Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM

TEKNOLOGI IRIS SCAN DALAM PROSES IDENTIFIKASI BIOMETRIK

Andreas Eko Kurniawan

Pembimbing I : Nana Juhana, S.T., M.T.
Pembimbing II : Irfan Maliki, S.T.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA

ABSTRAK

Sistem keamanan berbasis biometrik memanfaatkan keunikan karakteristik masing-masing individu. Salah satu yang dapat dipakai adalah sistem keamanan biometrik menggunakan keunikan pola iris mata (*iris recognition*). Sistem biometrik ini melewati empat tahapan proses, yaitu segmentasi, normalisasi, pengkodean, dan pencocokkan. Hasil pengujian pada sistem yang telah dibangun menunjukkan bahwa sistem mampu melakukan pendeteksian citra iris dengan tingkat akurasi 73% dari pengujian terhadap 108 pasangan citra iris dengan ambang batas bawah pencocokkan sebesar 65%. Rata-rata tingkat kecocokkan yang dihasilkan sistem sebesar 68%.

Kata kunci: biometrik, *iris recognition*, segmentasi, normalisasi, pengkodean, pencocokkan

I. PENDAHULUAN

Biometrik (*biometrics*) yang berasal dari bahasa Yunani – *bios* (kehidupan/*life*) dan *metrikos* (ukuran/*measure*) – adalah suatu metode otomatisasi yang digunakan untuk mengidentifikasi seseorang berdasarkan karakteristik lingkungan atau karakteristik biologis orang tersebut. Dasar utama biometrik adalah bahwa tidak ada individu yang memiliki karakteristik yang sama atau memiliki kemungkinan yang sangat kecil untuk sama.

Pengenalan pola iris mata merupakan salah satu metode biometrik yang mempergunakan keunikan pola iris pada masing-masing individu. Iris sendiri adalah satu-satunya organ dalam tubuh yang terlihat dari luar. Iris terletak di belakang kornea dan cairan mata, tetapi di bagian depan lensa.

Citra iris digunakan dengan dasar pemikiran bahwa pola yang dihasilkan oleh citra iris merupakan pola dengan tingkat keacakan yang sangat tinggi (*high degrees of freedom*). Bahkan kemungkinan besar gambaran iris tidak memiliki pola yang pasti sehingga dapat dipergunakan sebagai identitas seseorang.

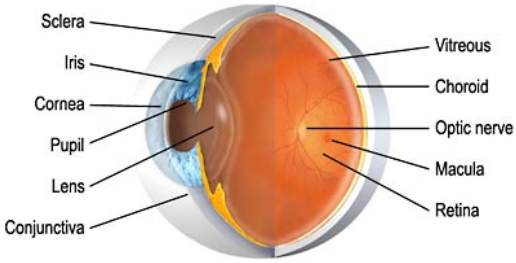
Tujuan penelitian ini adalah merancang dan membangun sebuah perangkat lunak pengenalan citra iris dimana perangkat lunak yang dibangun mampu memberikan keputusan apakah citra iris yang dibandingan merupakan citra iris milik individu yang sama atau bukan dengan keluaran berupa nilai kecocokkan dan status pengenalan.

II. LANDASAN TEORI

2.1 Perkembangan Bentuk dan Pewarnaan Iris

Iris manusia mulai terbentuk selama tiga bulan dalam masa kandungan. Pembentukan struktur iris menjadi pola yang berbeda selesai pada masa delapan bulan masa kandungan, sedangkan proses pewarnaan iris berlanjut hingga bayi mencapai usia satu tahun.

Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM



Gambar 1. Anatomi mata

Pola dan warna yang terbentuk pada iris manusia disebabkan oleh lapisan-lapisan iris dan pigmen melanin. Lapisan-lapisan iris dari belakang ke depan yaitu : lapisan pelindung berwarna gelap, otot mata dan otot melingkar, cairan penghubung, dan lapisan luar dari *chromataphores* dam *melanocytes.*

2.2 *Canny Edge Detection*

Canny edge detection adalah sebuah proses pendeteksian tepi yang mempergunakan dua proses pengolahan citra yaitu pelembutan citra menggunakan *Gaussian filter* dan deteksi tepi menggunakan operator gradien pertama Sobel.

