PERANCANGAN SISTEM PENDETEKSI GANGGUAN ORGAN JANTUNG MELALUI POLA IRIS MATA

Ryan Arief Mahendra[1], Yeffry Handoko Putra[2], Hidayat[3]
[1, 2, 3]Jurusan Teknik Komputer Unikom, Bandung
[1]ram.thole@gmail.com, [2]yeffry.handoko@gmail.com, [3]hiyan_05@yahoo.com

ABSTRAK

Kesehatan merupakan hal yang sangat penting bagi kehidupan sehari-hari. Hal ini mendorong berbagai metode diagnosa penyakit di bidang kesehatan. Iris mata merupakan representasi dari keadaan kondisi tubuh yang dikenal dengan diagnosa metode Iridologi. Sistem pendeteksi penyakit jantung menggunakan konsep Iridologi ini menggunakan Canny Edge Detection dan Histogram. Metoda deteksi Canny dipilih karena dapat mendeteksi dengan baik sehingga dapat melakukan pemilihan parameter-parameter untuk menandai semua tepi, melokalisasi dengan baik sehingga menghasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli. Perhitungan nilai dari citra yang telah melalu proses penyelarasan menggunakan Histogram dengan turunan Gaussian sebagai pengurang noise. Ketidakakuratan sistem ini dikarenakan faktor pengambilan citra yang akan diolah. Hasil data dari pencocokan citra uji dan latih berupa klasifikasi indikasi penyakit jantung atau jantung normal.

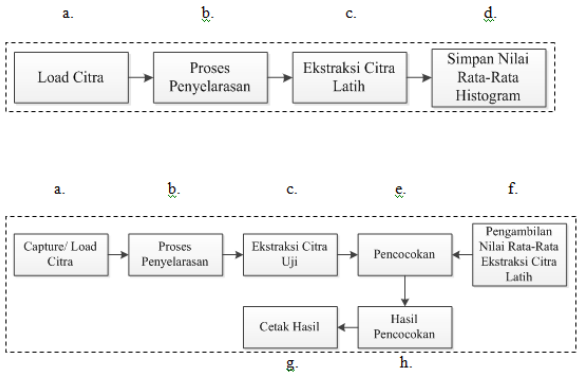
Kata kunci: deteksi, gangguan jantung, iridologi, iris mata.

1. Pendahuluan

Iridologi merupakan teknik analisis berdasarkan tanda-tanda seperti struktur dan warna iris mata,

2. Teori Penunjang

Pada sistem deteksi penyakit jantung ini terdapat dua proses utama yaitu proses pengumpulan data dan proses pencocokan data. Terlihat pada gambar 1.



Gambar 1. Diagram Blok Proses

2.1 Pengolahan Citra

Pengolahan citra merupakan proses yang memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau komputer. Sedangkan definisi citra sendiri adalah gambar pada bidang 2 dimensi atau secara matematis pengertiannya adalah fungsi yang kontinu dari intensitas cahaya pada bidang 2 dimensi.

2.2 Penyelarasan Citra

Merupakan proses yang bertujuan untuk menormalisasi iris dari citra yang didapat pada proses pengambilan citra. Proses ini terdari:

a. Grayscaling
b. Cropping
c. Segmentation
d. Resizing

2.3 Deteksi Tepi Metode Canny

Deteksi tepi Canny dipilih karena dapat mendeteksi dengan baik sehingga dapat melakukan pemilihan parameter-parameter untuk menandai semua tepi, melokalisasi dengan baik sehingga dapat menghasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli, dan respon yang jelas sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya. Pendekatan algoritma Canny dilakukan dengan konvolusi fungsi image

dengan operator Gaussian dan turunan-turunannya.

Konvolusi (*convolution*) adalah sebuah proses dimana citra dimanipulasi dengan menggunakan *eksternal mask*/ *subwindows* untuk menghasilkan citra yang baru. Sedangkan *Filtering* tanpa menggunakan *ekternal mask* tetapi hanya menggunakan *pixel* tetangga untuk mendapatkan *pixel* yang baru.

Operasi konvolusi dilakukan dengan menggeser kernel konvolusi pixel per pixel. Hasil konvolusi disimpan di dalam matriks yang baru.

