

BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa

Analisa merupakan tahapan yang tidak pernah terlepas dari rancang bangun perangkat lunak. Tahapan analisa digunakan untuk menganalisa masalah yang berhubungan dengan pembangunan sistem. Analisa didalam penelitian ini terdiri dari 2 analisa, yaitu analisa kebutuhan data analisa proses temu kembali gambar.

4.2 Analisa Kebutuhan Data

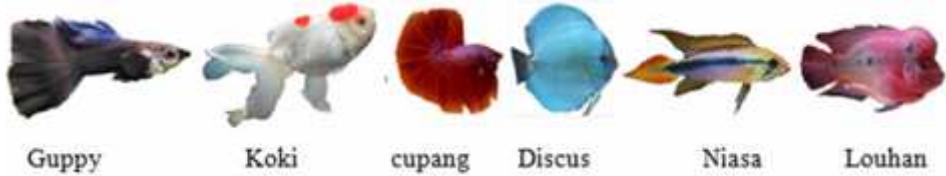
Analisa kebutuhan data merupakan tahapan dimana dilakukan analisa terhadap data yang digunakan didalam sistem. Data yang dianalisa didalam tugas akhir ini adalah data citra ikan hias yang dikelompokkan menjadi 3 data, yaitu data latih, data kueri dan data keluaran.

1. Data Latih

Data latih merupakan data yang digunakan sebagai korpus atau *database*. Citra data latih merupakan citra yang memiliki kriteria sebagai berikut.

Citra data latih yang dijadikan sebagai korpus atau *database* adalah citra ikan hias. Data citra ikan hias yang digunakan adalah 6 jenis ikan hias, yaitu : ikan Cupang, ikan Discus, ikan koki, ikan Louhan, ikan guppy dan jenis ikan hias Niasa. Data latih diperoleh dengan menggunakan kamera dan dari internet dengan alamat www.photobucket.com.

Citra data latih berjumlah 48 citra dari citra 6 jenis ikan hias dengan posisi citra berdasarkan sudut yang berbeda-beda. Data ikan hias diambil secara acak dari 450 spesies ikan hias air tawar berdasarkan perbedaan warna, bentuk dan tekstur. Selain berdasarkan perbedaan warna, dan bentuk 6 jenis ikan hias ini digunakan dalam tugas akhir karena 6 jenis ikan hias ini merupakan jenis ikan yang banyak digemari oleh pecinta ikan hias karena keindahan dan kepopulerannya (www.kkp.co.id). Gambar 6 jenis ikan hias yang digunakan dalam tugas akhir dapat dilihat pada gambar 4.1



Gambar 4.1 Citra 6 Ikan Hias

Format citra yang digunakan dalam tugas akhir adalah format JPG, hal ini untuk kemudahan dalam penelitian ini, karena data standar yang ada dalam sistem CBIR saat ini dalam format JPG.

Latar belakang (*background*) citra *database* dan citra kueri adalah putih. Hal ini bertujuan untuk menghindari kerancuan dalam deteksi tepi objek ikan saat dilakukan proses indeks sehingga mempermudah proses selanjutnya dalam penelitian ini. Pengubahan latar belakang pada citra menggunakan tools, tools yang digunakan adalah photoshop.

Ukuran citra yang digunakan didalam tugas akhir adalah 300 X 300 *pixel*. Hal ini bertujuan untuk mempercepat perhitungan dalam pemrosesan data dalam penelitian ini ketika diimplementasikan. Proses *resize* pada citra dilakukan dengan menggunakan tools photoshop.

2. Data Uji

Data uji merupakan data yang digunakan sebagai data kueri. Data dari sistem ini adalah citra 6 jenis ikan hias yang sama dengan gambar data latih. Data citra uji terdiri dari citra dengan posisi citra horizontal dan vertikal. Kriteria citra yang digunakan sebagai data uji sama dengan kriteria data latih.

3. Data keluaran

Data keluaran merupakan data hasil temu kembali gambar ikan hias. Data keluaran dianalisa menggunakan MSE, *precision* dan *recall*.

4.3 Analisa Proses Temu Kembali Citra

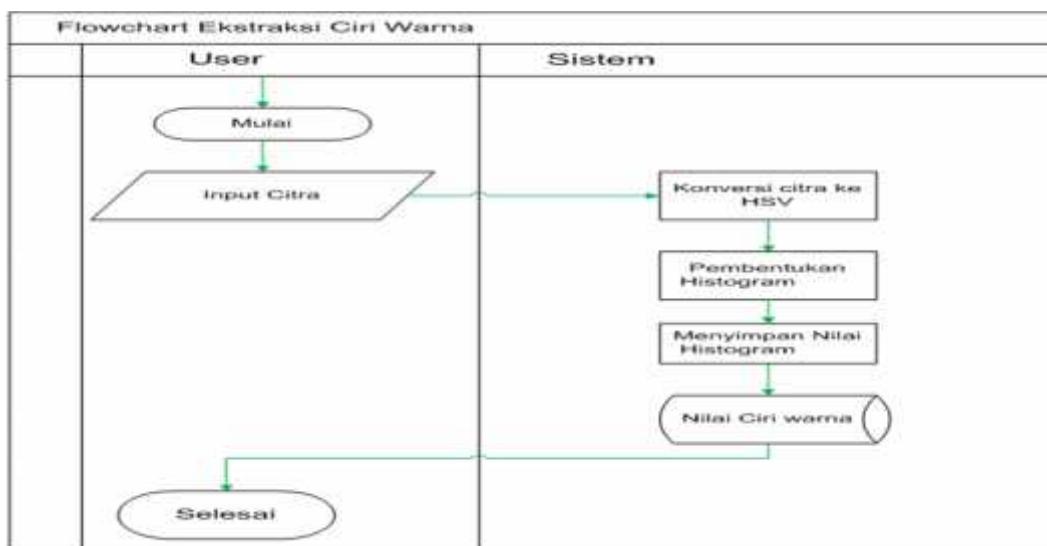
Analisa proses temu kembali gambar merupakan tahapan inti didalam tugas akhir. Tahapan analisa proses temu kembali citra merupakan analisa terhadap proses-proses yang dilakukan dalam pembangunan sistem. Tahapan analisa proses temu kembali citra terdiri dari ekstraksi ciri, indeks dan pengukuran kemiripan.

4.3.1 Analisa Ekstraksi Ciri

Ekstraksi ciri merupakan proses yang dilakukan untuk mendapatkan nilai ciri pada citra. Nilai ciri citra yang digunakan didalam sistem untuk mengenali citra didalam sistem berdasarkan nilai ciri warna, bentuk dan tekstur. Nilai ciri warna digunakan untuk mengenali warna citra ikan hias, nilai ciri bentuk digunakan untuk mengenali bentuk citra ikan hias, dan nilai tekstur digunakan untuk mengenali tekstur citra ikan hias.

4.3.1.1 Ekstraksi Ciri Warna

Ekstraksi ciri warna merupakan tahapan untuk mendapatkan nilai warna. Nilai warna didalam proses temu kembali gambar digunakan untuk mengenali warna dari citra ikan hias. Representasi warna didalam sistem direpresentasikan dengan nilai-nilai yang diperoleh dengan menggunakan pembentukan histogram. Histogram yang dibentuk didalam sistem ini adalah histogram dari model warna HSV. Flowchart untuk memperoleh nilai warna dengan menggunakan histogram dari model warna HSV dapat dilihat pada gambar 4.2.

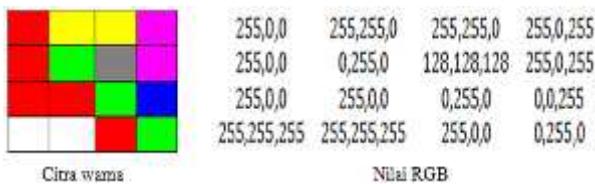


Gambar 4.2 Ekstraksi Ciri Warna

Berdasarkan gambar 4.2 flowchart ekstraksi ciri warna untuk mendapatkan nilai warna ikan hias langkah-langkahnya sebagai berikut.

1. Konversi citra RGB ke HSV

Ekstraksi ciri warna menggunakan citra model wara HSV dimulai dengan melakukan konversi ctra dari citra RGB menjadi citra HSV. Contoh perhitungan konversi citra dari RGB menjadi HSV sebagai berikut.



Gambar 4.3 Citra warna dan Nilai RGB

Contoh perhitungan normalisasi nilai RGB berdasarkan persamaan 2.4, 2.5 dan 2.6 dapat dilihat pada tabel 4.1.

Tabel 4.1 Normalisasi Citra RGB

Piksel Ke-	Nilai RGB Citra Asli			Nilai Citra RGB Normalisasi		
	R	G	B	R	G	B
1	255	0	0	(255/(0+255+0)) = 1	(0/(0+255+0)) = 0	(0/(0+255+0)) = 0
2	255	255	0	(255/(255+255+0))=0.50	(0/(255+255+0)) = 0	(255/(255+255+0)) = 0.50
3	255	255	0	(255/(255+255+0))= 0.50	(255/(255+255+0))=0.50	(0/(255+255+0)) = 0
4	255	0	255	(255/(255+0+255))= 0.50	(0/(255+0+255)) = 0	(255/(255+0+255))= 0.50
5	255	0	0	(255/(255+0+0)) = 1	(0/(255+0+0)) = 0	(0/(255+0+0)) = 0
6	0	255	0	(0/(0+255+0)) = 0	(255/(0+255+0)) = 1	(0/(0+255+0)) = 0
7	128	128	128	(128/(128+128+128)) = 0.33	(128/(128+128+128)) = 0.33	(128/(128+128+128)) = 0.33
8	255	0	255	(255/(255+0+255))= 0.50	(0/(255+0+255)) = 0	(255/(255+0+255))= 0.50
9	255	0	0	(255/(255+0+0)) = 1	(0/(255+0+0)) = 0	(0/(255+0+0)) = 0
10	255	0	0	(255/(255+0+0)) = 1	(0/(255+0+0)) = 0	(0/(255+0+0)) = 0
11	0	255	0	(0/(0+255+0)) = 0	(255/(0+255+0)) = 1	(0/(0+255+0)) = 0
12	0	0	255	(0/(0+0+255)) = 0	(0/(0+0+255)) = 0	(255/(0+0+255)) = 1
13	255	255	255	(255/(255+255+255))=0.33	(255/(255+255+255))=0.33	(255/(255+255+255))=0.33
14	255	255	255	(255/(255+255+255))= 0.33	(255/(255+255+255))= 0.33	(255/(255+255+255))= 0.33
15	255	0	0	(255/(255+0+0)) = 1	(0/(255+0+0)) = 0	(0/(255+0+0)) = 0
16	0	255	0	(0/(0+255+0)) = 0	(255/(0+255+0)) = 1	(0/(0+255+0)) = 0

1,0,0	0.50,0.50,0	0.50,0.50,0	0.50,0.50,0
1,0,0	0,1,0	0.33,0.33,0.33	0.50,0,0.50
1,0,0	1,0,0	0,1,0	0,0,1
0.33,0.33,0.33	0.33,0.33,0.33	1,0,0	0,1,0

Gambar 4.4 Matrik RGB Hasil Normalisasi

Konversi citra RGB menjadi HSV menggunakan persamaan 2.7, 2.8 dan 2.9 dapat dilihat pada tabel 4.2.

Tabel 4.2 Konversi RGB Menjadi Citra HSV

Piksel Ke	Nilai Rgb Normalisasi			Nilai Hsv		
	R	G	B	H	S	V
1	1	0	0	$60 \times (0 - 0) = 0$	$1 - \frac{0}{1} = 1$	1 (r)
2	0.50	0	0.50	$60 \times (0.5 - 0) = 30$ $-30 + 360 = 330$	$0.5 - \frac{0}{0.5} = 0.5$	0.5 (r)
3	0.50	0	0.50	$60 \times (0.5 - 0) = 30$ $-30 + 360 = 330$	$0.5 - \frac{0}{0.5} = 0.5$	0.5 (r)
4	0.50	0	0.50	$60 \times (0 - 0.5) = -30$ $-30 + 360 = 330$	$0.5 - \frac{0}{0.5} = 0.5$	0.5 (r)
5	1	0	0	$60 \times (0 - 0) = 0$	$1 - \frac{0}{1} = 1$	1 (r)
6	0	1	0	$60 \times (1 - 0) = 60$ $60 \times [1 - 1] = 0$	$1 - \frac{0}{1} = 1$	1 (g)
7	0.33	0.33	0.33	0	0	0
8	0.50	0	0.50	$60 \times (0 - 0.5) = -30$ $-30 + 360 = 330$	$0.5 - \frac{0}{0.5} = 0.5$	0.5 (r)
9	1	0	0	$60 \times (0 - 0) = 0$	$1 - \frac{0}{1} = 1$	1(r)
10	1	0	0	$60 \times (1 - 0) = 60$ $60 \times [1 - 1] = 0$	$1 - \frac{0}{1} = 1$	1(r)
11	0	1	0	$60 \times (1 - 0) = 60$ $60 \times [1 - 1] = 0$	$1 - \frac{0}{1} = 1$	1 (g)
12	0	0	1	$60 \times (1 - 1) = 0$	$1 - \frac{0}{1} = 1$	1 (b)
13	0.33	0.33	0.33	0	0	0
14	0.33	0.33	0.33	0	0	0
15	1	0	0	$60 \times (1 - 0) = 60$ $= 60$	$1 - \frac{0}{1} = 1$	1(r)
16	0	1	0	$60 \times (1 - 0) = 60$ $60 \times [1 - 1] = 0$	$1 - \frac{0}{1} = 1$	1 (g)

0,1,1	330,0,50,0,50	330,0,50,0,50	330,0,50,0
0,1,1	120,1,1	0,0,0	330,0,50,0,50
0,1,1	0,1,1	120,1,1	240,1,1
0,0,0	0,0,0	0,1,1	120,1,2

Gambar 4.5 Matrik Hasil Konversi Ke Citra HSV

Konversi citra RGB ke HSV pada matlab bisa menggunakan fungsi ‘rgb2HSV’. Contohnya seperti berikut :

```
G=imread('gambar1.jpg');
```

```
H=rgb2HSV(G);
```

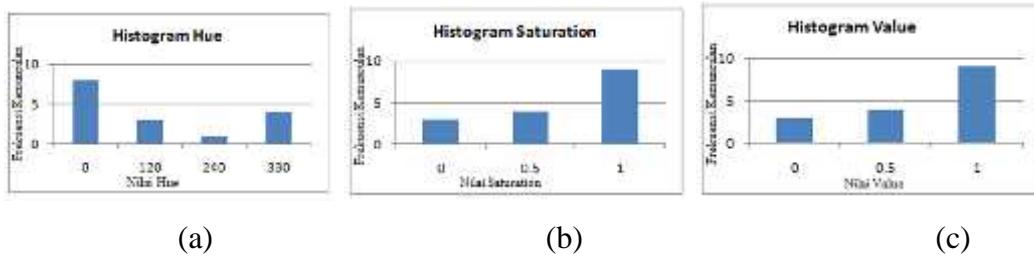
Maka gambar ‘gambar1.jpg’ akan berubah ke bentuk hsv untuk di proses pada pembuatan histogram model warna HSV.

