

## ***AUTOMATED FINGERPRINT IDENTIFICATION SYSTEM DENGAN TEKNIK MINUTIAE-BASED***

Heppy Oktianatasari, Wayan Firdaus Mahmudy, Bayu Rahayudi  
Program Studi Ilmu Komputer  
Fakultas MIPA, Universitas Brawijaya, Malang

---

### **ABSTRACT**

*The one known fingerprint recognition system is Automated Fingerprint Identification System (AFIS). AFIS's input is fingerprint and the output is people identity that has it. The last module from identification process is fingerprint matching. The one method has many used is minutiae based, because it has high accuracy and fast computational time. This system consist 2 main processes, such as registration and identification. The first at each process is did preprocessing which consist of segmentation, low pass filtering, local equalization and binarization. Than, do ridge thinning and minutiae detection. After that, at registration process minutiae data saved in database and at identification process minutiae are compared with all of data in database. The result shows that, the minutiae numbers are not consistent, and the system has accuracy 88.00% with standard deviation 17.25%. It's happen because image characteristic is deferent each other, false minutiae detection, and minutiae position has changed.*

**Keywords:** *fingerprint, AFIS, Minutiae-based*

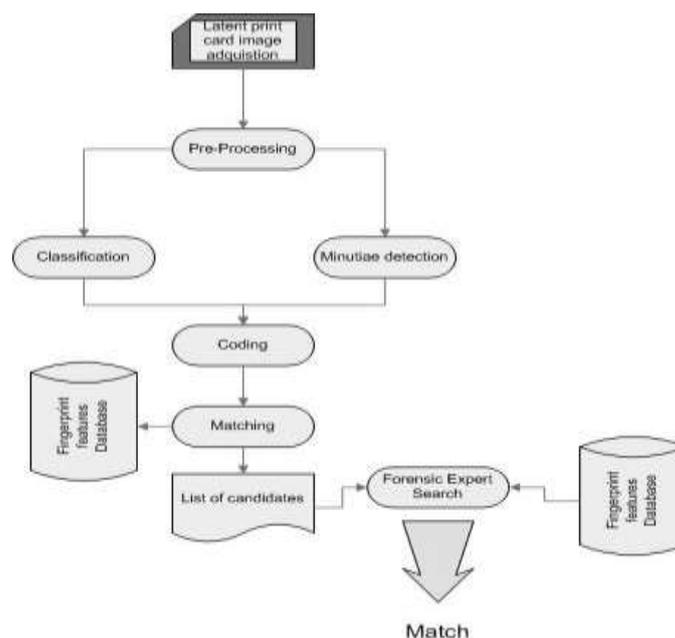
---

### **1. Pendahuluan**

*Fingerprint* atau sidik jari adalah penanda khas yang dimiliki oleh setiap individu, sehingga dapat digunakan sebagai pembeda antara individu satu dengan lainnya (Widianto, 2006). Dengan bertambah canggihnya komputer, sistem otomatis telah berhasil mengembangkan klasifikasi manual dan metode pengenalan sidik jari dan salah satunya adalah *Automated Fingerprint Identification System (AFIS)*. Input AFIS berupa sidik jari, sedangkan outputnya berupa identitas orang yang mempunyai sidik jari tersebut. (Jain, dkk., 1999). AFIS pada dua dekade terakhir ini telah digunakan dengan luas dalam bidang pengenalan individu dan akhirnya menjadi sangat relevan dengan aplikasi kependudukan karena sistem pengenalan sidik jari menawarkan sejumlah kelebihan untuk menambah kualitas dan kegunaan dari identifikasi. Proses utama dari identifikasi adalah perbandingan 2 buah sidik jari. Teknik perbandingan *Minutiae-based* mempunyai akurasi tinggi dan waktu komputasi yang cepat (Sharma, dkk., 2004).