Proses pertama yang dilakukan dalam *Canny edge detection* adalah melakukan proses pelembutan menggunakan *Gaussian filter*. *Gaussian filter* memiliki konsep dasar melawan *noise* menggunakan fungsi lain yang memiliki sebaran Gaussian.

Pendekatan yang digunakan dalam merancang *Gaussian filter* adalah menghitung pembobotan langsung dari distribusi diskrit Gaussian menggunakan rumus:

$$g(x,y) = ce^{\frac{(x^2+y^2)}{2\sigma^2}}$$

dimana c adalah konstanta normalisasi, σ adalah lebar dari fungsi Gaussian yang mana semakin lebar σ, maka semakin kuat efek penghalusan.

Deteksi tepi menggunakan operator Sobel dilakukan dengan memandang bahwa tepi merupakan suatu perubahan intensitas yang besar dalam jarak yang singkat sehingga memiliki nilai kemiringan yang besar. Operator Sobel meninjau pengaturan *pixel* di sekitar *pixel (x, y)*:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator Sobel merupakan magnitudo dari gradien yang dihitung menggunakan persamaan:

$$M = \sqrt{S_x^2 + S_y^2}$$

yang sering disederhanakan karena kerumitan penghitungan akar menjadi:

$$M[f(x,y)] \cong |S_x| + |S_y|$$

yang dalam hal ini, turunan parsial dihitung dengan:

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

dengan konstanta c = 2, sehingga S_x dan S_y dapat dinyatakan sebagai

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

dan

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sedangkan arah tepi dihitung dengan persamaan:

$$\alpha(x,y) = \tan^{-1}\left(\frac{S_y}{S_x}\right)$$

Nilai arah tepi kemudian digolongkan pada empat arah yang mungkin ketika menjelaskan *pixel* tetangga yaitu 0 sampai 22,5 dan 157,5 sampai 180 derajat digolongkan menjadi 0 derajat. Nilai tepi 22,5 sampai 67,5 derajat digolongkan menjadi 45 derajat. Nilai tepi 67,5 sampai 112,5 derajat digolongkan menjadi 90 derajat. Nilai tepi 112,5 sampai 157,5 derajat digolongkan menjadi 135 derajat.

Setelah proses penentuan arah tepi, dilakukan proses penjejakan tepi. Proses ini berguna untuk menghasilkan citra tepi yang tipis. Hal itu dapat dilakukan dengan membandingkan nilai hasil kekuatan tepi pada *pixel* tepi dengan kekuatan tepi *pixel* tetangga. Jika nilai yang didapat lebih besar, maka nilai *pixel* tersebut tidak diubah. Sebaliknya, jika nilai yang didapat lebih kecil, maka *pixel* itu diisi dengan nilai 0.

Proses *thresholding* merupakan proses terakhir pada *Canny edge detection*. Hal ini dilakukan untuk menentukan *pixel* mana yang benar-benar merupakan *pixel* tepi. Proses ini menggunakan dua buah nilai *threshold*. *Threshold* pertama (*T1*) disebut *threshold* tinggi dan *threshold* kedua (*T2*) disebut

Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM

threshold rendah dimana nilai *T1* > *T2*. Nilai *pixel* yang lebih besar dari *T1* dianggap sebagai tepi dan diberi nilai 1. *Pixel* yang memiliki nilai lebih kecil dari *T1* namun lebih besar dari *T2* dan bertetangga dengan *pixel* yang memiliki nilai lebih besar dari *T1* juga dianggap sebagai tepi dan diberi nilai 1. Selain itu, *pixel* tersebut diberi nilai 0.

2.3 Transformasi Hough

Transformasi Hough adalah standar algoritma pada *computer vision* yang digunakan untuk menentukan parameter objek geometri sederhana seperti garis dan lingkaran pada citra. Transformasi Hough pada pengenalan citra iris dapat digunakan untuk mengisolasi lokasi citra iris dari bagian lainnya seperti pupil dan sclera.