- Pembuatan histogram dilakukan pada citra yang sudah di konversi ke model HSV. Contoh pembentukan histogram model warna HSV sebagai berikut.

0,1,1	330,0,50,0,50	330,0,50,0,50	330,0,50,0
0,1,1	120,1,1	0,0,0	330,0,50,0,50
0,1,1	0,1,1	120,1,1	240,1,1
0,0,0	0,0,0	0,1,1	120,1,2

Gambar 4.6 Nilai Matrik Citra HSV

Histogram model warna HSV yang terbentuk berdasarkan citra yang dikonversi menjadi HSV sebagai berikut.



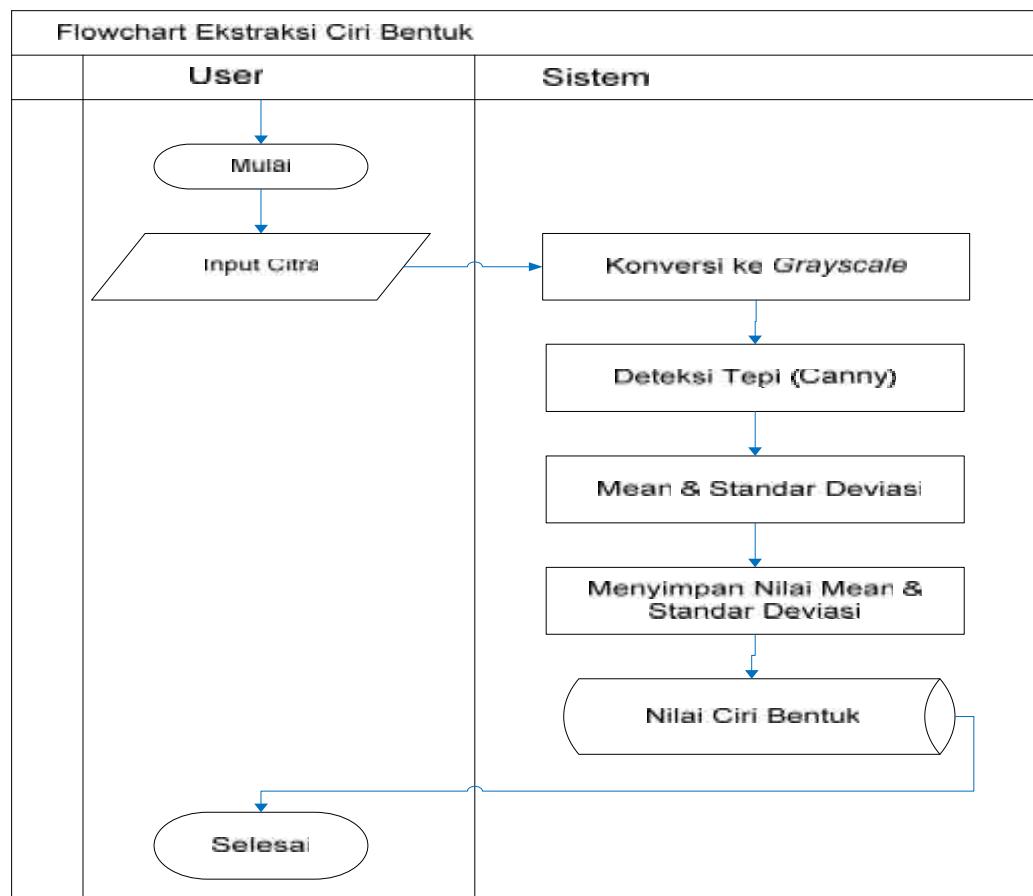
Gambar 4.7 (a) Histogram Hue (b) Histogram Saturation (c) Histogram Value

- Tahap selanjutnya setelah diperoleh nilai histogram maka nilai histogram disimpan sebagai nilai ciri warna.

4.3.1.2 Ekstraksi Ciri Bentuk

Ekstraksi ciri bentuk merupakan tahapan untuk mengenali bentuk dari citra ikan hias. Bentuk citra ikan hias direpresentasikan dengan nilai-nilai yang diperoleh dari proses penghitungan menggunakan penghitungan statistik.

Flowchart untuk mengenali bentuk citra ikan hias atau proses ekstraksi ciri bentuk dapat dilihat pada gambar 4.8.



Gambar 4.8 Flowchart Ekstraksi Ciri Bentuk

Berdasarkan gambar 4.8 flowchart ekstraksi ciri bentuk langkah-langkah untuk ekstraksi ciri bentuk adalah sebagai berikut.

1. Konversi citra ke *grayscale*.

Konversi proses pengubahan warna citra menjadi *grayscale*. Pengubahan warna menjadi *grayscale* ini merupakan tahapan *pre-processing*. Konversi citra menjadi citra *grayscale* didalam ekstraksi ciri bentuk dilakukan agar citra dapat dilakukan proses selanjutnya, yaitu proses deteksi tepi.

Proses konversi citra menjadi *grayscale* dilakukan dengan mencari nilai rata-rata dari total nilai warna red,green dan blue atau dengan cara menjumlahkan semua nilai red,green dan blue kemudian total nilai akan dibagi 3 (red,green dan blue). Fungsi matlab yang digunakan untuk mengubah warna menjadi

grayscale adalah fungsi ‘rgb2gray’. Berdasarkan rumus 2.11 Contoh perhitungan konversi citra *rgb* menjadi citra *grayscale* sebagai berikut.

Citra warna	Nilai RGB
Red	255,0,0
Yellow	255,255,0
Green	0,255,0
Grey	128,128,128
Magenta	255,0,255
Cyan	0,255,255
Black	255,255,255
White	255,255,255

Gambar 4.9 Citra RGB dan Nilai RGB

Konversi citra *RGB* menjadi citra *grayscale* menggunakan persamaan 2.11

$$G = \frac{255 + 0 + 0}{3} \quad \frac{255 + 255 + 0}{3} \quad \frac{255 + 255 + 0}{3} \quad \frac{255 + 0 + 255}{3}$$

$$\frac{255 + 0 + 0}{3} \quad \frac{0 + 255 + 0}{3} \quad \frac{128 + 128 + 128}{3} \quad \frac{255 + 0 + 255}{3}$$

$$\frac{255 + 0 + 0}{3} \quad \frac{255 + 0 + 0}{3} \quad \frac{0 + 255 + 0}{3} \quad \frac{0 + 0 + 255}{3}$$

$$\frac{255 + 255 + 255}{3} \quad \frac{255 + 255 + 255}{3} \quad \frac{255 + 0 + 0}{3} \quad \frac{0 + 255 + 0}{3}$$

$$G = \begin{matrix} 85 & 170 & 170 & 170 \\ 85 & 85 & 128 & 170 \\ 85 & 85 & 85 & 255 \\ 255 & 255 & 85 & 85 \end{matrix}$$

Citra Grayscale	Nilai Grayscale
White	85
Grey	85
Black	128
White	170
White	170
White	255
White	255
White	85
White	85

Gambar 4.10 Citra Grayscale Dan Nilai Grayscale

2. Proses deteksi tepi.

Deteksi tepi didalam ekstraksi ciri bentuk merupakan tahapan *processing* setelah konversi citra menjadi citra *grayscale*. Deteksi tepi dilakukan menggunakan operator canny. Proses-proses deteksi tepi menggunakan operator canny sebagai berikut.

a. Penghalusan atau *smoothing* citra.

Penghalusan citra dengan tujuan mengurangi noise yang terdapat pada citra. *Smoothing* dilakukan dengan melakukan konvolusi dengan *mask gaussian filter*.

b. Menghitung nilai gradient.

Nilai gradient diperoleh dengan melakukan perkalian matrik citra dengan salah satu turunan pertama, contohnya operator sobel, mask operator dapat dilihat didalam landasan teori.

c. Menghitung nilai arah tepian

Menghitung nilai arah tepian dilakukan berdasarkan nilai gradient yang diperoleh. Nilai arah tepian dilakukan untuk menemukan nilai garis didalam citra.

d. Menghubungkan garis berdasarkan nilai arah tepian dan gradient.

Proses menghubungkan garis citra yang terbentuk didalam citra berdasarkan nilai arah tepian dan nilai gradient. Proses ni akan menghasilkan citra dengan garis tepian yang tebal atau menumpuk.

e. Proses *Non-maxima suppression*, *Non-maxima suppression* ini dilakukan untuk menghilangkan titik yang tidak maksimum, dimana pada proses sebelumnya masih terdapat tepian yang salah, pada proses ini di hilangkan tepian yang tidak maksimum dan akan menghasilkan garis tepian lebih tipis.

f. Proses pengambangan ganda, pada proses pengambangan ini menggunakan nilai standar matlab. Hal ini karena dalam penelitian ini menggunakan fungsi canny yang disediakan oleh matlab. Contoh menggunakan fungsi canny sebagai berikut.

```
G=imread('gambar1.jpg')
```

```
H=edge(G,'canny')
```

Berdasarkan fungsi matlab tersebut maka gambar1.jpg akan di peroleh sketsa atau tepi dari citra. Contoh perhitungan deteksi tepi canny sebagai berikut.

Smoothing menggunakan operator *gaussian* matrik 3 x 3 berdasarkan persamaan 2.12.

$$g_{\delta} = \begin{matrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 2 & x \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{matrix} \quad \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} = \begin{matrix} 4 & 4 & 3 & 5 & 4 \\ 6 & \textcolor{red}{4} & 0 & 8 & 2 \\ 5 & 0 & 2 & 6 & 2 \\ 6 & 6 & 0 & 2 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{matrix}$$

Perkalian dengan turunan pertama untuk mendapatkan nilai gradient menggunakan operator sobel berdasarkan persamaan 2.13

$$g_x = \begin{array}{ccccccccc} 4 & 4 & 3 & 5 & 4 & & & & \\ 6 & 4 & 0 & 8 & 2 & -1 & -2 & -1 & \\ 5 & 0 & 2 & 6 & 2 & 0 & 0 & 0 & = \\ 6 & 6 & 0 & 2 & 3 & 1 & 2 & 1 & \\ 3 & 5 & 2 & 4 & 4 & & & & \end{array} \begin{array}{ccccccccc} 4 & 4 & 3 & 5 & 4 & & & & \\ 6 & 0 & 0 & 0 & 2 & & & & \\ 5 & 4 & 0 & 0 & 2 & & & & \\ 6 & 8 & 3 & 0 & 3 & & & & \\ 3 & 5 & 2 & 4 & 4 & & & & \end{array}$$

$$g_x = \begin{array}{ccccccccc} 4 & 4 & 3 & 5 & 4 & & & & \\ 6 & 4 & 0 & 8 & 2 & -1 & 0 & 1 & \\ 5 & 0 & 2 & 6 & 2 & -2 & 0 & 2 & = \\ 6 & 6 & 0 & 2 & 3 & -1 & 2 & 1 & \\ 3 & 5 & 2 & 4 & 4 & & & & \end{array} \begin{array}{ccccccccc} 4 & 4 & 3 & 5 & 4 & & & & \\ 6 & 16 & 15 & 5 & 2 & & & & \\ 5 & 0 & 12 & 5 & 2 & & & & \\ 6 & 0 & 0 & 8 & 3 & & & & \\ 3 & 5 & 2 & 4 & 4 & & & & \end{array}$$

$$M = \begin{array}{ccccc} 4 & 4 & 3 & 5 & 4 & & & & \\ 6 & 0 & 0 & 0 & 2 & 4 & 4 & & \\ 5 & 4 & 0 & 0 & 2 & + & 5 & 0 & 12 & 5 & 2 \\ 6 & 8 & 3 & 0 & 3 & & 6 & 0 & 0 & 8 & 3 \\ 3 & 5 & 2 & 4 & 4 & & 3 & 5 & 2 & 4 & 4 & = \\ & & & & & & 8 & 8 & 6 & 10 & 8 \\ & & & & & & 12 & 16 & 15 & 5 & 4 \\ & & & & & & 10 & 4 & 12 & 5 & 4 \\ & & & & & & 12 & 8 & 3 & 8 & 6 \\ & & & & & & 6 & 10 & 4 & 8 & 8 \end{array}$$

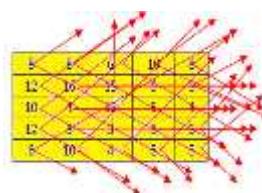
Setelah nilai gradient diperoleh maka nilai arah tepian dapat dihitung.

Proses mendapatkan nilai arah tepian menggunakan persamaan 2.14.

$$\frac{g_y}{g} = \begin{array}{ccccc} 4 & 4 & 3 & 5 & 4 \\ 6 & 16 & 15 & 5 & 2 \\ 5 & 0 & 12 & 5 & 2 \\ 6 & 0 & 0 & 8 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{array} = \begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 4 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{array}$$

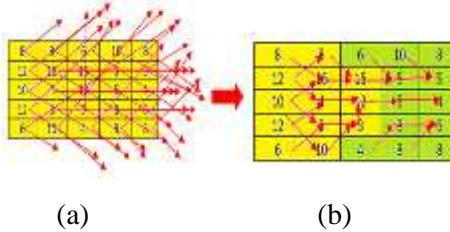
$$\theta(x, y) = \tan^{-1} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 4 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{array}{ccccc} 45 & 45 & 45 & 45 & 45 \\ 45 & 0 & 0 & 0 & 45 \\ 45 & 0 & 0 & 0 & 45 \\ 45 & 0 & 75,9637 & 0 & 45 \\ 45 & 45 & 45 & 45 & 45 \end{array}$$

Ilustrasi pembentukan garis berdasarkan nilai tepian arah gradient seperti pada gambar 4.11



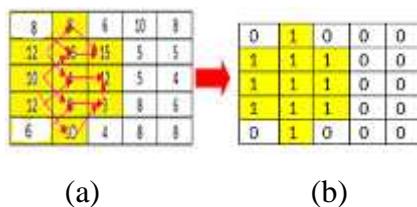
Gambar 4.11 Garis Berdasarkan Arah Tepian Gradient

Ilustrasi *non-maxima suppression* seperti pada gambar 4.12



Gambar 4.12 (a) Garis Berdasarkan Arah Tepian (b) Garis Berdasarkan Arah Tepian Gradien Dan Nilai Gradient

Ilustrasi penghilangan garis yang terputus dan memberi nilai 1 tepian dan 0 bukan tepian.



Gambar 4.13 (a) Garis Berdasarkan Arah Tepian Gradien Dan Nilai Gradient (b) Citra Dengan Tepian Berdasarkan Threshold

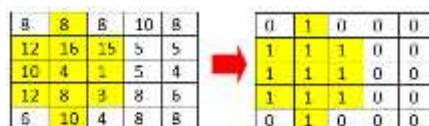
Citra hasil akhir dari deteksi tepi canny.