## 2. AFIS

AFIS terdiri dari empat komponen. Pertama yaitu suatu proses otomatisasi yang mengeliminasi untuk kebutuhan pengklasifikasian untuk meletakkan sidik jari dalam sebuah file dan membandingkan dua sidik jari (A). Kedua pencarian database yang terdiri dari gambar sidik jari (F) yang diperoleh dari individu dengan menggunakan kartu sidik jari atau dengan menggunakan alat elektronik yang hampir sama dengan sebuah scanner. Ketiga aspek identifikasi (I) yang terjadi ketika sidik jari seseorang discan yang menghasilkan gambar yang dicari perbandingannya dengan gambar sidik jari dalam database. Keempat AFIS merupakan sistem (S) karena menggunakan komputer dan software dan dapat berinteraksi dengan subsistem dan sistem identifikasi yang lain, termasuk juga sistem AFIS yang lain (Komarinski, 2005). Gambar 1 merupakan struktur dari AFIS.



Gambar 1 Struktur AFIS

### 2.1 Pre-processing

Proses ini bertujuan untuk menghilangkan area yang tidak dibutuhkan pada sidik jari. Proses yang dilakukan adalah perbaikan citra untuk mempertegas struktur *ridge* dan *valley* sidik jari dan mengeliminasi derau dengan cara memfilter citra tersebut. Hasil dari *pre-processing* adalah citra biner atau *gray-scale* yang telah diperbaiki.

#### 2.1.1 Segmentasi

Segmentasi citra adalah proses yang bertujuan untuk memisahkan antara daerah *foreground* dengan daerah *background* (Raymond Thai, 2003). *Foreground* adalah daerah yang terdapat *ridge* dan *valley*. Sedangkan *background* adalah daerah di luar daerah *foreground*. Daerah *background* biasanya mempunyai nilai varian yang rendah sedangkan daerah *foreground* mempunyai varian yang tinggi. Oleh karena itu, dapat dimanfaatkan metode pengembangan varian (*varian thresholding*) untuk proses ini. Pertama citra dibagi dalam beberapa blok, kemudian dihitung variannya untuk masing-masing blok. Apabila nilai varian lebih kecil dari nilai *global thresholding* maka daerah tersebut ditandai sebagai daerah *background* dan sebaliknya jika nilai varian lebih besar sama dengan *global thresholding*

maka daerah tersebut ditandai sebagai daerah *foreground*. Varian untuk masing-masing blok dapat dihitung dengan persamaan 1

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} (I(i, j) - M(k))^2 \quad (1)$$

di mana  $V(k)$  adalah varian pada blok  $k$ ,  $I(i, j)$  adalah citra pada pixel  $i, j$  dan  $M(k)$  adalah rata-rata (*mean*) pada blok  $k$ .

### 2.1.2 Perhitungan Orientasi Lokal

Orientasi dari sidik jari mendefinisikan sudut dari *ridge* lokal yang ada pada sidik jari. Berikut ini merupakan langkah untuk menghitung orientasi titik  $i, j$  (Julasayvake dan Choomchuay, 2007).

- Citra dibagi dalam blok ukuran  $w \times w$  dengan pixel  $i, j$  sebagai titik pusat.
- Pada masing-masing pixel dihitung gradien  $\partial_x(i, j)$  dengan menggunakan operator sobel horisontal dan operator sobel vertikal untuk  $\partial_y(i, j)$ . Gambar 2 merupakan operator sobel horisontal dan vertikal

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

Operator Sobel horisontal    Operator Sobel Vertikal

Gambar 2 Operator sobel horisontal dan vertikal

- Orientasi lokal dapat dihitung dengan menggunakan persamaan 4

$$V_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\partial_x(u, v)\partial_y(u, v) \quad (2)$$

$$V_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} \partial_x^2(u, v)\partial_y^2(u, v) \quad (3)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left( \frac{V_y(i, j)}{V_x(i, j)} \right) \quad (4)$$

$V_x(i, j)$  dan  $V_y(i, j)$  adalah orientasi field lokal. Dan  $\theta(i, j)$  adalah orientasi lokal citra