Misalkan citra *f(x,y)* yang berukuran 5 x 5 dan sebuah *kernel* atau *mask* yang berukuran 3 x 3 masing-masing adalah sebagai berikut:

$$f(x,y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix} \qquad g(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & \bullet 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Gambar 2. Contoh Perhitungan Konvolusi

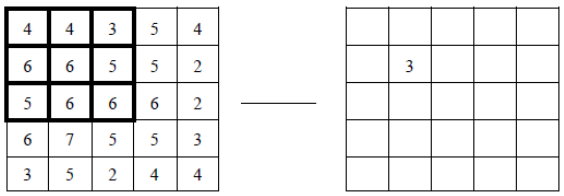
Keterangan: Tanda □ menyatakan posisi (0, 0) dari kernel,

Operasi konvolusi antara citra f(x, y) dengan kernel g(x, y),

f(*x*, *y*) * *g*(*x*, *y*)

dapat diilutrasikan sebagai berikut:

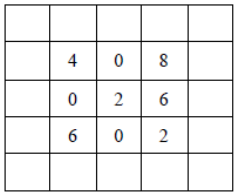
Tempatkan *kernel* pada sudut kiri atas, kemudian hitung nilai *pixel* pada posisi (0, 0) dari *kernel*.



Gambar 3. *Konvolusi pada titik (0,0)* Kernel

Hasil konvolusi = 3. Nilai ini didapatkan dari perhitungan dengan cara berikut: (0 □ 4) + (-1 □ 4) + (0 □ 3) + (-1 □ 6) + (4 □ 6) + (-1 □ 5) + (0 □ 5) + (-1 □ 6) + (0 □ 6) = 3

Dengan cara yang sama seperti di atas, maka *pixel-pixel* pada baris ketiga dikonvolusi sehingga menghasilkan:



Gambar 4. Hasil Konvolusi

Berikut adalah persamaan Gaussian:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-x^2}{2\sigma^2}} \quad (1)$$

Turunan pertamanya:

$$G'(x) = \frac{-x}{\sqrt{2\pi\sigma^3}} e^{\frac{-x^2}{2\sigma^2}} \quad (2)$$

dan turunan keduanya:

$$G''(x) = -\frac{1}{\sqrt{2\pi\sigma^3}} e^{\frac{-x^2}{2\sigma^2}} \left[1 - \frac{x^2}{\sigma^2}\right] \quad (3)$$

Turunan pertama dari fungsi yang dikonvolusikan dengan fungsi Gaussian,

$$g(x,y) = D[Gauss(x,y) * f(x,y)] \quad (4)$$

Ekivalen dengan fungsi citra yang dikonvolusikan dengan turunan pertama daring fungsi Gaussian,

$$g(x,y) = D[Gauss(x,y)] * f(x,y)] \quad (5)$$

2.4 Pencocokan Fitur

Pencocokan fitur adalah tahapan proses yang hanya terdapat pada proses pendeteksian citra mata. Proses ini merupakan proses perbandingan fitur yang telah diekstrak dari citra uji dengan fitur citra latih dari database.. Proses perbandingan ini dilakukan menggunakan metode perhitungan Histogram.

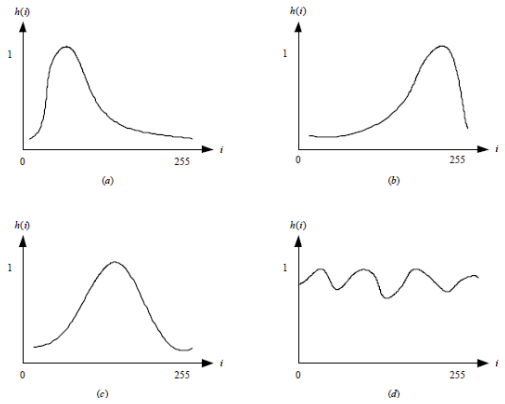
Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra.