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

Gambar 4.14 Citra Biner Hasil Deteksi Tepi Canny

- Menghitung nilai mean dan standar deviasi.

Tahapan penghitungan mean dan standar deviasi ini merupakan tahapan untuk mendapatkan nilai ciri untuk ekstraksi ciri bentuk. Langkah-langkah menghitung nilai mean dan standar deviasi sebagai berikut.



Gambar 4.15 Nilai Gradient Dan Citra Biner Hasil Deteksi Tepi Canny

Nilai M (gradient) diperoleh dari proses sebelumnya yaitu deteksi tepi.

Diketahui :

$$z_i = 8, 8, 12, 16, 15, 10, 4, 12, 12, 8, 3, 6, 10, 4$$

$$n = 14$$

Tabel 4.3 Nilai Peluang Kemunculan Berdasarkan Nilai Gradient

P(0)=0/14=0	P(9)=0/14=0
P(1)=0/14=0	P(10)=2/14= 0,1428
P(2)=0/14=0	P(11)=0/14=0
P(3)=1/14= 0,2500	P(12)=3/14= 0,2142
P(4)=2/14= 0,1428	P(13)=0/14=0
P(5)=0/14=0	P(14)=0/14=0
P(6)=1/14= 0,2500	P(15)=1/14= 0,2500
P(7)=0/14=0	P(16)=1/14= 0,2500
P(8)=3/14= 0,2142	

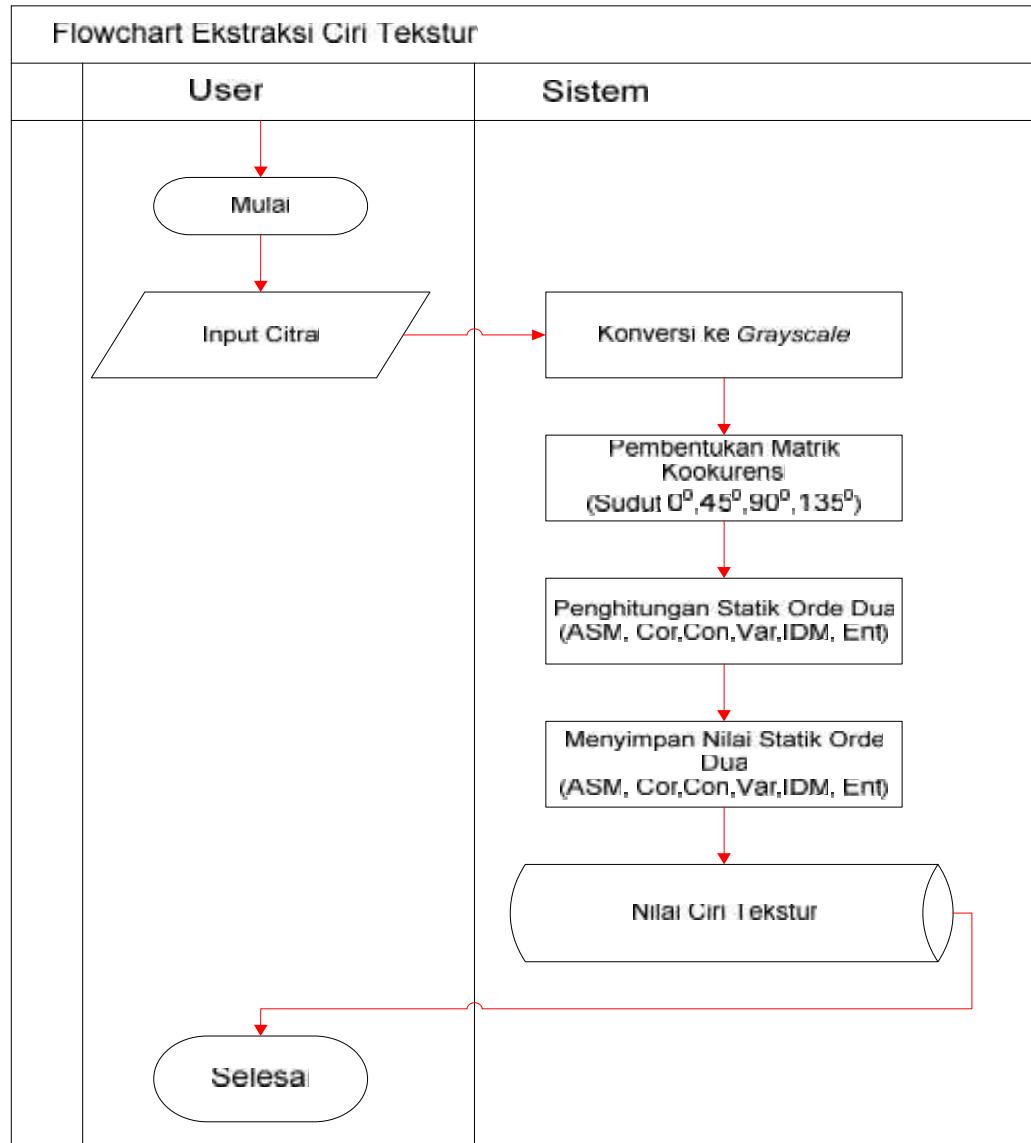
Menghitung nilai mean dan standar deviasi yang akan dijadikan nilai ciri bentuk citra ikan hias sebagai berikut.

$$\begin{aligned}\text{Mean} &= (8 \times 0.2142) + (8 \times 0.2142) + (12 \times 0.2142) + (16 \times 0.2500) + \\ &\quad (15 \times 0.2500) + (10 \times 0.1428) + (4 \times 0.1428) + (12 \times 0.2142) + \\ &\quad (12 \times 0.2142) + (8 \times 0.2142) + (3 \times 0.2500) + (6 \times 0.2500) + \\ &\quad (10 \times 0.1428) + (4 \times 0.1428) \\ &= 26.8504\end{aligned}$$

$$\begin{aligned}\text{Std} &= ((8 \times 26.8504)^2 + (8 \times 26.8504)^2 + (12 \times 26.8504)^2 + (16 \times 26.8504)^2 + \\ &\quad (15 \times 26.8504)^2 + (10 \times 26.8504)^2 + (4 \times 26.8504)^2 + (12 \times 26.8504)^2 + \\ &\quad (12 \times 26.8504)^2 + (8 \times 26.8504)^2 + (3 \times 26.8504)^2 + (6 \times 26.8504)^2 + \\ &\quad (10 \times 26.8504)^2 + (4 \times 26.8504)^2) / (14-1) \\ &= 264.3732\end{aligned}$$

4.3.1.3 Ekstraksi Ciri Tekstur

Ekstraksi ciri tekstur merupakan tahapan untuk mendapatkan nilai tekstur. Nilai tekstur didalam proses temu kembali citra digunakan untuk mengenali tekstur dari citra ikan hias. Ekstraksi ciri tekstur untuk mendapatkan nilai tekstur didalam sistem ini menggunakan penghitungan statik orde dua. Flowchart ekstraksi ciri tekstur dapat dilihat pada gambar 4.16.



Gambar 4.16 Flowchart Ekstraksi Ciri Tekstur

Berdasarkan gambar 4.16 flowchart ekstraksi ciri tekstur langkah-langkah ekstraksi ciri tekstur sebagai berikut.

1. Konversi citra ke *grayscale*.

Konversi citra menjadi *grayscale* ini merupakan tahapan *pre-processing*. Konversi citra menjadi citra *grayscale* didalam ekstraksi ciri tekstur dilakukan agar citra dapat dilakukan proses selanjutnya, yaitu proses pembentukan matrik kookurensi. Proses konversi citra menjadi *grayscale* didalam ekstraksi ciri tekstur mempunyai proses yang sama dengan proses citra RGB menjadi *grayscale* pada proses ekstraksi ciri bentuk. Yaitu dilakukan dengan mencari nilai rata-rata dari total nilai dari warna

RGB.Fungsi matlab yang digunakan untuk mengubah warna menjadi *grayscale* adalah fungsi ‘rgb2gray’. Contoh penggunaan fungsi ini adalah sebagai berikut.

```
G=imread('gambar1.jpg');
```

```
H=rgb2gray(G);
```

Contoh perhitungan konversi citra RGB menjadi citra *grayscale* dapat dilihat pada proses ekstraksi ciri bentuk.

2. Pembentukan matrik kookurensi

Pembentukan matrik kookurensi merupakan salah satu proses *pre-processing* setelah konversi citra menjadi *grayscale* didalam ekstraksi ciri tekstur. Pembentukan matrik kookurensi dilakukan pada citra yang sudah dikonversi menjadi citra *grayscale*.

Pembentukan matrik kookurensi berdasarkan 4 posisi sudut, dimana posisi sudut-sudut citra tersebut adalah sudut 0° , 45° , 90° dan 135° . Contoh pembentukan matrik kookurensi untuk citra ukuran 4×4 pada sudut 0° sebagai berikut.

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

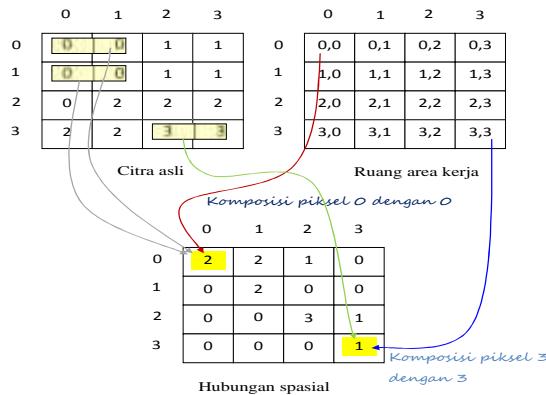
Gambar 4.17 Citra Asli

Pembentukan matrik kookurensi dimulai dengan menentukan ruang area kerja matrik berdasarkan ukuran citra. Contoh ruang area kerja matrik berdasarkan citra asli gambar 4.17

	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

Gambar 4.18 Ruang Area Kerja Matrik

Berdasarkan ruang area kerja matrik tersebut dicari hubungan spasial dengan matrik citra asli. Contoh pencarian hubungan spasial antara matrik asli dengan ruang area kerja matrik seperti berikut.



Gambar 4.19 Pencarian Hubungan Spasial

Berdasarkan gambar 4.19 Nilai 2 (0,0) pada matrik hubungan spasial berarti jumlah pasangan (0,0) pada matrik asli berjumlah 2. Sedangkan nilai 1 (3,3) pada matrik hubungan spasial diperoleh dari jumlah pasangan (0,1) pada matrik asli berjumlah 2.

Proses selanjutnya adalah berdasarkan matrik hubungan spasial akan dibentuk matrik simetris terhadap sumbu diagonal. Pembentukan matrik simetris ini dilakukan dengan cara melakukan perkalian antara matrik hubungan spasial dengan transposenya. Contoh pembentukan matris simetris sebagai berikut.

$$\begin{array}{cccc} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{array}
 + \begin{array}{cccc} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{array}
 = \begin{array}{cccc} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{array}$$

Matrik asli Transpose matrik simetris

Gambar 4.20 Pencarian Matrik Simetris Terhadap Sumbu

Proses selanjutnya akan dilakukan normalisasi. Normalisasi dilakukan dengan cara setiap nilai piksel pada matrik simetris akan dibagi dengan jumlah seluruh nilai piksel pada matrik simetris. Contoh normalisasi dapat dilihat sebagai berikut.

$$\begin{array}{cccc}
 \frac{4}{24} & \frac{2}{14} & \frac{1}{14} & \frac{0}{14} \\
 \frac{2}{14} & \frac{4}{14} & \frac{0}{14} & \frac{0}{14} \\
 \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} \\
 \frac{1}{14} & \frac{0}{14} & \frac{6}{14} & \frac{1}{14} \\
 \frac{0}{14} & \frac{0}{14} & \frac{1}{14} & \frac{2}{14} \\
 \hline
 \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14}
 \end{array}
 \Rightarrow \begin{array}{cccc}
 0.16 & 0.08 & 0.04 & 0 \\
 0.08 & 0.16 & 0 & 0 \\
 0.04 & 0 & 0.25 & 0.04 \\
 0 & 0 & 0.04 & 0.08
 \end{array}$$

Proses normalisasi Hasil normalisasi

Gambar 4.21 Proses Normalisasi Elemen Matrik

Berdasarkan gambar 4.21 hasil normalisasi merupakan matrik yang sudah ternormalisasi. Matrik yang sudah ternormalisasi tersebut adalah hasil akhir dari pembentukan matrik kookurensi sudut 0^0 . Proses pembentukan nilai matrik kookurensi untuk sudut yang lain sama dengan proses pembentukan matrik kookurensi sudut 0^0 .

Pada tahapan pembentukan matrik kookurensi nilai akhir dari keempat sudut akan diakukan penjumlahan secara keseluruhan dan setelah dilakukan penjumlahan maka akan dilakukan pembagian dengan jumlah banyaknya posisi sudut citra yang digunakan, dimana posisi sudut yang digunakan didalam pembentukan matrik kookuensi adalah $0,45,90$ dan 135 sehingga jumlah posisi sudut yang digunakan berjumlah 4 buah sehingga pada hasil akhir setiap matrik koorensi yang terbentuk dibagi dengan 4.