### 2.1.3 Pendeteksian Titik Tengah

Titik tengah dari suatu obyek adalah perkiraan tengah dari massa atau *centroid* dari objek (Fosu, dkk., 2008). Jika obyek dalam citra dinotasikan dengan  $B(x, y)$ , maka titik tengah dapat didefinisikan sebagai 2 dimensional ( $B_{10}$ ,  $B_{01}$ ) seperti persamaan 5 dan 6.

$$B_{10} = \sum \sum xB(x, y) / B_{00} \quad (5)$$

$$B_{01} = \sum \sum yB(x, y) / B_{00} \quad (6)$$

$xB(x, y)$  dan  $yB(x, y)$  adalah axis dan ordinat dari citra.  $B_{00}$  adalah total intensitas citra.

### 2.1.4 Low Pass Filtering

Gambar yang diambil dari alat / sensor dapat mengalami gangguan dan interferensi dari banyak sumber, termasuk didalamnya gangguan sensor listrik dan bintik yang bukan merupakan bagian dari gambar. Pengaruh seperti ini dapat dikurangi dengan menggunakan *filter* pada gambar yang ada.

Gangguan pada gambar diakibatkan sensor/ saluran transmisi biasanya tampak sebagai titik tetangganya. Konsep ini dipakai sebagai dasar algoritma pembersihan gangguan. Gangguan pada gambar akan mempunyai spektrum frekuensi spasial yang tinggi dari komponen gambar yang normal, yang diakibatkan tidak terhubungnya secara spasial. *Low*

*pass filter* secara efektif dapat mengurangi gangguan pada gambar masukan F dengan ukuran  $l \times l$ , dengan matrik konvolusi H berukuran  $n \times m$ , dapat diturunkan fungsi pada persamaan 7.

$$y(k1, k2) = \sum_{i=1}^n \sum_{j=1}^m f(k1+i, k2+j)h(i, j) \quad (7)$$

### 2.1.5 Local Equalization Histogram

Ide dasar dari *local histogram equalization* adalah dibutuhkankannya perbaikan citra secara mendetil dalam area yang kecil. Karena terkadang ada sejumlah kecil titik yang berpengaruh perubahan perhitungan global. Sehingga dikembangkan suatu fungsi perubahan yang berdasarkan pada level distribusi titik tetangga pada masing-masing titik. Ide ini berdasarkan pendefinisian daerah tetangga dan pemindahan titik tengah dari titik ke titik. Dengan kata lain pada masing-masing lokasi *histogram* dari titik dihitung. Langkah dari *local histogram equalization* adalah sebagai berikut:

Untuk setiap titik dalam citra lakukan :

- Mensetting area lokal
- Menghitung frekuensi citra pada area lokal
- Menghitung warna akhir titik dengan persamaan  $f(a) = (2^b - 1).P(a)$

### 2.1.6 Binerisasi

Binerisasi adalah proses mengubah citra dari *keabuan-scale* menjadi citra biner atau dua warna (Raymond Thai, 2003). Proses ini diawali dengan pengecekan nilai titik tiap citra. Apabila nilai titik lebih kecil dari toleransi maka nilai titik akan diubah menjadi 0 (hitam) dan sebaliknya jika lebih besar sama dengan toleransi maka akan diubah menjadi 1 (putih)

### 2.1.7 Penipisan Citra

Algoritma *hilditch* merupakan algoritma yang dibangun untuk tujuan *skeletonization*. *Skeletonization* adalah proses untuk menipiskan pola dari titik yang sangat banyak sebisa mungkin tanpa berpengaruh pada bentuk citra yang asli. Dengan kata lain setelah titik ditipiskan, pola seharusnya masih bisa dikenali. Sebuah *skeleton* terdiri dari properti: setipis mungkin, terhubung, dan nilai tengah. Ketika properti ini telah terpenuhi maka algoritma harus dihentikan. Gambar 2.3 merupakan contoh dari pola dan *skeleton*-nya. (Azar, 2007)

