2+(1-2)^2+(0-2)^2+(0-0)^2+(3-2)^2+(3-3)^2+(0-0)^2+(4-4)^2+(4-4)^2}{(3*3)-1}} = 0,866025$$

Iterasi 8

$$\sqrt{\frac{(1-2)^2+(0-2)^2+(4-2)^2+(3-2)^2+(3-3)^2+(2-3)^2+(4-4)^2+(4-4)^2+(3-3)^2}{(3*3)-1}} = 1,172604$$

Iterasi 9

$$\sqrt{\frac{(0-2)^2+(4-2)^2+(3-3)^2+(3-3)^2+(2-3)^2+(3-3)^2+(4-4)^2+(3-3)^2+(4-4)^2}{(3*3)-1}} = 1,06066$$

Sehingga didapatkan hasil matriks *standard deviation* seperti yang ditunjukkan pada

Tabel 3.4.

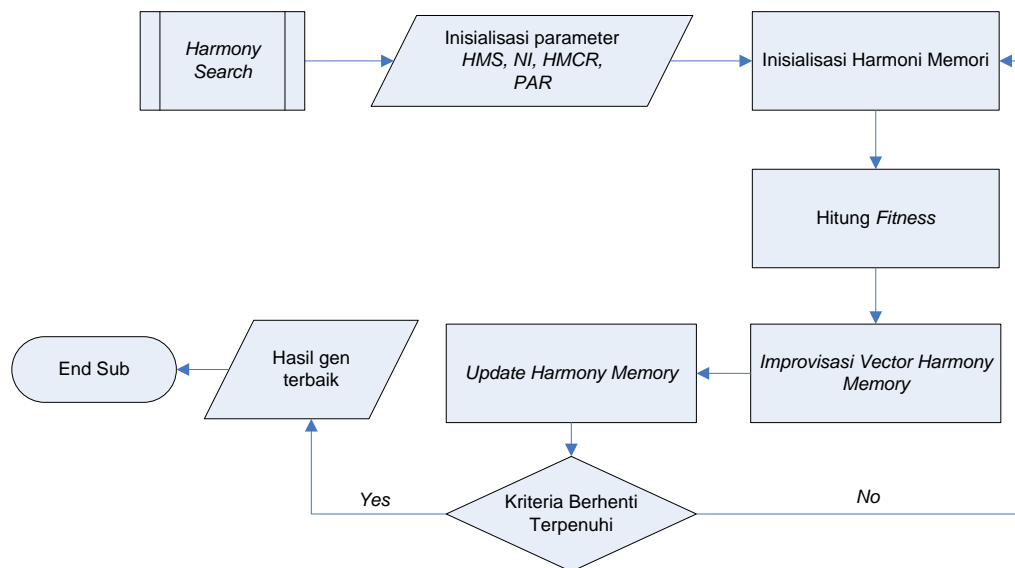
Tabel 3.4 Matriks *Standard Deviation*

4	4	2	0	4
2	1,118034	1,369306	1,118034	1
3	1,172604	1,457738	1,172604	3
0	0,866025	1,172604	1,06066	3
0	4	4	3	4

3.5.5 Harmony Search

Setelah menghitung algoritma *Standard Deviation* tahap selanjutnya adalah proses perhitungan algoritma *harmony Search*. *Harmony search* pada kasus ini difungsikan untuk pencarian otomatis terhadap parameter yang terdapat pada perumusan *GLCE* berbasis *harmony search* yang telah dituliskan pada bab 2.15.

Berikut alur dari algoritma *harmony search* yang direpresentasikan pada Gambar 3.3.

**Gambar 3.3** Alur *Harmony Search*

3.5.5.1 Inisialisasi Parameter

Pada tahap awal algoritma *harmony search* adalah tahap inisialisasi parameter. Inisialisasi parameter yang digunakan pada algoritma *harmony search* adalah *HMCR* (*Harmony Memory Consideration Rate*), *HMS* (*Harmony Memory size*), *PAR* (*Pitch Adjustment Rate*), dan *NI* (*Number of Improvisations*) yang telah dijelaskan pada bab 2.5. Setelah tahap inisialisasi parameter, tahapan selanjutnya adalah tahap inisialisasi *harmony* atau membangkitkan kandidat solusi.