3. Penghitungan statik orde dua.

Penghitungan statik orde dua merupakan tahapan processing didalam ekstraksi ciri tekstur. Tahapan ekstraksi ciri tekstur menggunakan penghitungan statik orde dua dimulai dengan mencari 6 nilai ciri (*angular second moment, contrast, Correlation, Variance, Inverse difference moment dan entropy*). Penghitungan 6 nilai ciri dilakukan berdasarkan nilai matrik kookuensi yang sudah didapatkan pada tahap *pre-processing* sebelumnya. Contoh penghitungan statik orde dua pada citra 4×4 sebagai berikut.

$$\begin{matrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{matrix}$$

Gambar 4.22 Matrik 4x4

Pembentukan matrik kookurensi sudut $0^0, 45^0, 90^0$ dan 135^0 dari citra asli.

$$0^0 = \begin{matrix} 0.1600 & 0.0800 & 0.0400 & 0 \\ 0.0800 & 0.1600 & 0 & 0 \\ 0.0400 & 0 & 0.2500 & 0.0400 \\ 0 & 0 & 0.0400 & 0.0800 \end{matrix}$$

$$45^0 = \begin{matrix} 0.1600 & 0 & 0.0800 & 0.0400 \\ 0 & 0.2500 & 0 & 0 \\ 0.0800 & 0 & 0.2500 & 0 \\ 0.0400 & 0 & 0 & 0.0800 \end{matrix}$$

$$90^0 = \begin{matrix} 0.2500 & 0 & 0.0800 & 0 \\ 0 & 0.1600 & 0.0800 & 0 \\ 0.0800 & 0.0800 & 0.0800 & 0.0800 \\ 0 & 0 & 0.0800 & 0 \end{matrix}$$

$$135^0 = \begin{matrix} 0.2500 & 0.0400 & 0.0400 & 0 \\ 0.0400 & 0.1600 & 0.0400 & 0 \\ 0.0400 & 0.0400 & 0.0800 & 0.1200 \\ 0 & 0 & 0.1200 & 0 \end{matrix}$$

$$\text{Matrikkookurensi} = (0^0 + 45^0 + 900 + 350^0) / 4;$$

$$\text{Matrikkookurensi} = \begin{matrix} 0.2050 & 0.0300 & 0.0600 & 0.0100 \\ 0.0300 & 0.1825 & 0.0300 & 0 \\ 0.0600 & 0.0300 & 0.1650 & 0.0600 \\ 0.0100 & 0 & 0.0600 & 0.0400 \end{matrix}$$

$$I = [1, 2, 3, 4];$$

$$\text{SumX} = \text{sum}(\text{Matrikkookurensi}) = [0.3050, 0.2425, 0.3150, 0.1100];$$

$$\text{SumY} = \text{sum}(\text{Matrikkookurensi}) = [0.3050, 0.2425, 0.3150, 0.1100];$$

$$\text{MeanX} = \text{sumX} \times I = 2.1750;$$

$$\text{MeanY} = \text{sumY} \times I = 2.1750;$$

$$x = \sqrt{(I - \text{meanX})^2 \times \text{SumX}} = 1.0046$$

$$y = \sqrt{(I - \text{meanY})^2 \times \text{SumX}} = 1.0046$$

Menghitung 6 nilai ciri tekstur berdasarkan rumus 2.21, 2.22, 2.23, 2.24, 2.25 dan 2.26.

$$\text{Ciri ASM} = \text{sum}(\text{sum}(\text{Matrikkookurensi}^2)) = 0.9458;$$

$$\begin{aligned} \text{CiriContrast} = & 0 + ((1-1) \times (1-1) \times 0.2050) + ((1-2) \times (1-2) \times 0.0300) + ((1-3) \times \\ & (1-3) \times 0.0600) + ((1-4) \times (1-4) \times 0.0100) + ((2-1) \times (2-1) \times \\ & 0.0300) + ((2-2) \times (2-2) \times 0.1825) + ((2-3) \times (2-3) \times 0.0300) + \\ & ((2-4) \times (2-4) \times 0) + ((3-1) \times (3-1) \times 0.0600) + ((3-2) \times (3-2) \times \\ & 0.0300) + ((3-3) \times (3-3) \times 0.1650) + ((3-4) \times (3-4) \times 0.0600) + \\ & ((4-1) \times (4-1) \times 0.0100) + ((4-2) \times (4-2) \times 0) + ((4-3) \times (4-3) \times \\ & 0.0600) + ((4-4) \times (4-4) \times 0.0400) \\ = & 0.9000 \end{aligned}$$

$$\begin{aligned} \text{CiriCorrelation} = & (0 + (1 \times 1 \times 0.2050) + (1 \times 2 \times 0.0300) + (1 \times 3 \times 0.0600) + \\ & (1 \times 4 \times 0.0100) + (2 \times 1 \times 0.0300) + (2 \times 2 \times 0.1825) + \\ & (2 \times 3 \times 0.0300) + (2 \times 4 \times 0) + (3 \times 1 \times 0.0600) + (3 \times 2 \times 0.0300) + \end{aligned}$$

$$\begin{aligned}
& (3 \times 3 \times 0.1650) + (3 \times 4 \times 0.0600) + (4 \times 1 \times 0.0100) + \\
& (4 \times 2 \times 0) + (4 \times 3 \times 0.0600) + (4 \times 4 \times 0.0400)) - \\
& (2.1750 \times 2.1750) / (1.006 \times 1.006) \\
= & \quad 0.7456
\end{aligned}$$

$$\begin{aligned}
\text{CiriVariance} = & \quad 0 + ((1-2.1750) \times (1-2.1750) \times 0.2050) + ((1-2.1750) \times \\
& (2-2.1750) \times 0.0300) + ((1-2.1750) \times (3-2.1750) \times 0.0600) + \\
& ((1-2.1750) \times (4-2.1750) \times 0.0100) + ((2-2.1750) \times (1-2.1750) \\
& \times 0.0300) + ((2-2.1750) \times (2-2.1750) \times 0.1825) + ((2-2.1750) \times \\
& (3-2.1750) \times 0.0300) + ((2-2.1750) \times (4-2.1750) \times 0) + \\
& ((3-2.1750) \times (1-2.1750) \times 0.0600) + ((3-2.1750) \times (2-2.1750) \\
& \times 0.0300) + ((3-2.1750) \times (3-2.1750) \times 0.1650) + ((3-2.1750) \times \\
& (4-2.1750) \times 0.0600) + ((4-2.1750) \times (1-2.1750) \times 0.0100) + \\
& ((4-2.1750) \times (2-2.1750) \times 0) + ((4-2.1750) \times (3-2.1750) \times \\
& 0.0600) + ((4-2.1750) \times (4-2.1750) \times 0.0400) \\
= & \quad 0.5593
\end{aligned}$$

$$\begin{aligned}
\text{CiriIDM} = & \quad 0 + (0.2050 / (1 + (1-1) \times (1-1))) + (0.0300 / (1 + (1-2) \times \\
& (1-2))) + (0.0600 / (1 + (1-3) \times (1-3))) + (0.0100 / (1 + (1-4) \times \\
& (1-4))) + (0.0300 / (2 + (2-1) \times (2-1))) + (0.1825 / (2 + (2-2) \times \\
& (2-2))) + (0.0300 / (2 + (2-3) \times (2-3))) + (0 / (2 + (2-4) \times (2-4))) + \\
& (0.0600 / (3 + (3-1) \times (3-1))) + (0.0300 / (3 + (3-2) \times (3-2))) + \\
& (0.1650 / (3 + (3-3) \times (3-3))) + (0.0600 / (3 + (3-4) \times (3-4))) + \\
& (0.0100 / (4 + (4-1) \times (4-1))) + (0 / (4 + (4-2) \times (4-2))) + \\
& (0.0600 / (4 + (4-3) \times (4-3))) + (0.0400 / (4 + (4-4) \times (4-4))) \\
= & \quad 0.4531
\end{aligned}$$

$$\begin{aligned}
\text{CiriEntropy} = & \quad 0 - (0.2050 \times (-2.2863)) - (0.0300 \times (-5.0589)) - (0.0600 \times (- \\
& 4.0589)) - (0.0100 \times (-6.6439)) - (0.0300 \times (-5.0589)) - (0.1825 \times \\
& (-2.4540)) - (0.0300 \times (-5.0589)) - (0 \times (-52)) - (0.0600 \times (- \\
& 4.0589)) - (0.0300 \times (-5.0589)) - (0.1650 \times (-2.5995)) - (0.0600 \times (- \\
& 4.0589)) - (0.0100 \times (-6.6439)) - (0 \times (-52)) - (0.0600 \times (- \\
& 4.0589)) - (0.0400 \times (-4.6439)) \\
= & \quad 3.2453
\end{aligned}$$

4.4 Analisa Indeks

Proses indeks dilakukan untuk menyimpan nilai – nilai ciri hasil ekstraksi ciri sebelum dilakukan proses pengukuran kemiripan. Nilai- nilai ciri di simpan dalam file.MAT. Proses dilakukan pada citra latih atau citra yang dijadikan sebagai *database*. Data citra yang di indeks berjumlah 20 citra ikan hias dari 5 jenis ikan hias. Stuktur penyimpanan nilai ciri terdiri atas penyimpanan nama file citra, nilai ciri warna, nilai bentuk terdiri dari nilai mean dan standar deviasi dan nilai ciri tekstur terdiri dari nilai ASM,CON,COR,VAR,IDM dan ENT. Rincian struktur penyimpanan didalam *database* sebagai berikut.

Tabel 4.4 Struktur Penyimpanan Nama File Citra

Nama File Citra <i>Database</i> Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.5 Struktur Penyimpanan Nilai Warna

Nilai Histogram Citra <i>Database</i> Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.6 Struktur Penyimpanan Nilai *Angular Second Moment*.

Nilai ASM Citra <i>Database</i> Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.7 Struktur Penyimpanan Nilai *Contrast*

Nilai CON Citra <i>Database</i> Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.8 Struktur Penyimpanan Nilai *Correlation*

Nilai COR Citra <i>Database</i> Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.9 Struktur Penyimpanan Nilai *Variance*

Nilai VAR Citra <i>Database</i> Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.10 Struktur Penyimpanan Nilai *Inverse Different Moment*

Nilai IDM Citra Database Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.11 Struktur Penyimpanan Nilai *Entropy*

Nilai ENT Citra Database Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.12 Struktur Penyimpanan Nilai *Mean*

Nilai Mean Citra Database Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabel 4.13 Struktur Penyimpanan Nilai *Standar Deviasi*

Nilai Mean Citra Database Dari Citra Ke 1- 20																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

4.5 Analisa Pengukuran Kemiripan

Proses pengukuran kemiripan merupakan analisa yang dilakukan untuk melihat tingkat kemiripan antara citra *database* dengan citra kueri. Proses pengukuran kemiripan diukur dengan menggunakan rumus jarak euclidean. Proses pengukuran kemiripan dilakukan terhadap 4 model pencarian citra ikan hias, yaitu pengukuran kemiripan citra pada sistem pencarian berdasarkan fitur warna, pengukuran kemiripan citra berdasarkan fitur bentuk, pengukuran kemiripan citra pada sistem pencarian berdasarkan fitur tekstur dan pengukuran kemiripan citra berdasarkan gabungan fitur warna, bentuk dan tekstur citra.

4.5.1 Pengukuran Kemiripan Citra Berdasarkan Ciri Warna

Pencarian citra berdasarkan ekstraksi fitur warna merupakan model pencarian citra ikan hias berdasarkan nilai warna yang terkandung didalam citra ikan hias tanpa memperhatikan bentuk ataupun tekstur ikan hias.

Proses pengukuran kemiripan citra pada pencarian citra ikan hias berdasarkan warna diproses dengan cara mencari nilai jarak antara histogram model warna HSV kueri dengan histogram semua citra didalam *database*. Semua histogram yang terbentuk dari citra *database* dibandingkan dengan histogram citra

kueri sehingga akan diperoleh nilai jarak kemiripan. Setelah diperoleh nilai jarak kemiripan antar citra, maka citra akan diurutkan berdasarkan nilai jarak terendah, dimana nilai jarak warna terendah merupakan citra yang memiliki tingkat kemiripan lebih dekat dengan citra kueri. Contoh perhitungan pengukuran kemiripan berdasarkan persamaan 2.28 sebagai berikut.

Tabel 4.14 Nilai Histogram Citra Kueri

Citra	Nilai H	Frekuensi Kemunculan	Nilai S	Frekuensi Kemunculan	Nilai V	Frekuensi Kemunculan
Kueri	0	4	0	11	0	6
	120	15	0.5	9	0.5	13
	240	4	1	5	1	6
	330	2				

Tabel 4.15 Nilai Histogram Citra Database

Citra Basidata	Nilai H	Frekuensi Kemunculan	Nilai S	Frekuensi Kemunculan	Nilai V	Frekuensi Kemunculan
Citra 1	0	8	0	11	0.2	5
	110	4	0.3	12	0.3	6
	120	7	0.6	1	0.6	10
	240	5	1	1	0.9	4
	330	1				
Citra 2	0	6	0.2	14	0.4	4
	50	5	0.4	4	0.7	12
	150	7	0.7	2	0.9	7
	210	5	1	7	1	2
	300	2				
Citra 3	20	6	0	6	0	8
	50	8	0.5	9	0.4	7
	100	10	0.6	4	0.7	2
	170	1	0.9	5	0.9	7
			1	1	1	1
Citra 4	35	10	0.7	6	0.7	3
	70	12	0.8	9	0.8	3
	80	1	0.9	5	0.9	13
	120	1	1	5	1	6
	350	1				
Citra 5	0	4	0	12	0	6
	8	9	0.5	4	0.7	17
	16	7	0.8	7	0.9	1
	50	7	0.9	2	1	1
	280	3				

Nilai jarak kemiripan citra kueri dengan citra *database* citra 1

$$d_e = \sqrt{(4-8)^2 + (0-4)^2 + (15-7)^2 + (4-5)^2 + (2-1)^2 + (11-11)^2 + (0-12)^2 + (9-0)^2 + (0-1)^2 + (5-1)^2}$$
$$= 18.4391$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 2

$$d_e = \sqrt{(4-6)^2 + (0-5)^2 + (15-0)^2 + (0-7)^2 + (0-5)^2 + (0-2)^2 + (2-0)^2 + (11-0)^2 + (0-14)^2 + (0-4)^2 + (9-0)^2 + (0-2)^2 + (5-7)^2 + (6-2)^2 + (0-4)^2 + (13-0)^2 + (0-12)^2 + (0-7)^2 + (9-2)^2}$$
$$= 34.6554$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 3

$$d_e = \sqrt{(4-0)^2 + (0-6)^2 + (0-8)^2 + (0-10)^2 + (15-0)^2 + (0-1)^2 + (2-0)^2 + (11-6)^2 + (9-9)^2 + (0-4)^2 + (0-5)^2 + (5-1)^2 + (6-8)^2 + (0-7)^2 + (13-0)^2 + (0-2)^2 + (0-7)^2 + (9-1)^2}$$
$$= 29.4449$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 4

$$d_e = \sqrt{(4-0)^2 + (0-10)^2 + (0-12)^2 + (0-1)^2 + (15-1)^2 + (2-0)^2 + (0-0)^2 + (11-0)^2 + (9-0)^2 + (0-6)^2 + (0-9)^2 + (0-5)^2 + (5-5)^2 + (6-6)^2 + (13-0)^2 + (0-3)^2 + (0-3)^2 + (0-13)^2}$$
$$= 34.0881$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 5

$$d_e = \sqrt{(4-4)^2 + (0-9)^2 + (0-7)^2 + (0-2)^2 + (15-0)^2 + (4-0)^2 + (0-3)^2 + (2-0)^2 + (11-12)^2 + (9-4)^2 + (0-7)^2 + (0-2)^2 + (5-0)^2 + (6-6)^2 + (13-0)^2 + (0-17)^2 + (0-1)^2 + (6-1)^2}$$
$$= 31.2410$$

Tabel 4.16 Nilai Jarak Kemiripan Citra Berdasarkan Ekstraksi Ciri Warna

Citra	Rangking	Nilai Jarak
Citra 1	1	18.4391
Citra 2	5	34.6554
Citra 3	2	29.4449
Citra 4	4	34.0881
Citra 5	3	31.2410

Berdasarkan nilai jarak kemiripan menggunakan euclidean distance pada ekstraksi ciri warna, maka urutan citra hasil temu kembali citra yang akan tampil atau keluar sebagai output adalah dimulai dari citra 1, citra 3, citra 5, citra 4 dan terakhir citra 2.