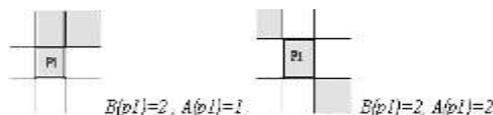
P9	P2	P3
P8	P1	P4
P7	P6	P5

Gambar 3 Delapan tetangga

Mengacu pada Gambar 3 ada 8-tetangga dari titik p1. Jika p1 akan dijadikan sebagai hasil dari *skeleton*, maka 8 tetangga dari p1 diatur searah jarum jam dengan 2 fungsi:

$B(p1)$  = jumlah dari tetangga p1 yang tidak kosong, dan

$A(p1)$  = jumlah dari pola 0,1 pada urutan p2, p3, p4, p5, p6, p7, p8, p9, p2



Gambar 4. Fungsi B(p1) dan A(p1)

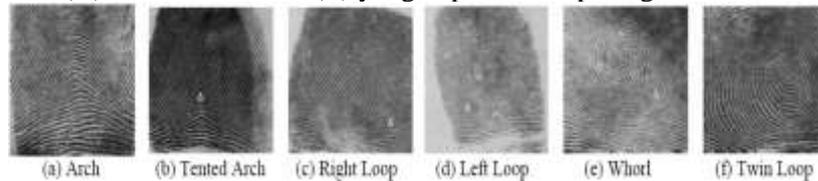
Ada dua versi dari algoritma *Hilditch* yang satu menggunakan kotak 4x4 dan yang lain 3x3. Algoritma *Hilditch* terdiri dari beberapa bentuk yang melewati pola dan masing-masing jalur. Algoritma ini mengecek semua titik dan memutuskan untuk mengubah suatu titik dari warna hitam ke warna putih jika diperlukan sesuai dengan kondisi berikut:

- $2 \leq B(p1) \leq 6$
- $A(p1) = 1$
- $p2.p4.p8 = 0$  or  $A(p2) \neq 1$
- $p2.p4.p6 = 0$  or  $A(p4) \neq 1$

Dan akan berhenti jika tidak ada yang berubah (tidak ada titik yang dapat dipindah)

## 2.2 Klasifikasi Sidik Jari

Henry mengelompokkan sidik jari dalam 5 kelompok yaitu: *Right Loop*(R), *Left Loop*(L), *Whorl*(W), *Arch* (A), dan *Tented Arch*(T) yang dapat dilihat pada gambar 4.



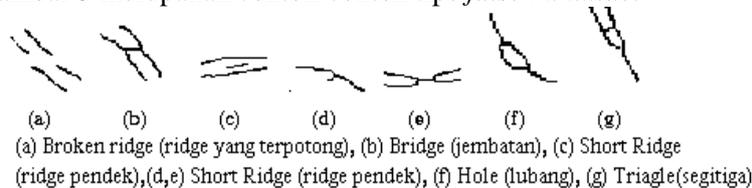
Gambar 4 Pengelompokan sidik jari menurut Henry

Pengklasifikasian pada AFIS berguna untuk menyeleksi data dari basis data sehingga hanya data yang mempunyai tipe klasifikasi sama dengan data yang akan dikenali saja yang akan dibandingkan. Oleh karena itu, diharapkan dengan adanya proses ini akan mempercepat waktu pengenalan.

## 2.3 Pendeteksian Minutiae

*Minutiae* adalah bagian dari *ridge* yang terletak pada titik paling awal atau paling akhir, perpotongan pada dua *ridge* ataupun bentuk *ridge* sendiri yang merupakan titik tunggal. Berbagai macam prosedur dapat digunakan untuk mendeteksi poin dari *minutiae* pada sebuah citra. Dan umumnya proses ini sudah masuk dalam *pre-processing* suatu citra sidik jari. Pendekatan yang sering digunakan, meliputi penipisan dari *ridge* atau sering disebut juga sebagai *skeletonization*. Proses ini membutuhkan ekstensif *pre-processing* yang memerlukan waktu yang cukup lama.