3.5.5.2 Inisialisasi *Harmony Memory*

Tahap selanjutnya adalah proses inisialisasi *harmony memory* atau membangkitkan kandidat solusi secara random untuk parameter a , b , c , k yang terdapat pada perumusan *GLCE+HS* yang dituliskan pada bab 2.15. Untuk nilai a terdiri dari rentang nilai (0 - 1,5), nilai b (0 – $\text{globalMean} / 2$), nilai c (0 - 1), dan k (0,5 – 1,5). Berikut adalah contoh perhitungan inisialisasi *harmony* dengan menggunakan inisialisasi parameter *HMCR* (*Frekuensi Memory Consideration*) = 0.9, *NI* (*Nilai Iterasi*) = 5, *HMS* (*Kandidat Solusi*) = 2, *PAR* (*Frekuensi Pitch Adjustment*) = 0.6 serta banyak variabel = 4 (a, b, c, k) dan perumusan pembentukan kandidat awal yang telah dituliskan pada bab 2.5.

Kandidat ke 1.

$$x(1) = 0 + (1,5 - 0) * 0,8 = 1,2 \quad \rightarrow \text{Untuk nilai parameter } a$$

$$x(2) = 0 + (0,4 - 0) * 0,3 = 0,12 \quad \rightarrow \text{Untuk nilai parameter } b$$

$$x(3) = 0 + (1 - 0) * 0,6 = 0,6 \quad \rightarrow \text{Untuk nilai parameter } c$$

$$x(4) = 0,5 + (1,5 - 0,5) * 0,7 = 1,2 \quad \rightarrow \text{Untuk nilai parameter } k$$

Kandidat ke 2.

$$x(1) = 0 + (1,5 - 0) * 0,124 = 0,186 \rightarrow \text{Untuk nilai parameter } a$$

$$x(2) = 0 + (0,4 - 0) * 0,6271 = 0,25 \rightarrow \text{Untuk nilai parameter } b$$

$$x(3) = 0 + (1 - 0) * 0,6 = 0,95 \rightarrow \text{Untuk nilai parameter } c$$

$$x(4) = 0,5 + (1,5 - 0,5) * 0,076 = 0,576 \rightarrow \text{Untuk nilai parameter } k$$

3.5.5.3 Hitung *Fitness*

Langkah selanjutnya adalah menghitung nilai *fitness*. Nilai *fitness* pada tugas akhir ini berfungsi untuk menentukan kandidat terbaik, semakin besar nilai yang diperoleh maka semakin banyak *edge* yang terdeteksi begitu juga sebaliknya. Perhitungan nilai *fitnes* didapatkan dengan mengkombinasikan nilai *entropy*, jumlah intensitas, serta informasi *edge* dari sebuah citra. Perumusan untuk mendapatkan nilai *fitnes* dapat dilihat pada bab 2.15.

A Pembentukan Matriks Kandidat Solusi

Pembentukan matriks kandidat solusi adalah langkah pertama yang dilakukan untuk mendapatkan nilai *fitness* dari setiap kandidat parameter. Berikut merupakan perhitungan matriks kandidat solusi berdasarkan penjelasan pada bab 2.15.

GLCE berbasis *harmony search* kandidat ke 1

$$g(1,1) = \left(\frac{1,2*2,48}{1,118034+0,12} \right) * ((4 - 0,6 * 2) + 2^{1,2}) = 12,2532$$

$$g(1,2) = \left(\frac{1,2*2,48}{1,369306+0,12} \right) * ((1 - 0,6 * 2) + 2^{1,2}) = 4,19112$$

$$g(1,3) = \left(\frac{1,2*2,48}{1,118034+0,12} \right) * ((3 - 0,6 * 2) + 2^{1,2}) = 9,84937$$

$$g(2,1) = \left(\frac{1,2*2,48}{1,172604+0,12} \right) * ((1 - 0,6 * 2) + 2^{1,2}) = 4,82889$$