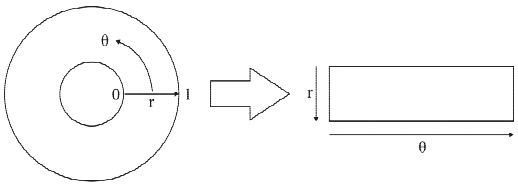
Proses yang dilakukan adalah dengan mengambil parameter lingkaran yang melalui setiap titik pada citra tepi hasil pendeteksian tepi. Parameter ini adalah pusat koordinat x_c dan y_c dan radius *r*, yang mampu untuk mendefinisikan setiap lingkaran yang sesuai dengan rumus

$$x_c^2 + y_c^2 - r^2 = 0$$

Titik maksimum pada ruang hough akan sama dengan radius dan pusat koordinat dari lingkaran terbaik yang didefinisikan oleh titik-titik tepi.

2.4 *Daugman's Rubbers Sheet Model*

Daugman's Rubber Sheet Model digunakan untuk melakukan normalisasi terhadap citra iris yang berbentuk lingkaran menjadi bentuk persegi panjang. Gambar 2 memberikan ilustrasi normalisasi pola iris.



Gambar 2. Ilustrasi Daugman's Rubber Sheet Model

Model tersebut memetakan setiap titik pada area iris menjadi sebuah pasangan koordinat kutub (r,θ) dimana *r* berada pada interval $[0,1]$ dan θ adalah sudut $[0,2\pi]$. Pemetaan tersebut dapat dilakukan menggunakan persamaan:

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta)$$

dengan

$$x(r,\theta) = (1-r)x_p(\theta) + rx_l(\theta)$$
$$y(r,\theta) = (1-r)y_p(\theta) + ry_l(\theta)$$

dimana $I(x,y)$ adalah area citra iris, (x,y) adalah koordinat kartesian, (r,θ) adalah koordinat kutub yang dinormalisasi, dan x_p, y_p dan x_l, y_l adalah koordinat batas antara pupil dan iris sepanjang θ.

2.5 *Gabor Filter*

Gabor Filter pada proses pengolahan citra dipergunakan sebagai sebuah proses yang mengubah sebuah citra menjadi citra diskrit yang memiliki nilai 1 atau 0. Proses ini mampu menggambarkan sebuah citra yang memiliki perbedaan ketinggian atau kedalaman.

Representasi *Gabor filter* pada suatu citra *(x,y)* dapat ditulis sebagai berikut:

$$G(x,y) = e^{-\pi[(x-x_0)^2/\beta^2]} e^{-2\pi i[u_0(x-x_0)+v_0(y-y_0)]}$$

dimana (x_0, y_0) menentukan posisi citra, (α, β) menentukan lebar dan tinggi efektif, dan (u_0, v_0) menentukan modulasi yang memiliki frekuensi $\omega_0 = \sqrt{u_0^2 + v_0^2}$.

Proses demodulasi dan fase penghitungan pada citra menggunakan *Gabor filter* dapat direpresentasikan sebagai:

$$h_{\{\mathrm{Re,Im}\}} = \mathrm{sgn}_{\{\mathrm{Re,Im}\}} \int_{\rho}\int_{\phi} I(\rho,\phi) e^{i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} \rho d\rho d\phi$$

dimana $h_{\{\mathrm{Re,Im}\}}$ dapat dianggap sebagai sebuah nilai bit kompleks dimana komponen nyata dan imajiner bergantung pada tanda dari integral 2D, $I(\rho,\phi)$ adalah citra mentah iris pada dimensi sistem koordinat kutub, dan ω adalah frekuensi gelombang.

2.6 *Hamming Distance*

Hamming Distance merupakan proses perbandingan dua pola yang sangat sensitif terhadap perbedaan nilai bit. Pada kasus pembandingan pola bit *X* dan *Y*, *Hamming Distance* didefinisikan sebagai penjumlahan bit yang tidak sama pada *n* bit pada pola bit. Penjelasan itu dapat ditulis dalam bentuk persamaan:

Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM

$$HD = \frac{1}{N}\sum_{j=1}^{N} X_j (XOR) Y_j$$

Penentuan apakah pola yang dibandingan merupakan pola yang sama dapat dilihat dari nilai HD yang dihasilkan. Semakin kecil nilai HD yang dihasilkan (semakin mendekati 0), semakin sama pola bit yang dibandingkan. Sebaliknya, semakin besar nilai HD yang dihasilkan (semakin mendekati 1), semakin berbeda pola bit yang dibandingkan.