Masalah utama yang sering muncul dalam proses ini adalah adanya *minutiae* yang salah (*false minutiae*). Pada dasarnya *minutiae* yang benar hanya dibatasi pada 2 tipe yaitu *ridge ending* dan *ridge bifurcation* dan tipe lainnya dianggap sebagai *false minutiae*. Banyak penelitian menekankan pendeteksian *false minutiae* yang dibatasi pada kriteria jarak. (Kim, dkk., 2006). Gambar 5 merupakan contoh-contoh tipe *false minutiae*.



Gambar 5 Contoh-contoh tipe *false minutiae*

## 2.4 Perbandingan Sidik Jari

Teknik perbandingan sidik jari dapat dikelompokkan ke dalam 2 kategori: *minutiae-based* dan *correlation-based*. Pada teknik *minutiae-based* yang dibandingkan adalah poin *minutiae* yang mengandung informasi mengenai lokasi, jenis, jumlah *ridge* dan kualitas area di sekitar *minutiae*. Sedangkan pada metode *correlation-based*, *template* dan *query* gambar sidik jari dikorelasikan untuk mencari derajat kesamaannya. Masing-masing metode mempunyai kekurangan dan kelebihan. Untuk mengurangi kekurangan tersebut maka dikembangkan metode perbandingan dengan menggunakan lebih dari dua metode dasar seperti *FingerCode*-

based (gabungan metode perbandingan dengan *ridge* dan *minutiae-based*) dan *Local-Correlation-based* (gabungan metode *correlation-based* dan *minutiae-based*).

#### 2.4.1 Perbandingan *Minutiae-based*

Jika dimisalkan “T” sebagai gambar yang akan dibandingkan dan “Q” sebagai gambar dari queri. Maka  $Q(\Delta x, \Delta y, \theta)$  adalah presentasi gambar pada queri yang dirotasi oleh sudut  $\theta$  sebesar  $\Delta x, \Delta y$  pada arah x dan y secara berturut-turut. Yang dapat dinotasikan sebagai persamaan 2.11

$$S(T, Q) = \max_{\Delta x, \Delta y, \theta} CC(T, Q^{(\Delta x, \Delta y, \theta)}) \quad (8)$$

Dengan menggunakan teorema tersebut maka domain frekuensinya dapat dihitung dengan persamaan 2.12

$$CC(T, Q) = F^{-1}(F^*(T)F(Q)) \quad (9)$$

Pada T, Q mempresentasikan vektor elemen dari *minutiae* yang mempunyai atribut posisi, orientasi, tipe, tetangga dari *minutiae*, dll. Umumnya *minutiae* m menggunakan 3 atribut  $\{x, y, \theta\}$  yang mengindikasikan x, y adalah letak koordinat *minutiae* dan  $\theta$  adalah sudut dari *minutiae*

$$T = \{m_1, m_2, \dots, m_n\}, m_i = \{x_i, y_i, \theta_i\}, i = 1, 2, \dots, n \quad (10)$$

$$Q = \{m_1, m_2, \dots, m_n\}, m_i = \{x_i, y_i, \theta_i\}, i = 1, 2, \dots, n \quad (11)$$

Jika  $m_i \in T$  dan  $m_j \in Q$  dibandingkan, dan jika jarak antara keduanya ( $sd$ ) lebih kecil dari toleransi jarak ( $r_0$ ) dan selisih arah keduanya ( $dd$ ) lebih kecil dari toleransi sudut ( $\theta_0$ ) maka dapat dinotasikan dalam persamaan 2.14 dan 2.15.