$$g(2,2) = \left(\frac{1,2*2,48}{1,457738+0,12} \right) * ((0 - 0,6 * 2) + 2^{1,2}) = 2,06996$$

$$g(2,3) = \left(\frac{1,2*2,48}{1,172604+0,12} \right) * ((4 - 0,6 * 2) + 2^{1,2}) = 11,7359$$

$$g(3,1) = \left(\frac{1,2*2,48}{0,866025+0,12} \right) * ((3 - 0,6 * 2) + 2^{1,2}) = 12,3667$$

$$g(3,2) = \left(\frac{1,2*2,48}{1,172604+0,12} \right) * ((3 - 0,6 * 3) + 3^{1,2}) = 11,3670$$

$$g(3,3) = \left(\frac{1,2*2,48}{1,06066+0,12} \right) * ((2 - 0,6 * 3) + 3^{1,2}) = 9,92418$$

GLCE berbasis *harmony search* kandidat ke 2

$$g(1,1) = \left(\frac{0,576*2,48}{1,118034+0,25} \right) * ((4 - 0,95 * 2) + 2^{0,186}) = 3,38066$$

$$g(1,2) = \left(\frac{0,576*2,48}{1,369306+0,25} \right) * ((1 - 0,95 * 2) + 2^{0,186}) = 0,20961$$

$$g(1,3) = \left(\frac{0,576*2,48}{1,118034+0,25} \right) * ((3 - 0,95 * 2) + 2^{0,186}) = 2,33647$$

$$g(2,1) = \left(\frac{0,576*2,48}{1,172604+0,25} \right) * ((1 - 0,95 * 2) + 2^{0,186}) = 0,23859$$

$$g(2,2) = \left(\frac{0,576*2,48}{1,457738+0,25} \right) * ((0 - 0,95 * 2) + 2^{0,186}) = -0,6377$$

$$g(2,3) = \left(\frac{0,576*2,48}{1,172604+0,25} \right) * ((4 - 0,95 * 2) + 2^{0,186}) = 3,25098$$

$$g(3,1) = \left(\frac{0,576*2,48}{0,866025+0,25} \right) * ((3 - 0,95 * 2) + 2^{0,186}) = 2,86407$$

$$g(3,2) = \left(\frac{0,576*2,48}{1,172604+0,25} \right) * ((3 - 0,95 * 3) + 3^{0,186}) = 1,38240$$

$$g(3,3) = \left(\frac{0,576*2,48}{1,06066+0,25} \right) * ((2 - 0,95 * 3) + 3^{0,186}) = 0,41058$$

Sehingga didapatkan dua kandidat matriks *GLCE* berbasis *harmony search* yang direpresentasikan pada Tabel 3.5 untuk kandidat 1 dan Tabel 3.6 kandidat 2.

Tabel 3.5 Kandidat 1

4	4	2	0	4
2	12	4	10	1
3	5	2	12	3
0	12	11	9	3
0	4	4	3	4

Tabel 3.6 Kandidat 2

4	4	2	0	4
2	3	0	2	1
3	0	0	3	3
0	3	1	0	3
0	4	4	3	4

B Perhitungan *Entropy*

Langkah selanjutnya adalah perhitungan nilai *entropy*. Hal ini dimaksudkan untuk mengetahui pemerataan histogram dari citra. Semakin besar maka semakin baik pemerataan histogramnya. Dimana proses perhitungan *entropy* dari matriks *GLCE* berbasis *harmony search kandidat 1* berdasarkan perumusan yang dijelaskan pada bab 2.13 direpresentasikan pada Tabel 3.7.

Tabel 3.7 Intensitas Piksel kandidat 1

Value	Banyak	Probabilitas
0	3	$3/(5*5)$
1	1	$1/(5*5)$
2	3	$3/(5*5)$
3	4	$4/(5*5)$
4	7	$7/(5*5)$
5	1	$1/(5*5)$
9	1	$1/(5*5)$
10	1	$1/(5*5)$
11	1	$1/(5*5)$

Value	Banyak	Probabilitas
12	3	$3/(5*5)$
Entropy		2,96721

Sedangkan proses perhitungan *entropy* dari matriks *GLCE* berbasis *harmony search* kandidat 2 berdasarkan perumusan yang dijelaskan pada bab 2.13 direpresentasikan pada Tabel 3.8.