Secara matematis histogram citra dapat dihitung dengan rumus:

$$h_i = \frac{n_i}{n}, \qquad i = 0, 1, 2, \ldots, \mathrm{L} - 1 \qquad (6)$$

Yang dalam hal ini,
h_i = jumlah *pixel* yang memiliki derajat keabuan *i*
n = jumlah seluruh *pixel* dalam citra

Histogram citra banyak memberikan informasi penting sebagai berikut:



Gambar 5. (a) Citra Gelap, (b) Citra Terang, (c) Citra Normal High Brightness, (d) Citra High Brightness dan High Contrast

2.5 Perangkat Keras

Ada beberapa perangkat keras (hardware) yang digunakan dalam sistem mendeteksi gangguan organ jantung melalui pola iris mata ini, diantaranya *Webcam*, *Personal Computer*, *Printer*.

2.6 Perangkat Lunak

Perangkat lunak yang digunakan dalam sistem ini yaitu Microsoft Visual Studio 2010 Ultimate sebagai IDE (*Integrated Development Environtment*) dengan menggunakan bahasa C# (*C Sharp*) serta EmguCV sebagai *library*.

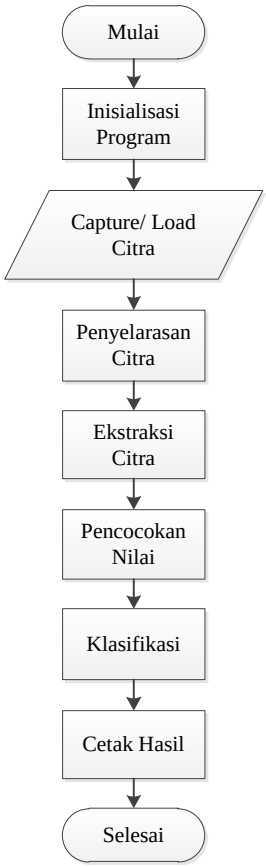
3. Perancangan

Perancangan sistem ini terdiri dari perangkat keras yang digunakan sesuai dengan diagram blok sistem perangkat keras dibawah ini:



Gambar 6. Diagram Blok *Hardware*

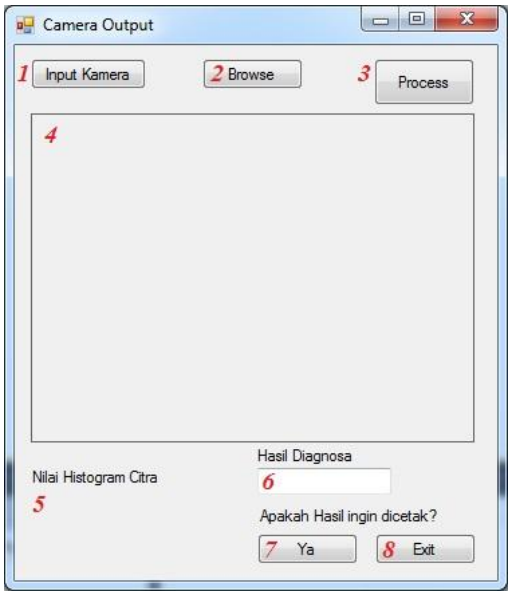
Berikut ini adalah diagram alir yang akan digunakan dalam membuat algoritma program untuk mendukung sistem pendeteksian iris mata:



Gambar 7. Diagram Alir Pencocokan & Klasifikasi Citra

3.1 Antarmuka Sistem

Antarmuka sistem yang dirancang berupa windows form application, didalamnya terdapat button-button dan frame video berikut keterangan masing-masing tombolnya seperti gambar 7. dibawah ini:

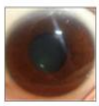
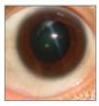


Gambar 8. Prototype Antarmuka Windows Form Application.