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0 \quad (12)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360 - |\theta'_j - \theta_i|) \leq \theta_0 \quad (13)$$

Jika  $m''_j$  adalah hasil fungsi pemetaan map dari  $m'_j$  berdasarkan suatu transformasi geometri terhadap perpindahan  $[\Delta x, \Delta y]$  dengan arah rotasi searah jarum jam maka:

$$map_{\Delta x, \Delta y, \theta}(m'_j) = m''_j = \{x'_j, y'_j, \theta'_j + \theta\} \quad (14)$$

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (15)$$

Jika  $\zeta$  adalah fungsi indikator yang menghasilkan nilai 1 apabila  $m_i$  dan  $m''_j$  cocok maka:

$$\zeta(m''_j, m_i) = \begin{cases} 1 & \text{Jika } sd \leq r_0 \text{ dan } dd \leq \theta_0 \\ 0 & \text{selain itu} \end{cases} \quad (16)$$

Sehingga perbandingan ini dapat dirumuskan sebagai berikut:

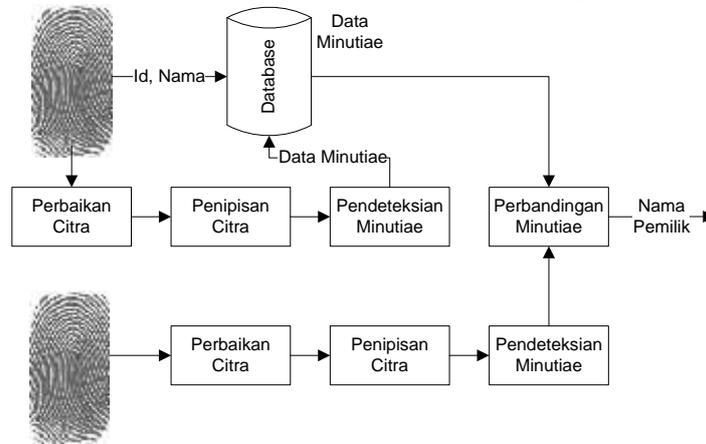
$$\max_{\Delta x, \Delta y, \theta, P} imize = \sum_{i=1}^n \zeta(map_{\Delta x, \Delta y, \theta}(m'_{P(i)}), m_i) \quad (17)$$

Notasi  $\Delta x, \Delta y, \theta$  pada persamaan 15, 16 dan 17 disebut dengan parameter posisi (*alignment*). Persamaan 2.19 dapat digunakan jika  $\Delta x, \Delta y, \theta$  diketahui. Pada kenyataannya nilai  $\Delta x, \Delta y, \theta$  sangat sulit ditentukan, dan dapat ditentukan dengan nilai-nilai penaksiran. Dalam pengembangannya nilai  $\Delta x, \Delta y, \theta$  diselesaikan dengan pendekatan *brute force* (Barral, dkk., 2004)

### 3. Perancangan Perangkat Lunak

AFIS yang akan dibuat merupakan sistem yang akan membaca file gambar sidik jari yang

sudah tersimpan dalam komputer (*off line*). Proses pada sistem ini terbagi menjadi 2 yaitu yang pertama pengisian data master pada basis data (registrasi data pemilik sidik jari) dan yang kedua adalah pengenalan sidik jari (identifikasi sidik jari). Proses pengisian data master pada basis data merupakan proses penyimpanan data pemilik sidik jari dan gambar sidik jarinya, sedangkan proses pengenalan sidik jari merupakan proses pengenalan pemilik sidik jari dari gambar sidik jari yang dimasukkan. Proses-proses tersebut akan diterangkan lebih rinci pada sub bab selanjutnya. Alur dari sistem ini dapat dilihat pada Gambar 6.