Tabel 3.8 Intensitas Piksel kandidat 2

Value	Banyak	Probabilitas
0	7	$7/(5*5)$
1	2	$2/(5*5)$
2	3	$3/(5*5)$
3	7	$7/(5*5)$
4	6	$6/(5*5)$
Entropy		2,18115

C Perhitungan Sobel

Pada tahap ini merupakan pembentukan matriks *sobel*. Tujuan dari proses ini adalah untuk mendapatkan informasi *edge* dari citra. Berikut merupakan perhitungan *sobel* dari Tabel 3.5 dan Tabel 3.6 dengan perumusan yang telah dituliskan pada Bab 2.6.

GLCE berbasis *harmony search* kandidat 1

$$\begin{aligned} sobel(1,1) &= \sqrt{(3 + (2 * 5) + 2 - 4 - (2 * 4) - 2)^2 + (2 + (2 * 4) + 2 - 4 - (2 * 2) - 3)^2} \\ &= 1,414214 \end{aligned}$$

$$\begin{aligned} sobel(1,2) &= \sqrt{(5 + (2 * 2) + 12 - 4 - (2 * 2) - 0)^2 + (0 + (2 * 10) + 12 - 4 - (2 * 12) - 5)^2} \\ &= 13,0384 \end{aligned}$$

$$\begin{aligned} sobel(1,3) &= \sqrt{(2 + (2 * 12) + 3 - 2 - (2 * 0) - 4)^2 + (4 + (2 * 1) + 3 - 2 - (2 * 4) - 2)^2} \\ &= 23,19483 \end{aligned}$$

$$\begin{aligned} sobel(2,1) &= \sqrt{(0 + (2 * 12) + 11 - 2 - (2 * 12) - 4)^2 + (4 + (2 * 2) + 11 - 2 - (2 * 3) - 0)^2} \\ &= 12,08305 \end{aligned}$$

$$\begin{aligned} sobel(2,2) &= \sqrt{(12 + (2 * 11) + 9 - 12 - (2 * 4) - 10)^2 + (10 + (2 * 12) + 9 - 12 - (2 * 5) - 12)^2} \\ &= 15,81139 \end{aligned}$$

$$\begin{aligned} sobel(2,3) &= \sqrt{(11 + (2 * 9) + 3 - 4 - (2 * 10) - 1)^2 + (1 + (2 * 3) + 3 - 4 - (2 * 2) - 11)^2} \\ &= 11,40175 \end{aligned}$$

$$\begin{aligned} sobel(3,1) &= \sqrt{(0 + (2 * 4) + 4 - 3 - (2 * 5) - 2)^2 + (2 + (2 * 11) + 4 - 3 - (2 * 0) - 0)^2} \\ &= 25,17936 \end{aligned}$$

$$\begin{aligned} sobel(3,2) &= \sqrt{(4 + (2 * 4) + 3 - 5 - (2 * 2) - 12)^2 + (12 + (2 * 9) + 3 - 5 - (2 * 12) - 4)^2} \\ &= 6 \end{aligned}$$

$$\begin{aligned} sobel(3,3) &= \sqrt{(4 + (2 * 3) + 4 - 2 - (2 * 12) - 3)^2 + (3 + (2 * 3) + 4 - 2 - (2 * 11) - 4)^2} \\ &= 21,2132 \end{aligned}$$