III. ANALISIS DAN PERANCANGAN SISTEM

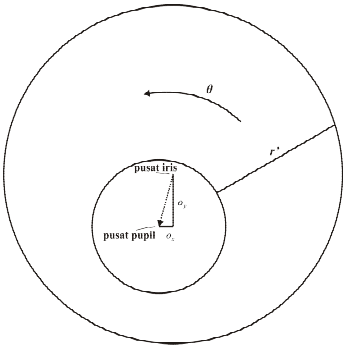
Analisis dan perancangan sistem mengawali proses pembangunan perangkat lunak pengenalan iris. Hal tersebut sangat berguna untuk menjamin perangkat lunak yang dibangun tetap sesuai dengan rencana.

3.1 Analisis Sistem

Proses pengenalan iris mata melewati empat tahapan proses. Tahapan itu adalah segmentasi, normalisasi, pengkodean, dan pembandingan atau pencocokkan. Setiap tahapan merupakan langkah penting untuk proses selanjutnya.

Analisis sistem pengenalan iris mata meliputi:

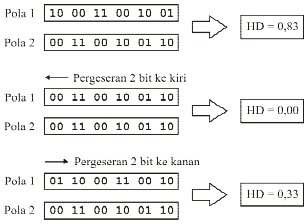
1. Penentuan arah tepi pada tahapan segmentasi dimana persamaan yang digunakan adalah:
   $theta = inv\tan(S_y / S_x)$
   akan menyebabkan masalah jika nilai $S_x = 0$ sedangkan $S_y \neq 0$. Masalah terjadi karena hasil pembagian antara kedua variabel tersebut akan menghasilkan nilai yang tak terhingga. Dalam kasus ini, jika nilai $S_x = 0$, maka arah tepi akan sama dengan 90 derajat jika $S_y \neq 0$ atau 0 derajat jika $S_y = 0$.
2. Citra iris dalam beberapa kasus khusus memiliki lokasi titik pusat lingkaran yang berbeda dengan titik pusat pupil. Hal itu menyebabkan perbedaan jari-jari (*r*), sehingga persamaan pada *Daugman's Rubber Sheet Model* pada proses normalisasi perlu sedikit didefinisikan ulang.



Gambar 3. Normalisasi pada iris dengan pusat berbeda dari pusat pupil

Nilai *r* pada persamaan
$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta)$
didefinisikan sebagai
$r' = \sqrt{\alpha\beta} \pm \sqrt{\alpha\beta^2 - \alpha - r_I^2}$
dengan
$\alpha = o_x^2 + o_y^2$
$\beta = \cos\left(\pi - \arctan\left(\frac{o_y}{o_x}\right) - \theta\right)$
dimana pemindahan pusat pupil relatif terhadap pusat iris diberikan oleh o_x dan o_y. r' adalah jarak antara tepi pupil dan tepi iris pada sudut θ sepanjang area iris, sedangkan r_I adalah radius iris seperti terlihat pada gambar 3.

3. Proses pembandingan atau pencocokkan menggunakan *Hamming Distance* sangat sensitif terhadap perbedaan nilai bit. Oleh karena itu diperlukan sebuah proses pergeseran bit untuk mencegah dua pola iris yang sama hasil normalisasi tidak dapat dikenali akibat perbedaan kondisi pengambilan citra.



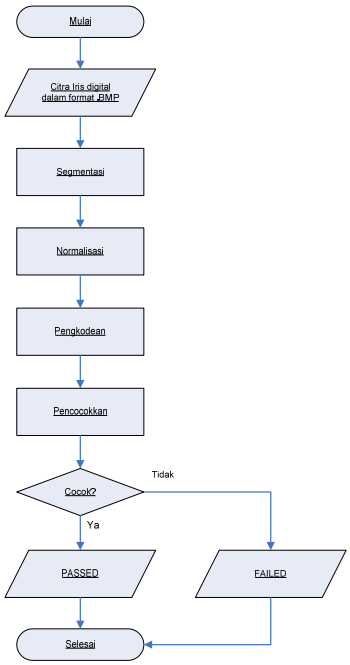
Gambar 4. Ilustrasi proses pergeseran bit

Melalui proses pergeseran bit, nilai *Hamming Distance* dengan nilai terkecil yang kemudian dipakai untuk hasil keluaran sistem.

Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM

3.2 Perancangan Sistem

Empat tahapan proses pengenalan iris seperti segmentasi, normalisasi, pengkodean, dan pembandingan atau pencocokkan dilakukan secara berurutan pada citra iris masukkan.



Gambar 5. Flowchart pengenalan iris

IV. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

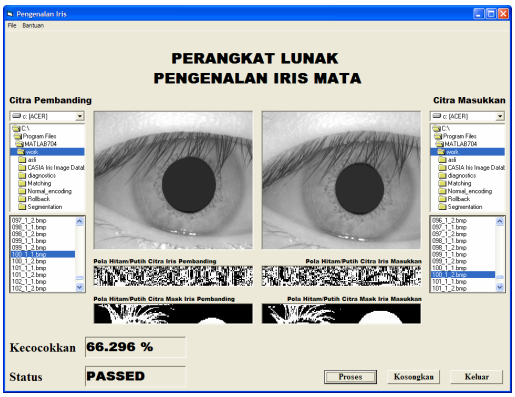
Implementasi perangkat lunak pengenalan pola iris mata dilakukan dengan memanfaatkan keunggulan dua perangkat lunak pemrograman, Visual Basic 6.0 sebagai *tool programming* antar muka dan MATLAB 7.0.4 (R14) sebagai *tool programming* untuk komputasi.

Komunikasi antara kedua bahasa pemrograman dilakukan dengan memanfaatkan *programmatic identifier (ProgID)*. Karena Visual Basic berperan sebagai *tool programming* antar muka, Visual Basic dapat dianggap sebagai *client*, sedangkan MATLAB sebagai *server*. Dengan demikian, Visual Basic membutuhkan *ProgID* MATLAB dimana *ProgID* MATLAB adalah **matlab.application**. Perintah yang digunakan untuk komunikasi adalah:

```
Dim MatLab as Object
Set MatLab = CreateObject("Matlab.Application")
```

Sistem dibangun pada processor berkecepatan 1,7 GHz dengan memori jenis DDR2 sebesar 512 MB.

Gambar 6 memperlihatkan jendela antar muka yang dibangun menggunakan bahasa pemrograman Visual Basic 6.0.

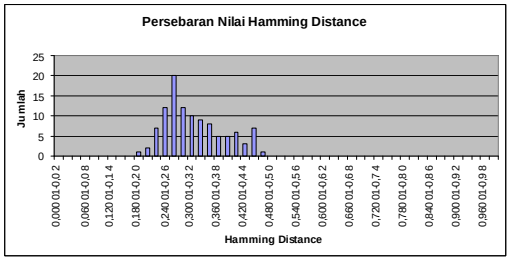


Gambar 6. Antar muka perangkat lunak pengenalan citra iris

4.2 Pengujian

Pengujian dilakukan terhadap 108 pasangan citra iris milik *The Chinese Academy of Sciences – Institute of Automation (CASIA) Iris Image Database* yang telah berbentuk citra digital dalam format .bmp 8-bit (*grayscale*) dan memiliki ukuran 320 × 280 *pixels*.

Gambar 7 memperlihatkan persebaran nilai *Hamming Distance* dari 108 pasangan citra iris. Pengujian juga menghasilkan tingkat akurasi sebesar 73,148% atau 79 pasangan citra iris yang dikenali dari 108 pasangan citra dengan nilai ambang batas kecocokkan sebesar 65%. Nilai rata-rata kecocokkan yang dihasilkan oleh perangkat lunak adalah 68,2985%.