Gambar 6. Alur Sistem

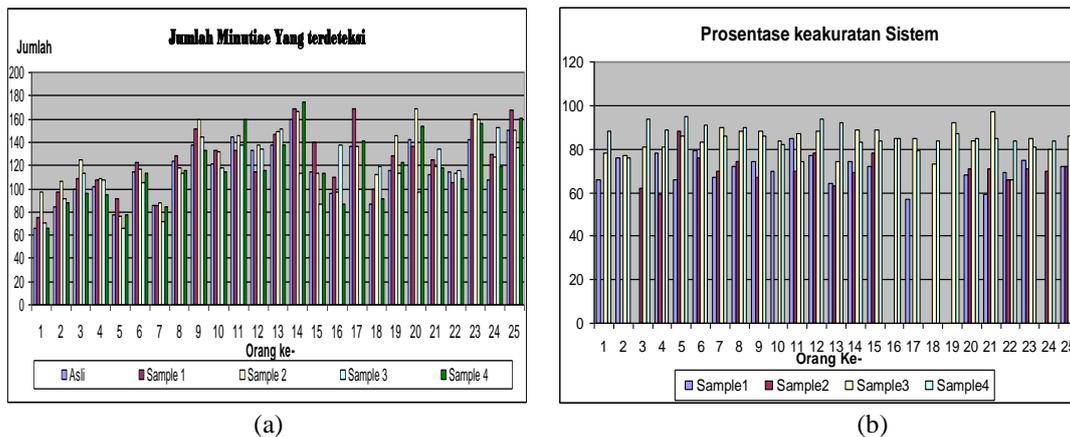
Untuk proses perbaikan citra, proses dibagi menjadi 5 tahapan yaitu: segmentasi, low pass filtering, perhitungan orientasi lokal, ekualisasi histogram lokal, dan binerisasi.

Uji coba dilakukan menggunakan 25 citra sidik jari, pada setiap sidik jari memiliki 5 sampel yang berbeda. Sampel ini terdiri dari 1 sampel diperoleh dari citra yang diambil dari sensor sidik jari U.are.U 4000B (citra asli) dan 4 sampel yang diperoleh dari memproses citra asli. Sampel citra asli akan digunakan sebagai citra yang dideteksi *minutiae*-nya dan hasilnya disimpan dalam basis data (master data dalam proses registrasi). Sedangkan 4 sampel lainnya digunakan sebagai citra yang akan diidentifikasi dengan basis data. Keempat citra sampel ini terdiri dari : 2 sampel yang diperoleh dari citra asli yang diputar, 1 sampel yang diperoleh dari citra asli yang diceraikan warnanya, dan 1 sampel yang diperoleh dari citra asli yang digeser.

#### 4. Uji Coba

Pengujian sistem dilakukan dengan memberikan nilai *global thresholding* =100 pada proses Segmentasi, faktor pengali =1/9 pada proses *Low Pass Filter*, lebar lokal area=10 pada proses ekulisasi lokal *histogram*, batas ambang 128 pada proses binerisasi, maksimal jarak *false minutiae* = 10, dan minS = 55 pada proses perbandingan *minutiae*. Nilai-nilai tersebut diperoleh dari beberapa percobaan yang dilakukan sebelumnya.

Gambar 7 (b) apabila dihitung rata-rata, nilai tengah dan standart deviasinya, menghasilkan tabel 2. Pada gambar 7 (a) jumlah minutiae yang terdeteksi menunjukkan jumlah yang tidak konsisten. Dan pada gambar 7 (b) menunjukkan jumlah prosentase keakuratan sistem kurang maksimal yaitu mempunyai rata-rata rata-rata nilai kesesuaian 43.92 untuk sampel 1, 33.04 untuk sampel 2, 84.96 untuk sampel 3 dan 85.76 untuk sampel 4.



(a) Jumlah minutiae yang terdeteksi (b) Prosentase keakuratan sistem  
 Gambar 7 Hasil uji coba sistem

Tabel 1. Hasil Perhitungan *Mean*, *Median* dan Standart Deviasi

	sampel				% Keakuratan
	1	2	3	4	
Rata-Rata	43.92	33.04	84.96	85.76	88.00
Standart deviasi	55.61	62.13	5.79	5.25	17.85
Median	69.00	69.00	85.00	85.00	100.00
FAR	12 / 100 x 100 =				12.00