4.5.2 Pengukuran Kemiripan Citra Berdasarkan Fitur Bentuk

Pencarian citra berdasarkan fitur bentuk ini merupakan pencarian dengan memanfaatkan nilai bentuk pada gambar, sehingga pencarian hanya berdasarkan pada bentuk citra ikan hias tanpa memperdulikan warna ataupun tekstur dari citra ikan hias.

Proses pengukuran kemiripan citra pada tipe pencarian berdasarkan bentuk dilakukan dengan cara menghitung nilai jarak antara nilai mean dan standar deviasi citra kueri dengan nilai mean dan standar deviasi yang terdapat didalam citra *database*. Proses pengukuran kemiripan citra ini memperoleh nilai jarak kemiripan citra, dimana nilai jarak yang terendah merupakan citra yang memiliki tingkat kemiripan lebih dekat dengan citra kueri. Setelah diperoleh nilai jarak maka akan dilakukan pengurutan citra, dimana gambar yang memiliki nilai jarak bentuk terdekat akan tampil dengan rangking paling tinggi. Contoh perhitungan pengukuran kemiripan berdasarkan ekstraksi ciri bentuk menggunakan persamaan 2.28 sebagai berikut.

Tabel 4.17 Nilai Ciri Bentuk Citra kueri

Citra	Mean	STD
Kueri	32.4000	18.6667

Tabel 4.18 Nilai Ciri Bentuk Citra Basidata

Citra Basidata	Mean	STD
Citra 1	15.1500	24.4431
Citra 2	10	16
Citra 3	19	27.7441
Citra 4	45.2554	6.4775
Citra 5	23.5523	22.1432

Nilai jarak kemiripan citra kueri dengan citra *database* citra 1

$$d_e = \sqrt{(32.400-15.1500)^2 + (18.6667-24.4431)^2}$$

$$\approx 18.1915$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 2

$$d_e = \sqrt{(32.400-10)^2 + (18.6667-16)^2}$$

$$\approx 22.5582$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 3

$$d_e = \sqrt{(32.400-19)^2 + (18.6667-24.7441)^2}$$

$$\approx 14.7138$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 4

$$d_e = \sqrt{(32.400-45.2554)^2 + (18.6667-6.4775)^2}$$

$$\approx 17.7155$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 5

$$d_e = \sqrt{(32.400-23.5523)^2 + (18.6667-22.1432)^2}$$

$$= 9.5062$$

Tabel 4.19 Nilai Jarak Kemiripan Citra Berdasarkan Ekstraksi Ciri Bentuk

Citra	Rangking	Nilai Jarak
Citra 1	4	18.1915
Citra 2	5	22.5582
Citra 3	2	14.7138
Citra 4	3	17.7155
Citra 5	1	9.5062

Berdasarkan nilai jarak kemiripan menggunakan euclidean distance maka urutan citra hasil temu kembali citra yang akan tampil atau keluar sebagai output adalah dimulai dari citra 5, citra 3, citra 4, citra 1 dan citra 2.

4.5.3 Pengukuran Kemiripan Citra Berdasarkan Ciri Tekstur

Pencarian citra berdasarkan ciri tekstur merupakan pencarian dengan memanfaatkan nilai tekstur pada gambar, sehingga pencarian hanya berdasarkan pada tekstur citra ikan hias tanpa memperdulikan warna ataupun benuk dari ikan hias.

Pengukuran kemiripan citra pada pencarian berdasarkan tekstur citra ikan hias dilakukan dengan cara membandingkan nilai jarak antara 6 nilai ciri tekstur pada citra kueri dengan 6 nilai ciri yang terdapat pada citra *database*.

Setelah diperoleh nilai jarak antara citra kueri dengan citra *database* maka citra akan diurutkan berdasarkan nilai kemiripan, dimana citra yang memiliki tingkat kemiripan tinggi adalah citra yang memiliki nilai jarak mendekati 0. Contoh perhitungan pengukuran kemiripan berdasarkan ekstraksi ciri tekstur menggunakan persamaan 2.28 sebagai berikut.

Tabel 4.20 Nilai Ciri Tekstur Citra Kueri

Citra	Asm	Contrast	Correlation	Variance	Idm	Entropy
Kueri	0.9458	0.9000	0.7456	0.5593	0.4531	3.2454

Tabel 4.21 Nilai Ciri Tekstur Citra Database

Citra <i>Database</i>	ASM	Contrast	Correlation	Variance	Idm	Entropy
Citra 1	0	0.4226	2	1.4431	0.5421	4.5241
Citra 2	0.5438	2.4465	4.2667	4.7241	4.2241	6.7741
Citra 3	2	3	4	7.2564	4.4411	1.2564
Citra 4	0.4765	4.7721	6.5446	9.2465	7.6124	5.4245
Citra 5	7.6445	6	4.6654	7.4456	9	6.7447

Nilai jarak kemiripan citra kueri dengan citra *database* citra 1

$$d_e = \sqrt{(0.9458-1)^2 + (0.900-0.4226)^2 + (0.7456-2)^2 + (0.5593-1.4431)^2 + (0.4531-0.5421)^2 + (3.2454-4.5241)^2}$$

$$= 2.0563$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 2

$$d_e = \sqrt{(0.9458-0.5438)^2 + (0.900-2.4465)^2 + (0.7456-4.2667)^2 + (0.5593-4.7424)^2 + (0.4531-4.2241)^2 + (3.2454-6.7741)^2}$$

$$= 7.6891$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 3

$$d_e = \sqrt{(0.9458-2)^2 + (0.900-3)^2 + (0.7456-4)^2 + (0.5593-7.2564)^2 + (0.4531-4.4411)^2 + (3.2454-1.2564)^2}$$

$$= 8.9902$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 4

$$d_e = \sqrt{(0.9458-0.4765)^2 + (0.900-4.7721)^2 + (0.7456-6.5446)^2 + (0.5593-9.2465)^2 + (0.4531-7.6124)^2 + (3.2454-5.4245)^2}$$
$$= 13.4281$$

Nilai jarak kemiripan citra kueri dengan citra *database* citra 5

$$d_e = \sqrt{(0.9458-7.6445)^2 + (0.900-6)^2 + (0.7456-4.6654)^2 + (0.5593-7.4456)^2 + (0.4531-9)^2 + (3.2454-6.7447)^2}$$
$$= 14.7974$$

Tabel 4.22 Nilai Jarak Kemiripan Citra Berdasarkan Ekstraksi Ciri Tekstur

Citra	Rangking	Nilai Jarak
Citra 1	1	2.0563
Citra 2	2	7.6891
Citra 3	3	8.9902
Citra 4	4	13.4281
Citra 5	5	14.7974

Berdasarkan nilai jarak kemiripan menggunakan euclidean distance maka urutan citra hasil temu kembali citra yang akan tampil atau keluar sebagai output adalah dimulai dari citra 1, citra 2, citra 3, citra 4 dan citra 5.

4.5.3 Pengukuran Kemiripan Citra Berdasarkan Gabungan Ciri

Model pencarian citra berdasarkan penggabungan ciri merupakan model pencarian citra ikan hias berdasarkan warna, bentuk dan tekstur citra ikan hias. Proses pengukuran kemiripan gambar berdasarkan penggabungan ekstraksi ciri dilakukan dengan cara menjumlahkan seluruh nilai jarak kemiripan ciri, kemudian berdasarkan seluruh nilai kemiripan ciri dilakukan pembagian dengan jumlah variabel yang digunakan. Dimana variabel yang digunakan berjumlah 3 yang terdiri dari variabel warna, bentuk dan tekstur. Setelah proses pembagian maka akan diperoleh nilai jarak citra.

Berdasarkan nilai jarak kemiripan citra maka citra akan diurutkan berdasarkan nilai jarak, dimana jarak nilai terendah merupakan citra didalam *database* yang memiliki tingkat kemiripan terdekat dengan citra kueri. Contoh

perhitungan untuk mencari nilai jarak kemiripan berdasarkan ekstraksi ciri gabungan dari warna, bentuk dan tekstur sebagai berikut.

Tabel 4.23 Nilai Jarak Kemiripan Citra Kueri Dengan Citra Database

Citra	Nilai Jarak Kemiripan Citra kueri Dengan Citra Database		
	Warna	Bentuk	Tekstur
Citra 1	18.4391	18.1915	2.0563
Citra 2	34.6554	22.5582	7.6891
Citra 3	29.4449	14.7138	8.9902
Citra 4	34.0881	17.7155	13.4281
Citra 5	31.2410	9.5062	14.7974

Nilai jarak kemiripan antara citra kueri dengan citra didalam basidata citra 1

$$d_e = (18.4391 + 18.1915 + 2.0563) / 3$$

$$= 12.8956$$

Nilai jarak kemiripan antara citra kueri dengan citra didalam basidata citra 2

$$d_e = (34.6554 + 22.5582 + 7.6891) / 3$$

$$= 21.6342$$

Nilai jarak kemiripan antara citra kueri dengan citra didalam basidata citra 3

$$d_e = (29.4449 + 14.7138 + 8.9902) / 3$$

$$= 17.7163$$

Nilai jarak kemiripan antara citra kueri dengan citra didalam basidata citra 4

$$d_e = (34.0881 + 17.7155 + 13.4281) / 3$$

$$= 21.7439$$

Nilai jarak kemiripan antara citra kueri dengan citra didalam basidata citra 5

$$d_e = (31.2410 + 9.5062 + 14.7974) / 3$$

$$= 18.5149$$

Tabel 4.24 Nilai Jarak Kemiripan Citra Berdasarkan Gabungan Ekstraksi Ciri

Citra	Rangking	Nilai Jarak
Citra 1	1	12.8956
Citra 2	4	21.6342
Citra 3	2	17.7163
Citra 4	5	21.7439
Citra 5	3	18.5149

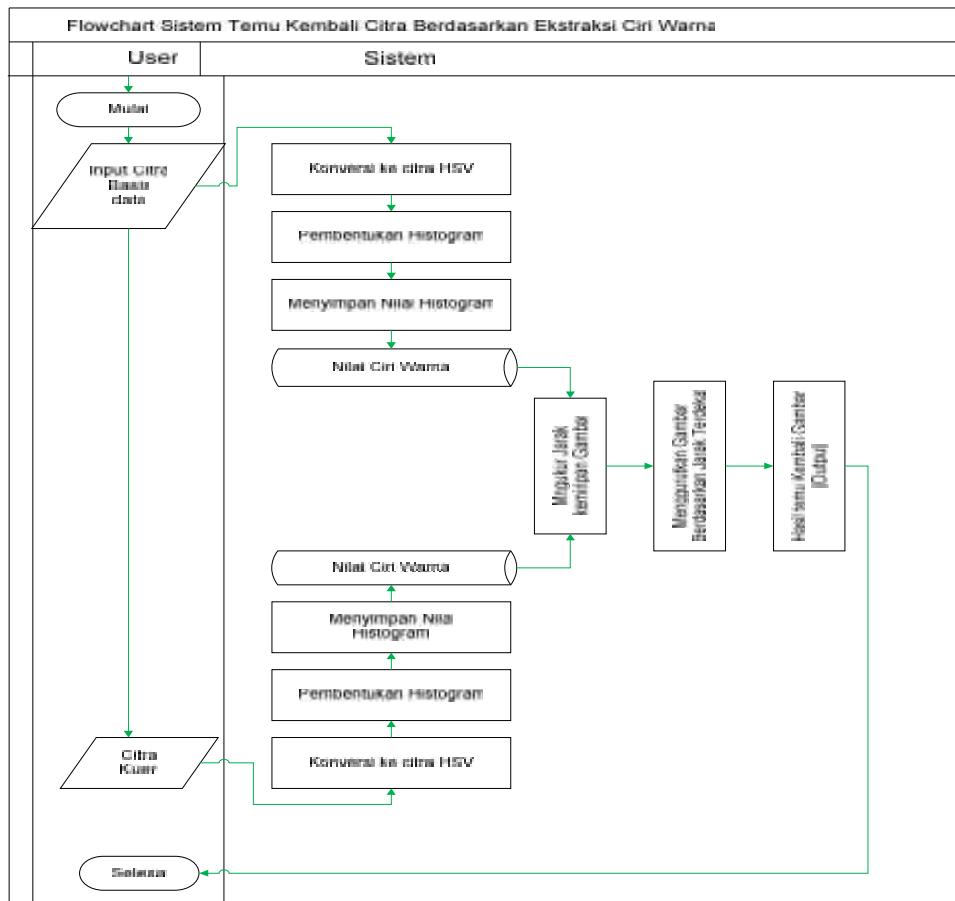
Berdasarkan nilai jarak kemiripan instance maka urutan citra hasil temu kembali citra yang akan tampil atau keluar sebagai output adalah dimulai dari citra 1, citra 3, citra 5, citra 2 dan citra 5.

4.6 Analisa Sistem Temu Kembali Citra Ikan Hias

Gambaran sistem pencarian merupakan langkah-langkah sistem yang akan dibangun. Gambaran sistem temu kembali citra ikan hias dilakukan berdasarkan ekstraksi ciri warna, bentuk, tekstur dan penggabungan dari ekstraksi ciri.

4.6.1 Analisa Sistem Sistem Temu Kembali Citra Berdasarkan Ekstraksi Ciri Warna

Gambaran sistem temu kembali citra ikan hias berdasarkan ekstraksi ciri warna digunakan untuk proses temu kembali citra ikan hias berdasarkan sebaran warna. Pada sistem pencarian citra ikan hias berdasarkan ciri warna tidak memperhatikan bentuk ataupun tekstur dari citra ikan hias, sehingga sistem pencarian hanya akan mencari citra-citra ikan hias didalam *database* yang mempunyai kemiripan sebaran warna dengan citra kueri. Flowchart gambaran sistem pencarian citra ikan hias berdasarkan ciri warna dapat dilihat pada gambar 4.23.