4. Hasil Pengujian dan Analisa

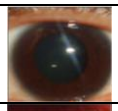
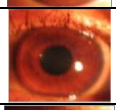
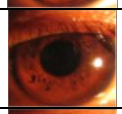
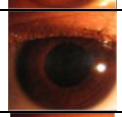
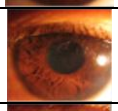
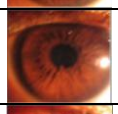
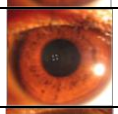
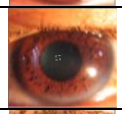
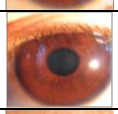
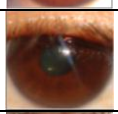
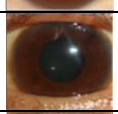
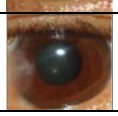
Pada bab ini akan dibahas pengujian dan analisa dari sistem yang telah dibuat secara keseluruhan.
Pada tabel dibawah ini terlihat hasil pengujian citra *cropping* dalam beberapa kondisi jarak pengambilan citra.

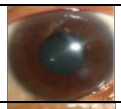
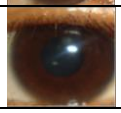
Tabel 1. Pengujian *Cropping* Citra

Kondisi Citra	Citra *Input*	Citra Hasil Cropping	Keterangan
Capture jarak ± 10cm			Hasil *Crop* Berhasil
Capture jarak terlalu jauh			Hasil *Crop* Tidak Tepat
Capture jarak terlalu dekat			Hasil *Crop* Tidak Berhasil

Selanjutnya dilakukan pengujian nilai histogram dari citra. Pengujian ini dilakukan dengan mengambil nilai rata-rata histogram dari 10 sample citra latih yang berpenyakit jantung.

Tabel 2. Sample citra latih

Data ke-n	Citra Latih	Nilai Histogram
1.		3,38
2.		4,62
3.		3,73
4.		2,69
5.		2,55
6.		3,10
7.		2,83
8.		2,71
9.		3,98
10.		4,51
11.		3,03
12.		4,03
13.		2,98
14.		3,04
15.		3,12
16.		2,85
17.		3,83
18.		3,44

19.		4,01
20.		5,23

Pada tabel 1. terlihat 20 sample citra yang memiliki penyakit jantung berikut nilai histogram dari masing-masing citra. Nilai dari histogram dari seluruh sample tersebut kemudian dijumlahkan dan diambil nilai rata-rata sebagain nilai acuan pembanding untuk citra uji.

$$Nilai\ pembanding = \frac{\text{rata} - \text{rata nilai histogram}}{\text{jumlah}}$$

$$Nilai\ pembanding = \frac{69{,}66}{20}$$

Berdasarkan perhitungan diatas didapatkan nilai histogram pembanding yaitu 3,48.

4.1 Pengujian Iris Penyakit Jantung

Pengujian sistem deteksi gangguan jantung melalui pola iris mata dilakukan dengan membandingkan nilai histogram rata-rata citra yang berpenyakit jantung dengan nilai histogram dari citra uji yang berpenyakit jantung.

Tabel 3. Hasil Pengujian dengan 5 contoh Iris Mata penyakit jantung.

Citra Uji ke-n	Keluaran yang diharapkan	Keluaran yang dihasilkan
1.	Indikasi Jantung	Hasil Crop Tidak Tepat
2.	Indikasi Jantung	Hasil Crop Tidak Berhasil
3.	Indikasi Jantung	Indikasi Jantung
4.	Indikasi Jantung	Indikasi Jantung
5.	Indikasi Jantung	Indikasi Jantung
6.	Indikasi Jantung	Jantung Normal
7.	Indikasi Jantung	Indikasi Jantung
8.	Indikasi Jantung	Indikasi Jantung
9.	Indikasi Jantung	Hasil Crop Tidak Berhasil
10.	Indikasi Jantung	Hasil Crop Tidak Berhasil

Pada hasil pengujian yang ditunjukkan tabel 2 terlihat ada kesalahan antara keluaran yang diharapkan dan keluaran yang dihasilkan dari beberapa *sample* citra. Berdasarkan tabel hasil pengujian di atas, dapat dihitung rata-rata keberhasilan yaitu:

$$keberhasilan = \frac{5}{10} \text{ x } 100\%$$

Tingkat keberhasilan 50%

4.2 Pengujian Iris Normal

Pengujian sistem deteksi gangguan jantung melalui pola iris mata dilakukan dengan membandingkan nilai histogram rata-rata citra yang berpenyakit jantung dengan nilai histogram dari citra uji yang tidak berpenyakit jantung.

