

Pengenalan Sidik Jari Menggunakan Algoritma Pencocokan Adaptif Berdasarkan Penjajaran *Minutiae*

Tri Legawa*, Achmad Hidayatno**, Imam Santoso**
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstract

In fingerprint matching contain several method that can used for fingerprint recognition system. One of them is minutiae based matching. This method utilising number and location of the minutiae that different on every body so that it could be distinguishing between one person and the other. However, in this method contain two main problem that is image deformation and inexact minutiae localisation.

In this final project is made a fingerprint recognition program based on minutiae matching. Program is made in three main step, that are image enhancement, minutiae extraction, and minutiae matching. Image enhancement consist of segmentation, normalisation, orientation estimation, ridge frequency estimation, filtering, binerization, and thinning process. Minutiae extraction consist of minutiae detection, eliminating false minutiae, and unification of ending and bifurcation. Minutiae matching consist of alignment step and matching step. In minutiae matching process a boundary with two types that is fixed size and adaptive boundary are used as the reference to determines a pair of minutiae is match or not. Determining of two fingerprint images is match or not based on percentage of number minutiae match from those images. Testing is done using 165 fingerprint images from eleven respondents.

Testing result shows average of percentage of recognize program that using fixed size boundary is 67,88% with the maximum value is 100% and minimum is 40%, whereas the program that using adaptive boundary has average of percentage of recognize as big as 92,12% with maximum value is 100% and minimum value is 86,67%. Thereby, usage of adaptive boundary make the system more robust to deformation and inexact localisation problem.

Keywords: *minutiae, boundary, alignment, minutiae matching*

I. Pendahuluan

1.1 Latar Belakang

Dewasa ini sistem verifikasi maupun identifikasi semakin banyak digunakan untuk berbagai aplikasi mengacu pada kebutuhan akan keamanan. Meskipun begitu, metode konvensional masih banyak digunakan secara luas. Umumnya metode konvensional dibagi menjadi dua kategori yaitu sesuatu yang diketahui dan sesuatu yang dimiliki. Sesuatu yang diketahui misalnya pin dan *password*, sedangkan sesuatu yang dimiliki misalnya kunci dan kartu. Metode ini memiliki kelemahan misalkan dengan kehilangan atau lupa yang dapat terjadi sewaktu-waktu. Oleh karena itu sekarang banyak dikembangkan sistem yang mengacu pada konsep biometri.

Biometri merupakan dasar identifikasi yang menggunakan karakteristik alami manusia. Salah satu karakteristik manusia yang bersifat khas dan dapat digunakan untuk membedakan antara orang yang satu dengan lainnya adalah sidik jari. Sidik jari memiliki pola-pola yang khas yang disebut sebagai *minutiae* atau titik minusi. Jumlah dan letak *minutiae* pada setiap orang berbeda-beda sehingga dapat dijadikan pembeda antara orang yang satu dengan yang lain.

Pada tugas akhir ini dibuat sistem pengenalan sidik jari berdasarkan penjajaran pola-pola *minutiae*. Metode ini terdiri dari dua tahap yaitu penjajaran dan pencocokan *minutiae*. Prinsipnya, *minutiae* antara dua citra sidik jari disejajarkan untuk kemudian dicocokkan dengan algoritma pencocokan adaptif. Pencocokan yang adaptif merujuk pada penggunaan suatu pembatas (*boundary*) pada tiap pencocokan sepasang *minutiae*

dengan ukuran yang berubah-ubah menyesuaikan radius *minutiae* dari *minutiae* referensi yang ditetapkan.

1.2 Tujuan

Tujuan tugas akhir ini adalah:

- 1) Membuat suatu program aplikasi untuk mengenali sidik jari manusia menggunakan algoritma pencocokan *minutiae*.
- 2) Meneliti tingkat pengenalan algoritma pencocokan *minutiae* yang digunakan dalam mengenali citra sidik jari yang diujikan.

1.3 Batasan Masalah

Batasan-batasan masalah dalam tugas akhir ini antara lain:

- 1) *Minutiae* yang digunakan untuk pengenalan adalah titik akhir (*termination*) dan percabangan (*bifurcation*).
- 2) Citra sidik jari diambil dari ibu jari tangan kanan tiap responden.
- 3) Saat pengakuisisian data, jari dalam keadaan bersih dan normal.
- 4) Citra yang digunakan sebagai data uji maupun data acuan memiliki ukuran yang sama yaitu 144x144 piksel dengan tipe skala keabuan.
- 5) Proses akuisisi data tidak terhubung langsung dengan sistem.