Gambar 4.23 Flowchart Sistem Pencarian Citra Berdasarkan Ciri Warna

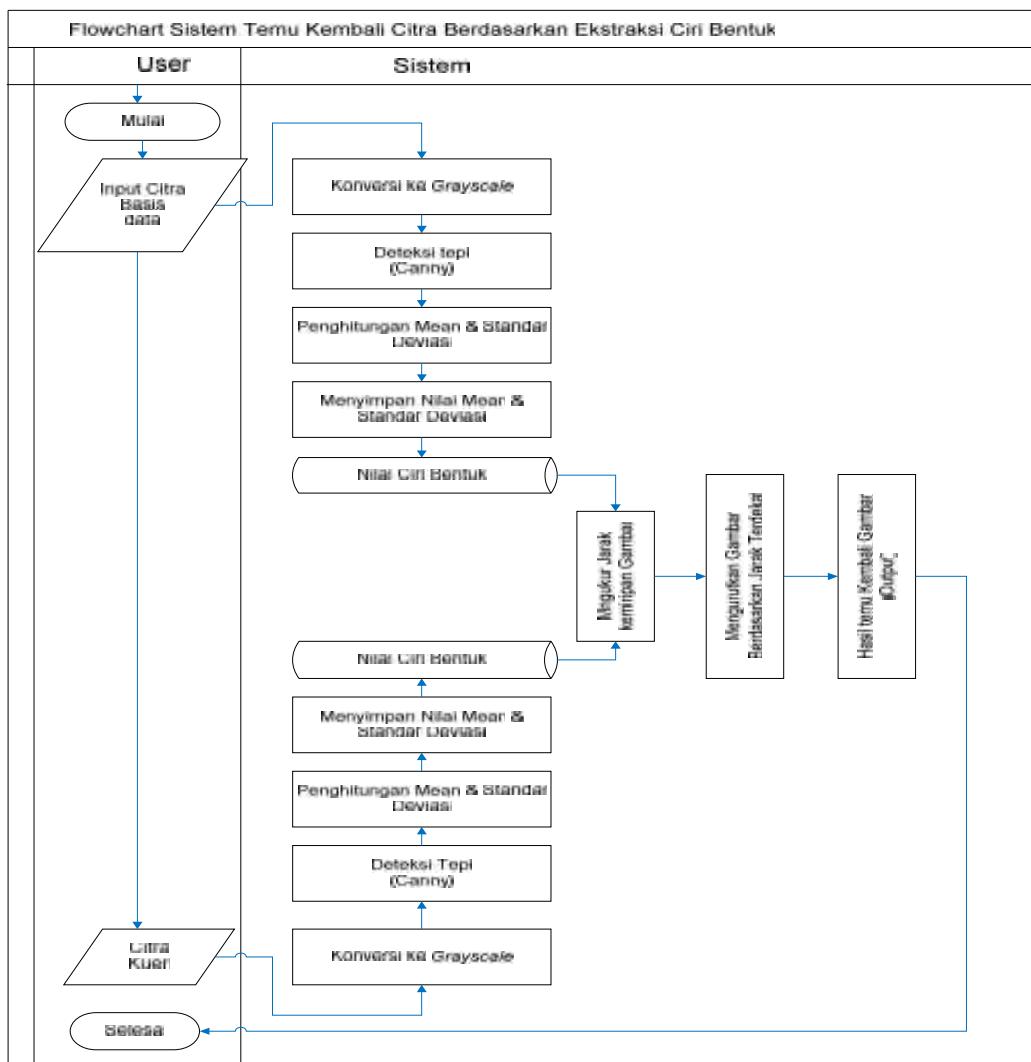
Berdasarkan gambar 4.23 flowchart pencarian citra ikan hias berdasarkan ciri warna penjelasannya sebagai berikut.

1. Citra latih dan citra kueri di inputkan, tetapi dalam hal ini citra latih di inputkan terlebih dahulu dan diproses terlebih dahulu, sehingga nilai ciri warna terlebih dahulu diperoleh dan disimpan didalam *database* terlebih dahulu. Setelah citra latih diproses kemudian proses dilanjutkan terhadap citra kueri.
2. Proses pencarian gambar ikan hias berdasarkan warna pertama citra ikan hias akan diubah menjadi citra dengan model warna HSV. Pengubahan citra menjadi HSV ini untuk mendapatkan histogram dari citra ikan hias.
3. Citra ikan hias yang sudah diubah menjadi citra dengan model warna HSV selanjutnya akan dibentuk histogram dan nilai histogram disimpan didalam *file temporary* atau file penyimpanan sementara.
4. Sistem membaca *file temporary* dari citra kueri dan *file database* dari citra *database*, kemudian membandingkan nilai-nilai pada citra kueri dan citra *database* untuk mendapatkan nilai jarak *euclidean distance* dari semua file.
5. Setelah didapatkan nilai jarak *euclidean distance* dari semua file nilai diurutkan dari nilai terkecil hingga terbesar. Citra yang memiliki nilai jarak terkecil memiliki tingkat kemiripan lebih tinggi.
6. Output atau hasil akhir pencarian citra berdasarkan ekstraksi ciri warna dengan menggunakan model warna HSV dari sistem adalah citra-citra didalam *database* yang memiliki kemiripan warna dengan citra kueri. Rangking ataupun urutan citra hasil pencarian berdasarkan proses pengurutan sebelumnya, dimana citra yang memiliki jarak kemiripan terdekat akan tampil sebagai citra dengan rangking tertinggi.

4.6.2 Analisa Sistem Sistem Temu Kembali Citra Berdasarkan Ekstraksi Ciri Bentuk

Gambaran sistem pencarian citra ikan hias berdasarkan ekstraksi ciri bentuk digunakan sistem pencarian berdasarkan bentuk citra ikan hias. Pada sistem pencarian citra ikan hias ini, sistem tidak memperhatikan warna ataupun tekstur dari citra ikan hias, sistem pencarian hanya akan mencari citra-citra ikan hias didalam *database* yang mempunyai kemiripan bentuk dengan citra kueri.

Flowchart gambaran sistem pencarian citra ikan hias berdasarkan ciri bentuk dapat dilihat pada gambar 4.24.



Gambar 4.24 Flowchart Sistem Pencarian Berdasarkan Bentuk Ikan Hias

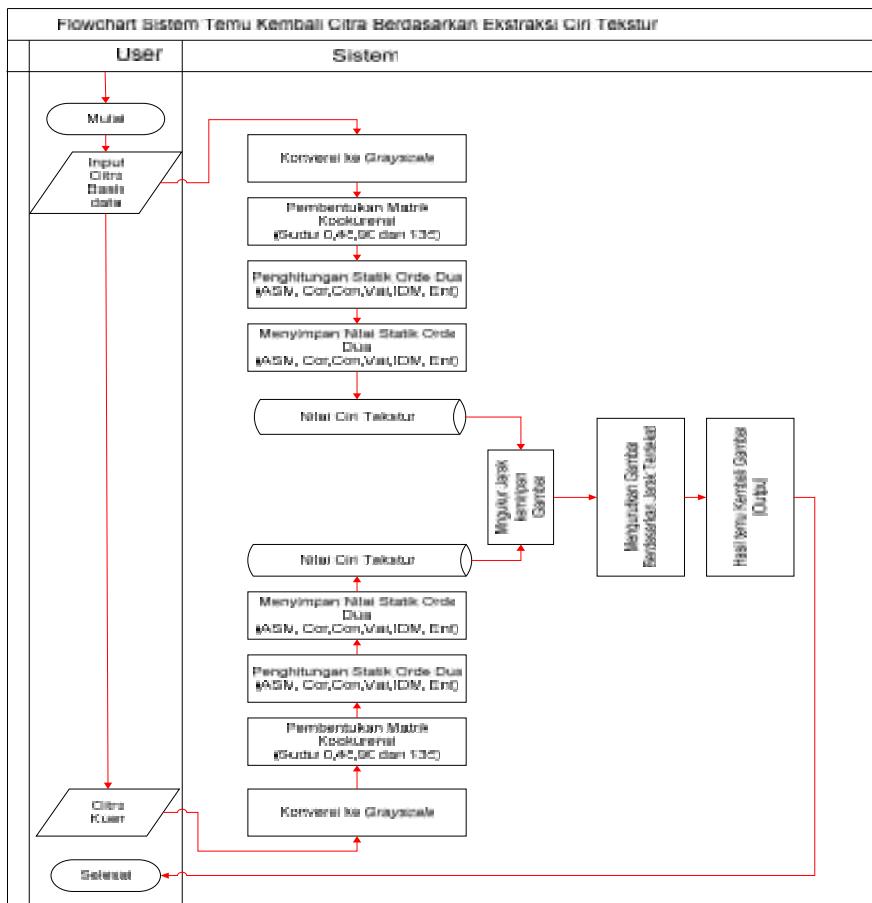
Berdasarkan gambar 4.24 langkah-langkah sistem pencarian citra ikan hias berdasarkan ciri bentuk citra ikan hias sebagai berikut.

1. Citra latih dan citra kueri di inputkan, tetapi dalam hal ini citra latih di inputkan terlebih dahulu dan diproses terlebih dahulu, sehingga nilai ciri bentuk terlebih dahulu diperoleh dan disimpan didalam *database* terlebih dahulu. Setelah citra latih diperoses kemudian proses dilanjutkan terhadap citra kueri.

2. Citra ikan hias akan diubah menjadi citra *grayscale*. Pengubahan citra menjadi *grayscale* ini untuk membantu proses selanjutnya yaitu proses deteksi tepi.
3. Deteksi tepi canny dilakukan setelah citra diubah menjadi *grayscale*. Proses deteksi tepi dilakukan untuk mendapatkan bentuk sketsa atau bentuk tepi citra ikan hias berdasarkan sketsa tepi menggunakan canny kemudian dilakukan penghitungan nilai mean dan standar deviasi.
4. Nilai mean dan standar deviasi dari citra akan disimpan sebagai nilai ciri bentuk dari citra.
5. Nilai ciri dari citra kueri disimpan didalam *file temporary* atau file penyimpanan sementara dan nilai ciri bentuk citra *database* disimpan di dalam *file database*.
6. Sistem membaca *file temporary* dari citra kueri dan *file database* dari citra *database*, kemudian membandingkan nilai-nilai pada citra kueri dan citra *database* untuk mendapatkan nilai jarak *euclidean distance* dari semua file.
7. Setelah didapatkan nilai jarak *euclidean distance* dari semua file nilai diurutkan dari nilai terkecil hingga terbesar. Citra yang memiliki nilai terkecil memiliki tingkat kemiripan lebih tinggi.
8. Output atau hasil akhir pencarian citra berdasarkan fitur bentuk citra ikan hias dari sistem adalah citra-citra didalam *database* yang memiliki kemiripan bentuk dengan citra kueri. Rangking ataupun urutan citra hasil pencarian berdasarkan proses pengurutan sebelumnya, dimana citra yang memiliki jarak kemiripan terdekat akan tampil sebagai citra dengan rangking tertinggi.

4.6.3 Analisa Sistem Sistem Temu Kembali Citra Berdasarkan Ekstraksi Ciri Tekstur

Gambaran sistem temu kembali citra ikan hias berdasarkan ekstraksi ciri tekstur digunakan untuk proses sistem pencarian citra ikan hias berdasarkan tekstur. Pada sistem pencarian citra ikan hias berdasarkan ciri tekstur tidak memperhatikan warna ataupun bentuk dari citra ikan hias, sehingga sistem pencarian hanya akan mencari citra-citra ikan hias didalam *database* yang mempunyai kemiripan tekstur dengan citra kueri. Flowchart gambaran sistem pencarian citra ikan hias berdasarkan ciri tekstur dapat dilihat pada gambar 4.25



Gambar 4.25 Flowchart Sistem Pencarian Berdasarkan Tekstur Citra Ikan Hias

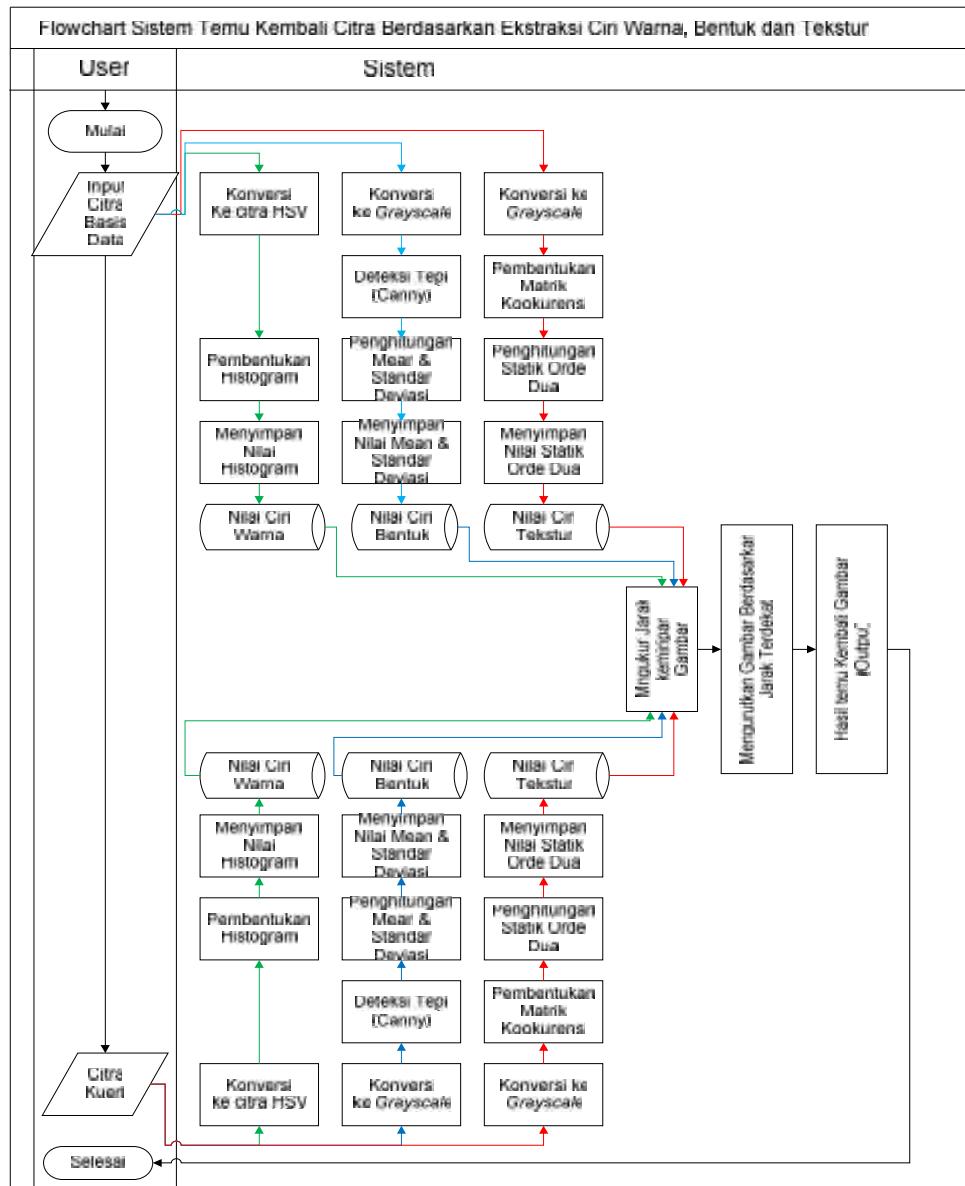
Berdasarkan gambar 4.25 langkah-langkah sistem pencarian citra ikan hias berdasarkan ciri tekstur citra ikan hias sebagai berikut.