Tabel 4. Hasil Pengujian dengan 10 contoh Iris Mata tidak berpenyakit jantung.

Citra Uji ke-n	Keluaran yang diharapkan	Keluaran yang dihasilkan
1.	Jantung Normal	Hasil Crop Tidak Tepat
2.	Jantung Normal	Indikasi Jantung
3.	Jantung Normal	Indikasi Jantung
4.	Jantung Normal	Jantung Normal
5.	Jantung Normal	Hasil Crop Tidak Tepat
6.	Jantung Normal	Hasil Crop Tidak Berhasil
7.	Jantung Normal	Jantung Normal
8.	Jantung Normal	Jantung Normal

9.	Jantung Normal	Hasil Crop Tidak Berhasil
10.	Jantung Normal	Jantung Normal

Pada hasil pengujian yang ditunjukkan tabel 3 terlihat ada kesalahan antara keluaran yang diharapkan dan keluaran yang dihasilkan dari beberapa *sample* citra. Berdasarkan tabel hasil pengujian di atas, dapat dihitung rata-rata keberhasilan yaitu:

$$keberhasilan = \frac{4}{10} \text{ x } 100\%$$

Tingkat keberhasilan 40%

5. Kesimpulan dan Saran

Merujuk pada hasil yang didapatkan pada bab pengujian dan analisa, didapatkan kesimpulan sebagai berikut:

1. Sistem mendeteksi gangguan organ jantung melalui pola iris mata dengan menggunakan metode deteksi tepi canny memiliki tingkat keakuratan di bawah 50%.
2. Hasil *cropping* dari citra iris mata sangat mempengaruhi nilai histogram dari citra, oleh sebab itu kondisi pengambilan citra haruslah dilakukan secara tepat yaitu dengan jarak kurang lebih 10cm dari *webcam*.

Saran yang dipertimbangkan untuk mengembangkan sistem ini lebih baik diantaranya:

1. Untuk mendapatkan sistem yang lebih akurat dapat dilakukan dengan menambahkan *sample* citra latih dari iris yang terindikasi memiliki penyakit jantung, dengan banyaknya *sample* tersebut membuat sistem memiliki banyak variabel pembanding yang dapat meningkatkan keakuratan hasil.
2. Diperlukannya citra yang memiliki resolusi tinggi dan pencahayaan pas pada sistem ini agar hasil yang didapat bisa lebih akurat.
3. Perlunya penggunaan metoda pendeteksian lain agar sistem ini memiliki tingkat keakuratan yang baik.

5. Daftar Pustaka

[1] Hiru, D. 2007. *Iridologi Mendeteksi Penyakit Hanya Dengan Mengintip Mata*. Jakarta: Gramedia Pustaka Utama.

[2] Reed, L. 2005. *Stressed Out*. Fort Mill:Nohinsa Books LLC.

[3] Gunawan, K. 2012. *Perancangan Sistem Pengenalan Wajah Berbasis Algoritma Principal Component Analysis (PCA)*. Bandung: Universitas Komputer Indonesia.

[4] Putra, D. 2010. *Pengolahan Citra Digital.* Yogyakarta: Penerbit Andi.

[5] Nassir, S. 2006. *Image Segmentation Based on Watershed and Edge Detection Techniques.* Vol. 3 No. 2: The International Arab Journal of Information Technology.

[6] William, P. X. 2011. How to Convert RGB to Grayscale. Diakses 6 Mei 2014 dari website: http://www.had2know.com/technology/rgb-to-gray-scale-converter.html

[7] Canny, J. 1986. *A Computational Approach To Edge Detection.* IEEE Trans.

[8] Anonim. 2010. *Microsoft Visual Studio*. Diakses 6 Mei 2014 dari website: id.wikipedia.org/wiki/Microsoft_Visual_Studio

[9] Wahana Komputer. 2011. *Microsoft Visual C# 2010*. Yogyakarta: Andi

[10] GNU Free Documentation License 1.2. 2014. *EmguCV Architecture Overview*. Diakses 5 Mei 2014 dari website: www.emgu.com/wiki/index.php/Main_Page