GLCE berbasis *harmony search* kandidat 2

$$\begin{aligned} sobel(1,1) &= \sqrt{(3 + (2 * 0) + 0 - 4 - (2 * 2) - 0)^2 + (2 + (2 * 0) + 0 - 4 - (2 * 2) - 3)^2} \\ &= 10,29563 \end{aligned}$$

$$\begin{aligned} sobel(1,2) &= \sqrt{(0 + (2 * 0) + 3 - 4 - (2 * 0) - 4)^2 + (0 + (2 * 2) + 3 - 4 - (2 * 3) - 0)^2} \\ &= 5,830952 \end{aligned}$$

$$\begin{aligned} sobel(1,3) &= \sqrt{(0 + (2 * 3) + 3 - 2 - (2 * 4) - 0)^2 + (4 + (2 * 1) + 3 - 2 - (2 * 0) - 3)^2} \\ &= 4,123106 \end{aligned}$$

$$sobel(2,1) = \sqrt{(0 + (2 * 3) + 1 - 2 - (2 * 3) - 0)^2 + (0 + (2 * 0) + 1 - 2 - (2 * 3) - 0)^2}$$

$$= 7,071068$$

$$sobel(2,2) = \sqrt{(3 + (2 * 1) + 0 - 3 - (2 * 0) - 2)^2 + (2 + (2 * 3) + 0 - 3 - (2 * 0) - 3)^2}$$

$$= 2$$

$$sobel(2,3) = \sqrt{(1 + (2 * 0) + 3 - 0 - (2 * 2) - 1)^2 + (1 + (2 * 3) + 3 - 0 - (2 * 0) - 1)^2}$$

$$= 9,055385$$

$$sobel(3,1) = \sqrt{(0 + (2 * 4) + 4 - 3 - (2 * 0) - 0)^2 + (0 + (2 * 1) + 4 - 3 - (2 * 0) - 0)^2}$$

$$= 9,486833$$

$$sobel(3,2) = \sqrt{(4 + (2 * 4) + 3 - 0 - (2 * 0) - 3)^2 + (3 + (2 * 0) + 3 - 0 - (2 * 3) - 4)^2}$$

$$= 12,64911$$

$$sobel(3,3) = \sqrt{(4 + (2 * 3) + 4 - 0 - (2 * 3) - 3)^2 + (3 + (2 * 3) + 4 - 0 - (2 * 1) - 4)^2}$$

$$= 8,602325$$

Sehingga matriks *sobel* untuk kandidat 1 ditunjukkan oleh Tabel 3.9 dan matriks *sobel* untuk kandidat 2 ditunjukkan oleh Tabel 3.10.

Tabel 3.9 *sobel kandidat 1*

4	4	2	0	4
2	1	13	23	1
3	12	16	11	3
0	25	6	21	3
0	4	4	3	4

Tabel 3.10 *sobel Kandidat 2*

4	4	2	0	4
2	10	6	4	1
3	7	2	9	3
0	9	13	9	3
0	4	4	3	4

Setelah mendapatkan matriks sobel dari tiap kandidat langkah berikutnya adalah melakukan perhitungan threshold sobel. *Threshold* sobel didapatkan dari perhitungan nilai tiap piksel pada matriks dibagi dengan nilai maksimum dari

matriks. Matriks *threshold sobel* untuk kandidat 1 ditunjukkan oleh Tabel 3.11 dan matriks *threshold sobel* untuk kandidat 2 ditunjukkan oleh Tabel 3.12.

Tabel 3.11 *sobel kandidat TH1*

0	0	0	0	0
0	0	1	1	0
0	0	1	0	0
0	1	0	1	0
0	0	0	0	0

Tabel 3.12 *sobel Kandidat TH2*

0	0	0	0	0
0	1	0	0	0
0	1	0	1	0
0	1	1	1	0
0	0	0	0	0

Sehingga didapatkan n_{edges} dari Tabel 3.11 adalah 4 dan n_{edges} dari Tabel 3.12 adalah 7.

Jumlah intensitas sobel didapatkan dari penjumlahan keseluruhan intensitas piksel yang dihasilkan dari proses *operator sobel*. Berikut adalah perhitungan *Jumlah intensitas sobel* yang telah dituliskan pada bab 2.6.

$$E(I_{s \text{ kandidat } 1}) = (1 + 13 + 23 + 12 + 16 + 11 + 25 + 6 + 21) = 128$$

$$E(I_{s \text{ kandidat } 2}) = (10 + 6 + 4 + 7 + 2 + 9 + 9 + 13 + 9) = 69$$

D Nilai Fitness Kandidat Solusi

Setelah mendapatkan informasi *entropy*, *jumlah intensitas sobel*, dan *banyak garis*, langkah selanjutnya adalah perhitungan nilai fitness. Berikut merupakan perhitungan manual nilai fitness berdasarkan bab 2.15.