II. Landasan Teori

2.1 Klasifikasi Sidik Jari

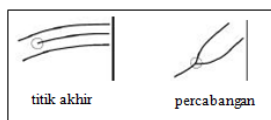
Klasifikasi sidik jari adalah membagi data pola garis alur bukit (*ridge*) sidik jari kedalam kelompok-

* Mahasiswa Teknik Elektro Universitas Diponegoro

** Dosen Teknik Elektro Universitas Diponegoro

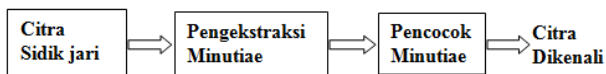
kelompok kelas ciri yang menjadi karakteristik sidik jari tersebut yaitu untuk mempercepat proses identifikasi. Ada dua jenis kategori sidik jari yaitu kategori bersifat umum (global) dan kategori yang bersifat khusus (lokal) yaitu untuk menggambarkan ciri-ciri khusus individual, seperti jumlah *minutiae*, jumlah dan posisi inti (*core*), dan jumlah dan posisi delta.

Untuk sistem pengenalan sidik jari, *minutiae* dapat digunakan sebagai parameter kesamaan dari dua buah sidik jari. *Minutiae* didefinisikan sebagai titik-titik akhir (*ending*) dan titik-titik awal percabangan (*bifurcation*) dari garis-garis alur yang memberikan informasi yang unik dari suatu sidik jari. Jumlah titik akhir, jumlah percabangan, dan posisinya masing-masing menjadikan suatu sidik jari memiliki ciri khusus yang membedakan antara sidik jari yang satu dengan lainnya. Dengan demikian *minutiae* dapat dijadikan dasar penentuan dua buah sidik jari sama atau berlainan.



Gambar 1 Pola *minutiae* [1]

Sistem pengenalan sidik jari dapat disederhanakan menjadi tiga bagian utama yaitu sensor untuk pengambilan data, pengekstraksi *minutiae*, dan pencocok *minutiae*.



Gambar 2 Penyederhanaan sistem pengenalan sidik jari berbasis *minutiae* [8]

Pengekstraksi *minutiae* berfungsi untuk mengekstraksi *minutiae* dari sidik jari yang bersangkutan sehingga diperoleh jumlah *minutiae* yang terdiri dari dua jenis yaitu titik akhir dan percabangan tiga berikut indeks masing-masing *minutiae* pada bidang gambar.

Pencocok *minutiae* berfungsi untuk mencocokkan keseluruhan *minutiae* dari dua buah sidik jari yang dibandingkan. Hasil pencocokan berupa nilai derajat kesamaan dari dua sidik jari tersebut yang kemudian dijadikan dasar penentuan apakah kedua sidik jari tersebut identik atau tidak.

2.2 Tapis Gabor

Tapis gabor dua dimensi terdiri dari ranah gelombang sinusoidal dari arah dan frekuensi yang dimodulasi dengan sampul Gaussian. Tapis Gabor simetri-genap adalah bagian nyata dari fungsi Gabor, yang diberikan dengan gelombang cosinus yang dimodulasi dengan Gaussian. Tapis Gabor simetri genap didefinisikan sebagai berikut. [8]

$$G(x, y; \varphi, f) = \exp \left\{ -\frac{1}{2} \left[\frac{x_{\varphi}^2}{\sigma_x^2} + \frac{y_{\varphi}^2}{\sigma_y^2} \right] \right\} \cos(2\pi f x_{\varphi}) \dots\dots\dots(1)$$

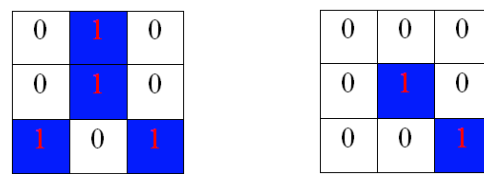
$$x_{\varphi} = x \cos\varphi + y \sin\varphi \dots\dots\dots(2)$$

$$y_{\varphi} = -x \sin\varphi + y \cos\varphi \dots\dots\dots(3)$$

φ adalah orientasi tapis Gabor, f adalah frekuensi gelombang cosinus, σ_x dan σ_y adalah standar deviasi dari nilai Gaussian sepanjang sumbu x dan y , dan x_{φ} dan y_{φ} menyatakan sumbu x dan y pada kerangka koordinat tapis.

2.3 Ekstraksi Minutiae

Konsep *Crossing Number* (CN) banyak digunakan untuk ekstraksi *minutiae*. Pada umumnya, untuk masing-masing jendela 3x3, jika piksel sentral adalah 1 dan memiliki tiga piksel tetangga yang bernilai 1, maka piksel sentral merupakan sebuah percabangan. Jika piksel sentral adalah 1 dan hanya memiliki satu piksel tetangga yang bernilai 1, maka piksel sentral merupakan sebuah titik akhir.