1. Citra latih dan citra kueri di inputkan, tetapi dalam hal ini citra latih di inputkan terlebih dahulu dan diproses terlebih dahulu, sehingga nilai ciri bentuk terlebih dahulu diperoleh dan disimpan didalam *database* terlebih dahulu. Setelah citra latih diperoses kemudian proses dilanjutkan terhadap citra kueri.
2. Citra ikan hias akan diubah menjadi citra *grayscale*. Pengubahan citra menjadi *grayscale* ini untuk membantu proses selanjutnya yaitu pembentukan matrik kookurensi.
3. Pembentukan matrik kookurensi dilakukan pada citra yang sudah diubah menjadi *grayscale*. Pembentukan matrik kookurensi dilakukan berdasarkan 4 sudut berbeda ($0^0, 45^0, 90^0$ dan 135^0). Pembentukan matrik kookurensi ini untuk membantu proses selanjutnya, yaitu proses penghitungan statik orde dua.

4. Penghitungan statik orde dua dilakukan berdasarkan nilai-nilai matrik kookurensi yang sudah diperoleh pada proses sebelumnya. Penghitungan statik orde dua ini dilakukan unutk mendapatkan 6 nilai ciri tekstur (asm, con, cor, var,idm dan ent), setelah dipwrolwh 6nilai ciri tekstur ini maka 6 nilai ciri ini kan disimpan sebagai nilai ciri tekstur.
5. Nilai ciri dari citra kueri disimpan didalam *file temporary* atau file penyimpanan sementara dan nilai ciri bentuk citra *database* di simpan di dalam *file database*.
6. Sistem membaca *file temporary* dari citra kueri dan *file database* dari citra *database*, kemudian membandingkan nilai-nilai pada citra kueri dan citra *database* untuk mendapatkan nilai *euclidean distance* dari semua file.
7. Setelah didapatkan nilai *euclidean distance* dari semua file nilai diurutkan dari nilai terkecil hingga terbesar. Citra yang memiliki nilai terkecil memiliki tingkat kemiripan lebih tinggi.
8. Output atau hasil akhir pencarian citra berdasarkan ciri tekstur citra ikan hias dari sistem adalah citra-citra didalam database yang memiliki kemiripan tekstur dengan citra kueri. Rangking ataupun urutan citra hasil pencarian berdasarkan proses pengurutan sebelumnya, dimana citra yang memilki jarak kemiripan terdekat akan tampil sebagai citra dengan rangking tertinggi.

4.6.4 Analisa Sistem Pencarian Citra Berdasarkan Gabungan Ciri

Gambaran sistem temu kembali citra ikan hias berdasarkan gabungan ekstraksi ciri digunakan untuk proses sistem pencarian citra ikan hias berdasarkan gabungan semua fitur (warna, bentuk dan tekstur).Flowchart gambaran sistem pencarian citra ikan hias berdasarkan gabungan ciri (warna, bentuk dan tekstur) dapat dilihat pada gambar 4.26.



Gambar 4.26 Flowchart Sistem Pencarian Berdasarkan Gabungan Ciri

Berdasarkan gambar 4.27 langkah-langkah sistem pencarian citra ikan hias berdasarkan penggabungan ciri citra ikan hias sebagai berikut.

1. Citra latih dan citra kueri di inputkan, tetapi dalam hal ini citra latih di inputkan terlebih dahulu dan diproses terlebih dahulu, sehingga nilai ciri basisdata terlebih dahulu diperoleh dan disimpan didalam *database* terlebih dahulu. Setelah citra latih diperoses kemudian proses dilanjutkan terhadap citra kueri.
2. Citra ikan hias akan diubah menjadi citra dengan model warna HSV. Pengubahan citra menjadi model warna HSV ini untuk membantu proses selanjutnya yaitu pembentukan histogram.

3. Citra ikan hias yang sudah diubah menjadi model warna HSV selanjutnya akan dibentuk histogram dan nilai histogram disimpan didalam *file temporary* atau file penyimpanan sementara.
4. Citra ikan hias akan diubah menjadi citra *grayscale*. Pengubahan citra menjadi *grayscale* ini untuk membantu proses selanjutnya yaitu proses deteksi tepi.
5. Deteksi tepi canny dilakukan setelah citra diubah menjadi *grayscale*. Proses deteksi tepi dilakukan untuk mendapatkan bentuk sketsa atau bentuk tepi citra ikan hias. berdasarkan sketsa bentuk tepi ini kemudian akan dilakukan proses penghitungan nilai mean dan standar deviasi.
6. Menggunakan penghitungan nilai mean dan standar deviasi dari citra diperoleh kemudian nilai mean dan standar deviasi disimpan sebagai nilai ciri bentuk dari citra.
7. Nilai ciri dari citra kueri disimpan didalam *file temporary* atau file penyimpanan sementara dan nilai ciri bentuk citra *database* disimpan didalam *file database*.
8. Nilai ciri dari citra kueri disimpan didalam *file temporary* atau file penyimpanan sementara dan nilai ciri bentuk citra *database* di simpan di dalam *file database*.
9. Citra ikan hias akan diubah menjadi citra *grayscale*. Pengubahan citra menjadi *grayscale* ini untuk membantu proses selanjutnya yaitu pembentukan matrik kookurensi.
10. Pembentukan matrik kookurensi dilakukan pada citra yang sudah diubah menjadi *grayscale*. Pembentukan matrik kookurensi dilakukan berdasarkan 4 sudut berbeda (0^0 , 45^0 , 90^0 dan 135^0). Pembentukan matrik kookurensi ini untuk membantu proses selanjutnya, yaitu proses penghitungan statik orde dua.
11. Penghitungan statik orde dua dilakukan berdasarkan nilai-nilai matrik kookurensi yang sudah diperoleh pada proses sebelumnya. Penghitungan statik orde dua ini dilakukan unutk mendapatkan 6 nilai ciri tekstur (ASM (*Angular Second Moment*), CON (*Contrast*), COR (*Correlation*), VAR

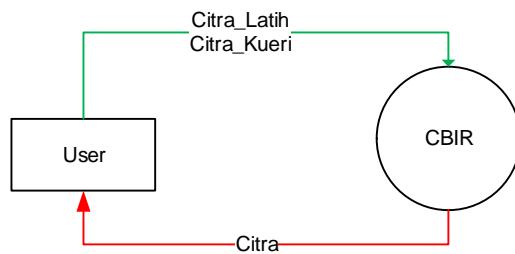
- (*Variance*), IDM (*Inverse Different Moment*) dan ENT (*Entropy*)), kemudian 6 nilai ciri tekstur disimpan sebagai nilai ciri tekstur.
12. Nilai ciri dari citra kueri disimpan didalam *file temporary* atau file penyimpanan sementara dan nilai ciri bentuk citra *database* disimpan di dalam *file database*.
 13. Sistem membaca *file temporary* dari citra kueri dan *file database* dari citra *database*, kemudian membandingkan nilai-nilai pada citra kueri dan citra *database* untuk mendapatkan nilai jarak *euclidean distance* dari semua file.
 14. Setelah didapatkan nilai jarak *euclidean distance* dari semua file nilai diurutkan dari nilai terkecil hingga terbesar. Citra yang memiliki nilai terkecil memiliki tingkat kemiripan lebih tinggi.
 15. Output atau hasil akhir pencarian citra berdasarkan gabungan ekstraksi ciri citra ikan hias dari sistem adalah citra-citra didalam database yang memiliki kemiripan warna, bentuk dan tekstur dengan citra kueri. Rangking ataupun urutan citra hasil pencarian berdasarkan proses pengurutan sebelumnya, dimana citra yang memiliki jarak kemiripan terdekat akan tampil sebagai citra dengan rangking tertinggi.

4.7 Analisa Alur Data

Analisa alur data merupakan tahapan yang menganalisa alur data dan proses secara keseluruhan proses sistem temu kembali gambar berdasarkan ekstraksi ciri warna, bentuk dan tekstur pada citra ikan hias. analisa alur data ini terdiri dari *context diagram* dan *Data Flow Diagram* (DFD).

4.7.1 Context Diagram

Context diagram digunakan untuk menggambarkan proses kerja sistem secara umum. *Conteks diagram* merupakan data flow diagram yang menggambarkan secara garis besar dari operasional sistem. *Context diagram* dari sistem pencarian citra ikan hias berdasarkan ekstraksi ciri warna, bentuk dan tekstur.



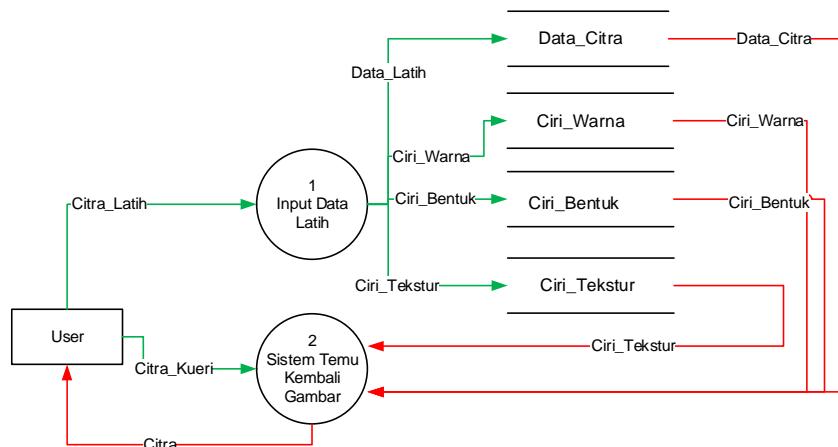
Gambar 4.27 Context Diagram

4.7.2 Data Flow Diagram (DFD)

Data flow diagram (DFD) digunakan untuk menggambarkan aliran data dan proses yang terdapat didalam sistem pencarian citra ikan hias. Proses-proses aliran data yang digambarkan pada context diagram akan dipaparkan lebih terperinci didalam *data flow diagram* (DFD)

4.7.2.1 Data flow Diagram (DFD) Level 1

Data Flow Diagram digunakan untuk merepresentasikan fungsi-fungsi yang ada didalam sistem pencarian citra ikan hias. *Data Flow Diagram* level 1 terdiri dari 1 entitas (*user*), 2 proses (input data latih dan sistem temu kembali gambar) dan menghasilkan 4 *data store* (Data_Citra, Ciri_Warna, Ciri_Bentuk dan Ciri_Tesktr).

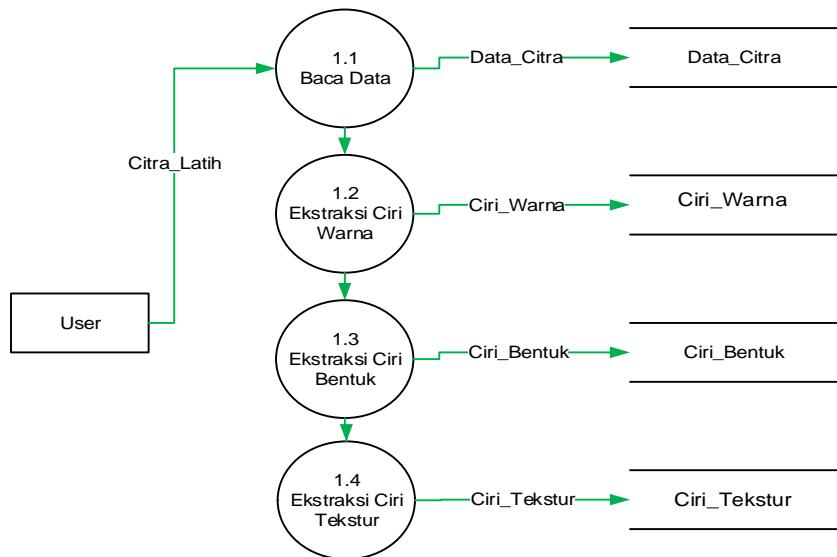


Gambar 4.28 Data flow Diagram (DFD) Level 1

4.7.2.2 Data flow Diagram (DFD) Level 2 Proses 1 Input Data Latih

Data flow diagram proses 1 yaitu proses input data latih. Proses ini merupakan proses lebih terperinci dari proses DFD level 1, dimana didalam DFD level 2 terdiri dari 1 entitas (*user*), 4 proses (baca data, ekstraksi ciri warna,

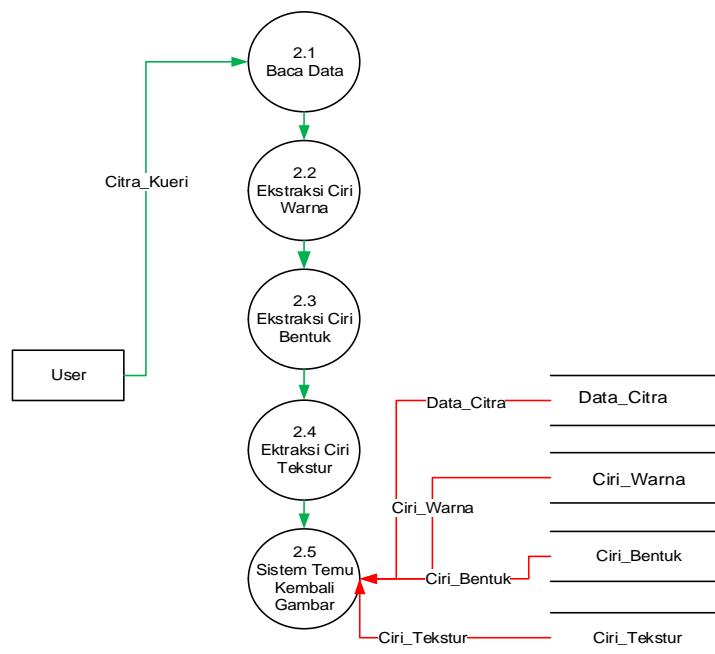
ekstraksi ciri bentuk dan ekstraksi ciri tekstur) dan menghasilkan 4 *data store* (Data_Citra, Ciri_Warna, Ciri_Bentuk dan Ciri_Tekstur).



Gambar 4.29 Data flow Diagram (DFD) Level 2 Proses 1 Input Data Latih

4.7.2.3 Data flow Diagram (DFD) Level 2 Proses 2 Temu Kembali Gambar

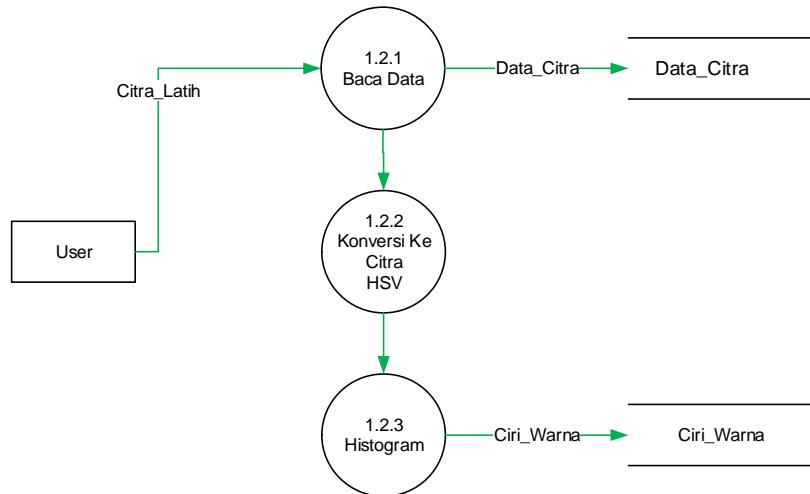
Data flow diagram level 2 proses 2 temu kembali gambar. Proses ini merupakan proses lebih terperinci dari proses DFD level 1, dimana didalam DFD level 2 terdiri dari 1 entitas (*user*) dan 4 proses (baca data, ekstraksi ciri warna, ekstraksi ciri bentuk dan ekstraksi ciri tekstur).