$$F(I_{e \text{ kandidat } 1}) = \log(\log(128)) * \frac{5}{5 * 5} * 2,96721 = 0,192102$$

$$F(I_{e \text{ kandidat } 2}) = \log(\log(69)) * \frac{6}{5 * 5} * 2,18115 = 0,138484$$

3.5.5.4 Improvisasi *Harmony Memory*

Setelah mendapatkan nilai *fitness* dari kandidat solusi langkah selanjutnya adalah dengan improvisasi *harmony memory*. Tujuan dari tahap ini adalah untuk mencari kombinasi parameter dari kandidat parameter yang telah ada untuk mencari kandidat yang terbaik.

3.5.5.5 Update *Harmony Memory*

Langkah selanjutnya adalah update *harmony memory*. Maksudanya adalah mengganti nilai kandidat a , b , c , k yang mempunyai nilai *fitnes* paling rendah dengan kandidat baru dari proses improvisasi *harmony memory* yang memiliki nilai *fitnes* lebih baik. Apabila kandidat baru mempunyai nilai *fitnes* lebih jelek maka tidak terjadi perubahan pada kandidat. Proses ini diulang terus hingga nilai iterasi (NI) terpenuhi

3.5.6 GLCE berbasis *Harmony Search*

Langkah selanjutnya adalah memasukan nilai parameter (a , b , c , k) yang mempunyai nilai *fitnes* terbaik kedalam perumusan $GLCE+HS$.

3.6 Skenario Pengujian

Pengujian yang dilakukan pada penelitian ini dilakukan dengan dua tahap yaitu pengujian hasil *image enhancement* dan pengujian metode *harmony search*. Pengujian metode *harmony search* dilakukan untuk mengetahui akurasi paramater yang dihasilkan, dengan cara melakukan kombinasi parameter secara manual dengan ketentuan nilai parameter terbaik diambil dari nilai *fitnes* yang terbesar. pengujian hasil *image enhancement* dilakukan menggunakan metode *AMBE*, *PSNR*, dan *fitnes*.

3.6.1 AMBE

Pada tahap pengujian ini dimaksudkan dengan tujuan untuk menguji citra keluaran perumusan *GLCE* berbasis *harmony search* dalam segi mempertahankan pencahayaan *edge*. Semakin besar nilai *AMBE* maka semakin tajam *edge* hasil keluaran citra dengan menggunakan *GLCE* berbasis *harmony search*. Sedangkan semakin kecil nilai *AMBE* maka semakin jelek hasil keluaran citra, sebab semakin kecil nilai *AMBE* maka hasil keluaran citra semakin mendekati citra originalnya. Pada citra original retinopati diabetes memiliki kelemahan yaitu bernoise, dan memiliki kontras yang rendah yang mengakibatkan ada beberapa *edge* yang terlihat kurang jelas.

3.6.2 PSNR

Pada tahap pengujian ini dimaksudkan dengan tujuan untuk menguji citra keluaran perumusan *GLCE* berbasis *harmony search* dalam segi mereduksi noise, serta peningkatan kontras. Semakin kecil nilai *PSNR* maka semakin baik hasil keluaran citra. Sedangkan semakin besar nilai *PSNR* maka semakin jelek hasil keluaran citra, sebab semakin besar nilai *PSNR* maka hasil keluaran citra semakin mendekati citra originalnya. Pada citra original retinopati diabetes memiliki kelemahan yaitu bernoise, dan memiliki kontras yang rendah.

3.6.3 Nilai *Fitness*

Pada tahap pengujian ini dimaksudkan dengan tujuan untuk menguji citra keluaran perumusan *GLCE* berbasis *harmony search* dalam segi pemeretaan kontras citra pada *edge* dari citra. Semakin besar nilai *fitnes* maka semakin baik

hasil pemerataan intensitas histogramnya sehingga *edge* dari citra akan terlihat lebih baik.