(a). Percabangan (b). Titik akhir
Gambar 3 Konsep *Crossing Number* [8]

Jarak rata-rata antar garis alur (D) juga dihitung pada tahap ini. Jarak rata-rata antar garis alur adalah rata-rata jarak antara dua garis alur yang berdekatan atau bertetangga. Nilai D dihitung dengan persamaan 4 berikut. [8]

$$D = \frac{\sum_{i=1}^m \frac{L_i}{b_i} + \sum_{j=1}^n \frac{L_j}{k_j}}{m+n} \dots\dots\dots(4)$$

dengan m dan n adalah jumlah baris dan kolom citra yang sudah ditipiskan polanya. L_i dan b_i adalah panjang baris ke i dan jumlah piksel bernilai 1 pada baris ke i . L_j dan k_j adalah panjang kolom ke j dan jumlah piksel bernilai 1 pada kolom ke j .

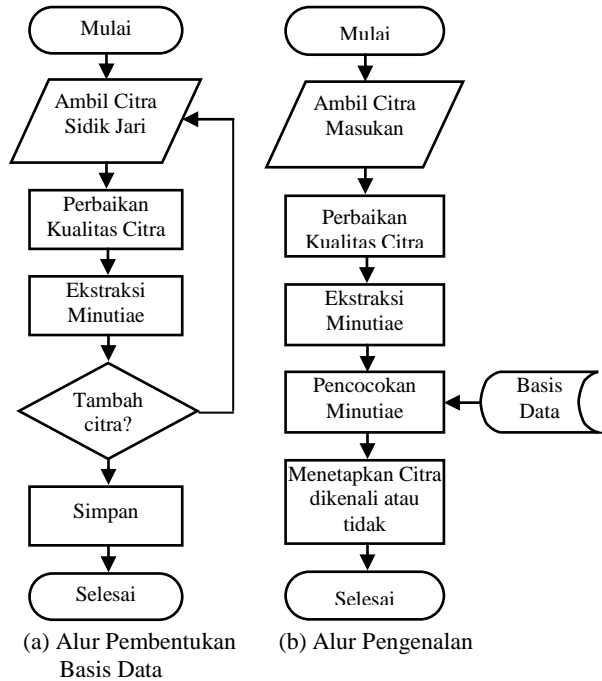
Masing-masing *minutiae* dikarakteristikan dengan parameter koordinat x , koordinat y , dan orientasi. Penghitungan orientasi untuk percabangan dilakukan secara khusus. Ketiga garis alur yang terhubung ke titik percabangan memiliki arah masing-masing. Percabangan dibagi menjadi tiga titik akhir. Tiga titik akhir yang baru adalah tiga piksel tetangga dari titik percabangan dan masing-masingnya bagian dari garis alur yang terhubung ke percabangan sebelum dipisahkan.

Suatu bagian dari garis alur ditelusuri mulai dari titik awal yaitu titik akhiran dan panjangnya sama dengan D . Seluruh koordinat x pada titik-titik di bagian garis alur dijumlahkan. Hasil penjumlahan kemudian dibagi dengan D untuk mendapatkan s_x . Kemudian s_y dicari dengan cara yang sama. Arah masing-masing titik akhir diperoleh sebagai berikut: [8]

$$\theta = \tan^{-1} \frac{(s_y - t_y)}{(s_x - t_x)} \dots\dots\dots(5)$$

III. Perancangan Program

Secara umum pembuatan program mengikuti alur sesuai yang ditunjukkan dalam gambar 4 berikut:



Gambar 4 Alur Perancangan Program Pengenalan Sidik Jari

3.1 Citra Masukan

Citra masukan adalah citra sidik jari hasil pemindaian sidik jari responden yang terlebih dahulu disimpan. Alat pemindai yang digunakan adalah *Billionton Fingerprint Security System* yang menghasilkan citra sidik jari 144x144 piksel dengan format bitmap.

3.2 Perbaikan Kualitas Citra

Perbaikan kualitas citra dilakukan dengan proses penapisan menggunakan tapis Gabor. Tapis Gabor digunakan pada citra sidik jari dengan mengkonvolusikan citra dengan tapis. Konvolusi piksel (i,j) pada citra memerlukan nilai orientasi O(i,j) dan nilai frekuensi F(i,j) garis alur dari piksel tersebut. Dengan demikian, penggunaan tapis Gabor G untuk mendapatkan citra yang lebih baik dapat didefinisikan sebagai berikut:^[8]

$$E(i, j) = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} G(u, v; O(i, j), F(i, j))N(i - u, j - v) \dots\dots\dots(6)$$

dengan O adalah orientasi citra, N adalah citra ternormalisasi, F adalah frekuensi alur bukit citra, dan w_x dan w_y adalah lebar dan tinggi penapis dari tapis Gabor.