## 5. Kesimpulan

1. Proses perbaikan citra pada sistem yang dibuat kurang maksimal sehingga berpengaruh pada proses selanjutnya. Ketidakmaksimalan ini dapat dilihat dari jumlah *minutiae* yang dideteksi pada kelima sampel menunjukkan nilai yang tidak konsisten. Ketidakmaksimalan proses ini disebabkan karena setiap citra memiliki karakteristik bervariasi sehingga pola *ridge* untuk citra bervariasi. Pada proses *low pass filtering* seharusnya faktor pengali dan *mask* diset berbeda untuk setiap citra yang disesuaikan dengan karakteristik pola *ridge*.
2. Selain akibat sifat pola *ridge*, ketidakkonsistenan jumlah *minutiae* diakibatkan oleh pendeteksian *false minutiae*, yang seharusnya tidak dideteksi sebagai *minutiae*. Pendeteksian *false minutiae* dalam sistem yang dibuat hanya menggunakan kriteria jarak *minutiae* dan jumlah tetangganya. Seharusnya kriteria ditambahkan dengan pencarian alur *ridge* sehingga dapat dideteksi *false minutiae* berbentuk lubang, jembatan dan segitiga. Selain itu juga bisa ditambahkan kriteria apabila *minutiae* yang terdeteksi berada atau merupakan dalam tetangga blok area *background* maka *minutiae* tersebut juga merupakan *false minutiae*.
3. Proses perbandingan citra yang dibuat menghasilkan keakuratan sebesar 88.00% dengan rata-rata nilai kesesuaian 43.92 untuk sampel 1, 33.04 untuk sampel 2, 84.96 untuk sampel 3 dan 85.76 untuk sampel 4. Penyebab dari kurang maksimalnya nilai rata-rata kesesuaian adalah posisi *minutiae* sidik jari yang berubah-ubah pada citra yang menyebabkan koordinat berubah, sehingga diperlukan posisi berdasarkan jarak titik *core* dengan koordinat *minutia*. Dan diperlukan juga penyesuaian posisi citra (*alignment*) sebelum poin-poin *minutiae*-nya dibandingkan

## DAFTAR PUSTAKA

- Anonim. 2007. *Low-Pass Filtering*. [http://homepages.inf.ed.ac.uk/rbf/HIPR2/Low-Pass\\_Filtering.htm](http://homepages.inf.ed.ac.uk/rbf/HIPR2/Low-Pass_Filtering.htm) diakses tanggal 24 September 2007

- Azar, Danielle. *Hilditch's Algorithm for Skeletonization* .  
<http://jeff.cs.mcgill.ca/%7Egodfried/teaching/projects97/azar/skeleton.html> diakses tanggal 24 September 2007
- Barral, Claude, Sebastian Coron dan David Naccache. 2004. *Externalized Fingerprint Matching*. <http://scompt.com/wordpress/wp-content/uploads/2007/02/021.pdf> diakses tanggal 25 Oktober 2007
- Fosu, C., G W Hein dan B Eissfeller. *DETERMINATION OF CENTROID OF CCD STAR IMAGES*. <http://iec.cugb.edu.cn/WorldComp2006/341.pdf> diakses tanggal 28 April 2008
- Jain, L.C., U. Halici, I. Hayashi, S.B Lee dan S Tsusui. 1999. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*. USA: CRC Press
- Julasayvake, Atipat dan Somsak Choomchuay. 2007. *An Algorithm for Fingerprint Core Point Detection*. Bangkok: King Mongkut's Institute of Technology Ladkrabang.
- Komarinski, Peter. 2005. *Automated Identification Fingerprint System*. California: Elsevier Academic Press
- Murni, Aniati dan Dina Chaahyati. 2005. *Pengolahan Citra Digital : Peningkatan Mutu Citra Pada Domain Spasial*. Jakarta : Universitas Indonesia.
- Sharma, Chandan, Mihir Mukerjii dkk. 2004. *DSP Implementation of Fingerprint- Based BiometricSystem*. Faculty of Engineering: University of Auckland.
- Thai, Raymond. *Fingerprint Image Enhancement and Minutiae Detection*. Australia: The University of Western Australia. 2003.
- Widianto, Yoga. 2006. *Perbandingan Pola Sidik Jari Antara Etnis Jawa, Cina, dan Arab di kota Malang*. . Malang: Universitras Brawijaya .