Gambar.4.30 Data flow Diagram (DFD) Level 2 Proses 2 Temu Kembali Citra

4.7.2.4 Data flow Diagram (DFD) Level 3 Proses 1.2 Ekstraksi Ciri Warna

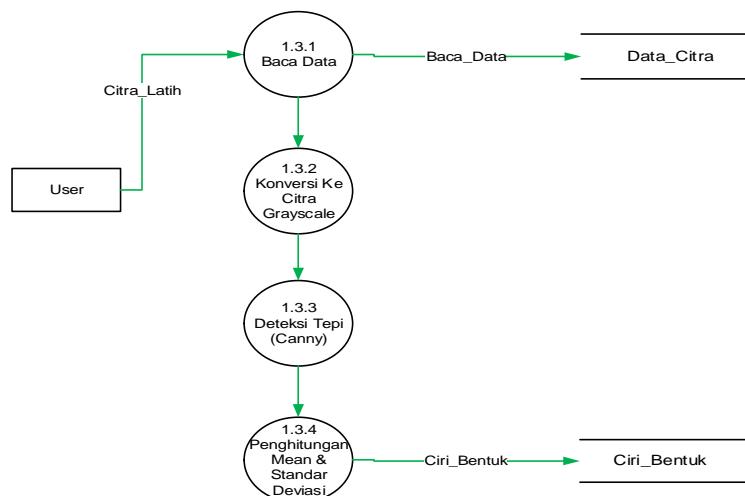
Data flow diagram level proses 1.2 ekstraksi ciri warna. Proses ini merupakan proses lebih terperinci dari proses DFD level 1, dimana didalam DFD level 2 terdiri dari 1 entitas (*user*), 3 proses (Baca Data, Konversi Ke Citra HSV dan Histogram) dan 2 *data store* (Data_Citra dan Ciri_Warna).



Gambar 4.31 Data flow Diagram (DFD) Level 3 Proses 1.2 Ekstraksi Ciri Warna

4.7.2.5 Data flow Diagram (DFD) Level 3 Proses 1.3 Ekstraksi Ciri Bentuk

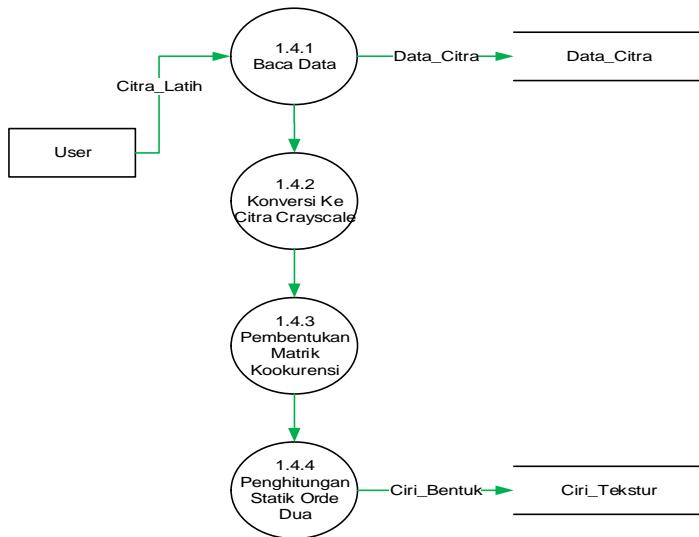
Data flow diagram level proses 1.2 ekstraksi ciri bentuk. Proses ini merupakan proses lebih terperinci dari proses DFD level 1, dimana didalam DFD level 2 terdiri dari 1 entitas (*user*), 4 proses (Baca Data, konversi ke citra grayscale, Deteksi Tepi (Canny) dan Penghitungan Mean & Standar Deviasi) dan 2 *data store* (Data_Citra dan Ciri_Bentuk).



Gambar 4.32 Data flow Diagram (DFD) Level 3 Proses 1.3 Ekstraksi Ciri Bentuk

4.7.2.5 Data flow Diagram (DFD) Level 3 Proses 1.4 Ekstraksi Ciri Tekstur

Data flow diagram level proses 1.2 ekstraksi ciri tekstur. Proses ini merupakan proses lebih terperinci dari proses DFD level 1, dimana didalam DFD level 2 terdiri dari 1 entitas (*user*), 4 proses (Baca Data, Konversi Ke Citra *Grayscale*, Pembentukan Matrik Kookurensi, dan Penghitungan Statik Orde Dua) dan 2 *data store* (Data_Citra dan Ciri_Tekstur).



Gambar 4.33 Data flow Diagram (DFD) Level 3 Proses 1.4 Ekstraksi Ciri Tekstur

4.8 Perancangan

Perancangan sistem merupakan tahapan-tahapan desain sistem yang akan dibangun berdasarkan analisa. Perancangan sistem ini akan membahas tentang rancangan *database* dan antar muka sistem.

4.8.1 Perancangan Database

Database merupakan tempat penyimpanan hasil ekstraksi ciri warna, bentuk dan tekstur. Sistem yang akan dirancang menggunakan *database* berupa MAT-file, dimana MAT-file menyimpan nilai dalam workspace matlab. Perancangan MAT-file didalam sistem yang akan dibangun sebagai berikut.

1. db.mat

Db.mat merupakan fungsi yang digunakan untuk menyimpan nilai ekstraksi ciri pada gambar yang ada didalam *database*. Nilai ciri yang disimpan didalam *database* adalah nilai ciri warna, nilai ciri bentuk dan nilai ciri tekstur. Struktur yang terdapat didalam db.mat terdapat didalam tabel 4.25.

Tabel 4.25 Struktur File db.mat

No	Nama variabel	Tipe data	Keterangan
1	Gambar	Cell	Menyimpan nama file citra
2	Hist	Cell	Menyimpan histogram model warna HSV
3	Mean	Cell	Menyimpan nilai <i>mean</i>
4	STD	Cell	menyimpan nilai standar deviasi
4	ASM	Cell	Menyimpan nilai <i>angular second moment</i>
5	CON	Cell	Menyimpan nilai <i>contrast</i>
6	COR	Cell	Menyimpan nilai <i>correlation</i>
7	VAR	Cell	Menyimpan nilai <i>variance</i>
8	IDM	Cell	Menyimpan nilai <i>inferse different moment</i>
9	ENT	Cell	Menyimpan nilai <i>entropy</i>

2. Hist.mat

Hist.mat merupakan fungsi yang digunakan untuk menyimpan nilai ekstraksi ciri warna dari gambar kueri. Struktur fungsi hist.mat terlihat pada tabel 4.26.

Tabel 4.26 Struktur Hist.mat

No	Nama variabel	Tipe data	Keterangan
1	Gambar	Cell	Menyimpan nama file citra
2	Hist	Cell	Menyimpan histogram model warna HSV

3. Bentuk.mat

Fungsi bentuk.mat digunakan untuk menyimpan nilai ciri bentuk kueri. Struktur bentuk.mat terlihat pada tabel 4.27.

Tabel 4.27 Struktur File Bentuk.Mat

No	Nama Variabel	Tipe Data	Keterangan
1	Gambar	Cell	menyimpan nama file citra kueri
2.	Mean	Cell	Menyimpan nilai mean citra Kueri
3	STD	Cell	Menyimpan nilai standar deviasi citra kueri

4. Tekstur.mat

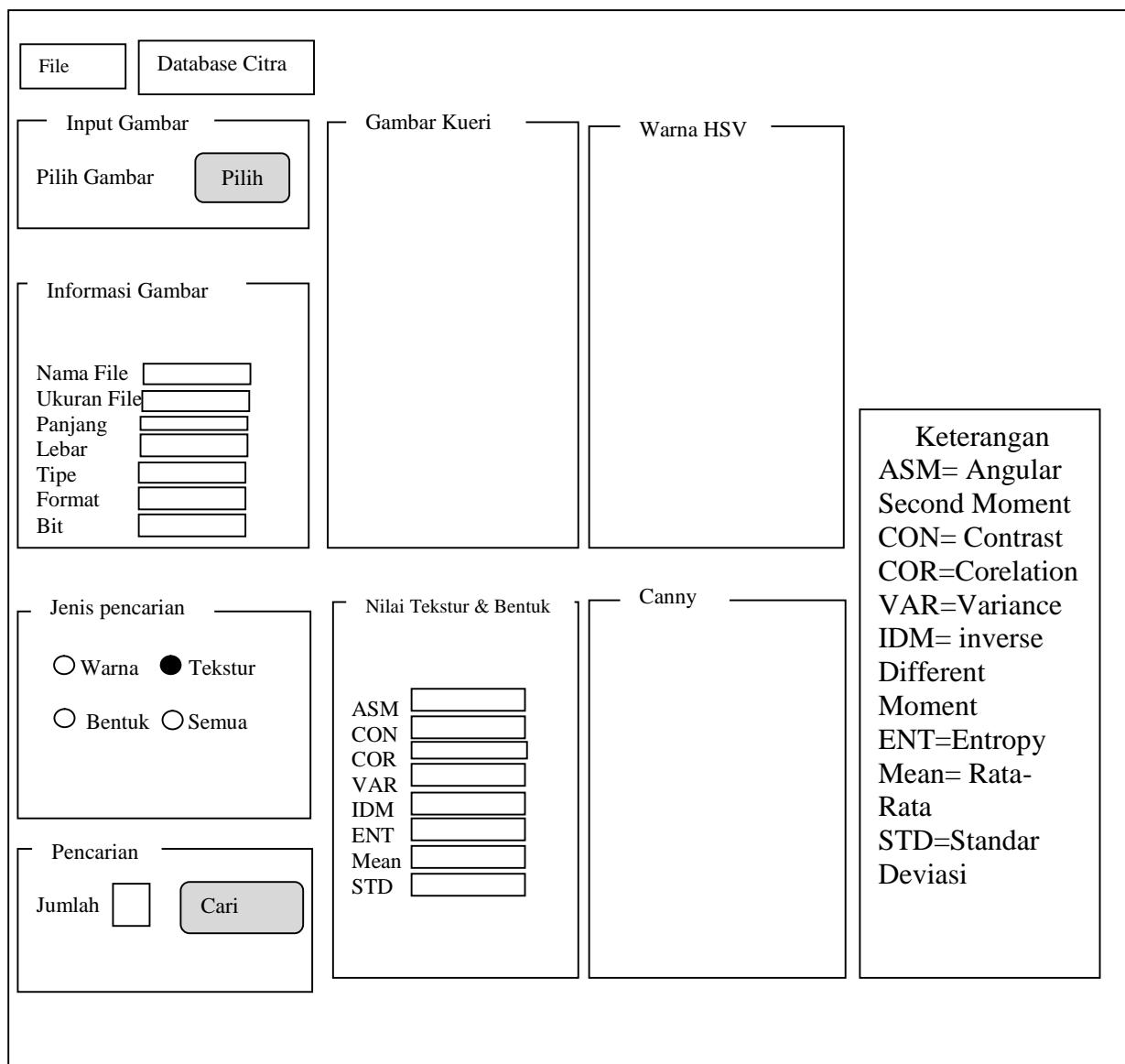
Tekstur.mat digunakan untuk menyimpan nilai ciri tekstur pada gambar kueri. Struktur tekstur.mat terlihat pada tabel 4.28.

Tabel 4.28 Struktur File Tekstur.Mat

No	Nama Variabel	Tipe Data	Keterangan
1	gambar	cell	menyimpan nama file citra
2	ASM	cell	menyimpan nilai <i>angular second moment</i> (tekstur)
3	CON	cell	menyimpan nilai <i>contrast</i> (tekstur)
4	COR	cell	menyimpan nilai <i>correlation</i> (tekstur)
5	VAR	cell	menyimpan nilai <i>varianse</i> (tekstur)
6	IDM	cell	menyimpan nilai <i>inferse different moment</i> (tekstur)
7	ENT	cell	menyimpan nilai <i>entropy</i> (tekstur)

4.8.2 Perancangan Antar Muka Sistem

Perancangan antar muka sistem merupakan tahapan dimana berdasarkan analisa akan di desain sistem yang akan dibangun. Antar muka merupakan alat komunikasi antara *user* dan sistem, agar sistem lebih mudah dan bisa dipergunakan oleh *user*. Rancangan antar muka untuk sistem temu kembali citra ikan hias berdasarkan warna, bentuk dan tekstur terdapat pada gambar.



Gambar 4.34 Rancangan Antar Muka Sistem

Rincian keterangan gambar 4.34 rancangan antar muka sistem temu kembali citra dapat dilihat pada Tabel 4.29.

Tabel 4.29 Keterangan Rancangan Antar Muka Aplikasi

No	Nama	Jenis	Keterangan
1	File	Button	keluar dari aplikasi
2	Database citra	Button	indeks citra <i>database</i>
3	Pilih	Button	Memilih gambar Kueri dari directory
4	Nama File	Text	Menampilkan nama citra
5	Ukuran File	Text	Menampilkan ukuran file citra
6	Panjang	Text	Menampilkan panjang citra
7	Lebar	Text	Menampilkan lebar citra
8	Tipe	Text	Menampilkan tipe citra
9	Format	Text	Menampilkan format citra
10	Bit	Text	Menampilkan bit citra
11	Warna	Radio	Pencarian citra berdasarkan fitur warna
12	Bentuk	Radio	Pencarian citra berdasarkan fitur bentuk
13	Tekstur	Radio	Pencarian citra berdasarkan fitur tekstur
14	Semua	Radio	Pencarian berdasarkan penggabungan fitur
15	Jumlah	Text	Menentukan jumlah hasil pencarian citra
16	Temukan	Button	Proses temu kembali citra.
17	Gambar Kueri	Axes	Menampilkan citra kueri.
18	ASM	Text	Menampilkan nilai ASM citra kueri
19	CON	Text	Menampilkan nilai CON citra kueri
20	COR	Text	Menampilkan nilai COR citra kueri
21	VAR	Text	Menampilkan nilai VAR citra kueri
22	IDM	Text	Menampilkan nilai IDM citra kueri
23	ENT	Text	Menampilkan nilai ANT citra kueri
24	Mean	Text	Menampilkan nilai Mean citra Kueri
25	STD	Text	Menampilkan nilai Standar deviasi citra Kueri
25	Warna HSV	Axes	Menampilkan citra model warna HSV
25	Canny	Axes	Menampilkan citra hasil deteksi tepi canny