Gambar 7. Persebaran nilai *Hamming Distance* pada 108 citra pasangan iris

Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM

V. KESIMPULAN

5.1 Kesimpulan

Berdasarkan perancangan dan pengujian perangkat lunak yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Perangkat lunak pengenalan citra iris mampu melakukan pengenalan citra iris. Hasil pengujian dengan nilai threshold sebesar 65% menunjukkan tingkat akurasi sebesar 73,148%.
2. Kegagalan dalam pendeteksian lingkaran iris menyebabkan rendahnya nilai HD akibat tidak sesuainya citra iris yang dapat dibandingkan. Kegagalan tersebut disebabkan karena rendahnya perbedaan nilai intensitas keabuan antara iris dan *sclera* (bagian sebelah kiri dan kanan luar lingkaran iris).
3. Proses pengenalan citra iris sangat sensitif terhadap besarnya luas penampang iris yang akan digunakan dalam proses pembandingan, sehingga semakin besar luas penampang iris yang dapat dibandingkan, semakin akurat tingkat kecocokkan pada iris milik individu yang sama.
4. Perbedaan kondisi dan posisi dalam pengambilan citra iris menyebabkan nilai kecocokkan terhadap dua citra iris milik individu yang sama tidak menghasilkan nilai kecocokkan yang sempurna. Hal itu dapat dilihat dari pengujian yang menghasilkan nilai rata-rata kecocokkan sebesar 68,2985%.

5.2 Saran

Berdasarkan penjabaran pada bagian kesimpulan, dapat dirumuskan beberapa hal sebagai saran yang dapat digunakan untuk memperbaiki kinerja dan pengembangan sistem selanjutnya, seperti:

1. Mengimplementasikan algoritma yang lain untuk menemukan algoritma yang lebih cepat dan akurat dalam pengenalan citra iris, seperti *Daugman's Integro-differential Operator* dan *Active Contour Models* pada proses segmentasi; *Image Registration* dan *Virtual Circles* pada proses normalisasi; *Wavelet Encoding*, *Zero-crossing of the 1D wavelet, Haar wavelet*, dan *Laplacian of Gaussian Filter* untuk proses pengkodean.
2. Perangkat lunak pengenalan iris dapat dikembangkan agar mampu mendeteksi citra iris palsu (pola iris yang dicetak pada *contact lens*) dengan memanfaatkan algoritma *2D Fourier Spectrum.*
3. Perangkat lunak dapat dikembangkan sehingga mampu melakukan proses pembandingan secara otomatis dengan mengintegrasikan sebuah *database* yang menyimpan berbagai pola citra iris.
4. Perangkat lunak dapat dikembangkan sehingga mampu melakukan proses *scanning* pergeseran citra untuk mendapatkan posisi titik pusat iris yang sama pada citra pembanding dan citra masukkan.

DAFTAR PUSTAKA

[1] Ahmad, Usman. Pengolahan Citra Digital dan Teknik Pemrogramannya. Yogyakarta: Graha Ilmu, 2005.

[2] *Canny Edge Detection*, http://fourier.eng.hmc.edu.

[3] *Canny Edge Detection*, http://www.pages.drexel.edu.

[4] Duane Hanselman dan Bruce Littlefield. MATLAB Bahasa Komputasi Teknis. Yogyakarta: Andi, 2001.

[5] Halvorson, Michael. *Microsoft Visual Basic 6.0 Professional Step by Step*. Jakarta: Elex Media Komputindo, 2002.

[6] *Iris Recognition: Gabor Filtering*, http://cnx.org/content/m12493/latest/.

[7] J. Daugman. *Iris Recognition*, http://www.cl.cam.ac.uk/~jgd1000/.

[8] J. Daugman. *How Iris Recognition Work*. The Computer Laboratory, University of Cambridge, http://www.CL.cam.ac.uk/users/jgd1000/.

[9] Jain, Anil. *Biometrics.* Microsoft Encarta 2006. Redmond, WA: Microsoft Corporation, 2005.

[10] Masek, Libor. *Recognition of Human Iris Patterns for Biometrics Identification*. A Dissertation. School of Computer Science and Software Engineering, The University of Western Australia, 1998.

Copyright 2006 pada Andreas Eko, Teknik Informatika UNIKOM

[11] *MATLAB Automation Server Support*, MATLAB Help.

[12] Munir, Rinaldi. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika, 2004.

[13] P. Kovesi. *MATLAB Function for Computer Vision and Image Analysis*, http://www.cs.uwa.edu.au/~pk/Research/MatlabFns/index.html.

[14] Pressman, Roger S. Rekayasa Perangkat Lunak, Pendekatan Praktisi Buku Satu. Yogyakarta: Andi, 2002.

[15] *The Chinese Academy of Sciences – Institute of Automation (CASIA) Iris Image Database*, http://www. sinobiometrics.com