Lebar bidang penapis, yang menentukan interval frekuensi yang ditanggapi oleh tapis, didefinisikan dengan standar deviasi σ_x dan σ_y . Nilai σ_x dan σ_y dapat dinyatakan dengan:^[8]

$$\sigma_x = k_x F(i, j) \dots\dots\dots(7)$$

$$\sigma_y = k_y F(i, j) \dots\dots\dots(8)$$

dengan F adalah frekuensi garis alur citra, k_x adalah konstanta untuk σ_x , dan k_y adalah konstanta untuk σ_y . Ukuran tapis dapat dinyatakan sebagai fungsi dari standar deviasi sebagai berikut:^[8]

$$w_x = 6\sigma_x \dots\dots\dots(9)$$

$$w_y = 6\sigma_y \dots\dots\dots(10)$$

dengan w_x dan w_y adalah lebar dan tinggi penapis (*mask*) dari tapis Gabor, dan σ_x dan σ_y adalah standar deviasi sampel Gaussian sepanjang sumbu x dan y. Nilai lebar dan tinggi tapis ditetapkan 6σ karena informasi berada diantara daerah $[-3\sigma, 3\sigma]$ menjauh dari poros y.

Setelah proses penapisan dilakukan binerisasi dan penipisan pola agar *minutiae* pada citra dapat diekstraksi. Penggunaan tapis Gabor menjadikan citra keluaran tapis yaitu E(i,j) memiliki rerata nilai piksel sama dengan nol. Jadi binerisasi dapat dilakukan dengan mengambil nilai ambang sama dengan nol. Kemudian pola alur bukit ditipiskan hingga hanya memiliki ketebalan satu piksel saja.

3.3 Ekstraksi Minutiae

Setelah citra diperbaiki kualitasnya, maka ekstraksi *minutiae* dapat dilakukan. Proses ekstraksi dimulai dengan proses penandaan *minutiae*.

Penandaan *minutiae* dilakukan dengan cara membagi citra menjadi blok-blok citra berukuran 3x3 piksel. Pendeteksian *minutiae* baik itu titik akhir ataupun percabangan dilakukan menggunakan konsep crossing number.

Setelah *minutiae* dideteksi, titik-titik *minutiae* palsu dihilangkan. Nilai rerata jarak antar alur bukit (D) dihitung terlebih dahulu selanjutnya dihitung jarak antara titik-akhir dengan percabangan, jarak titik akhir dengan titik akhir, dan jarak percabangan dengan percabangan. Apabila jarak antara dua *minutiae* kurang dari nilai D maka kedua *minutiae* adalah *minutiae* palsu dan dihilangkan. ROI (Region of Interest) ditambahkan untuk mengakomodasi daerah yang diperlukan saja untuk diproses pada citra sidik jari sehingga *minutiae* diluar ROI akan dianggap tidak ada.

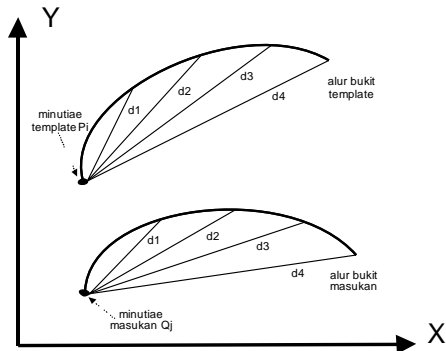
Langkah terakhir dari ekstraksi *minutiae* yaitu penyatuan titik akhir dan percabangan untuk mempermudah proses pencocokan *minutiae*. Tiap percabangan diubah menjadi tiga titik akhir dengan titik akhirnya adalah masing-masing piksel tetangganya yang bernilai 1. Orientasi tiap titik akhir dicari dengan mencuplik titik-titik pada alur bukit yang berhubungan dengan titik akhir tersebut sebanyak D. Orientasinya adalah dihitung sesuai persamaan 5. Dengan demikian pada proses ekstraksi *minutiae* ini dihasilkan satu set titik akhir dengan parameter koordinat x, koordinat y dan orientasi.

3.4 Pencocokan Minutiae

Pada proses pencocokan ini, *minutiae* pada citra masukan dicocokkan dengan *minutiae* pada tiap citra di data acuan. Proses pencocokan meliputi dua tahap yaitu tahap penjarangan dan tahap pencocokan.

3.4.1 Tahap Penjajaran

Pada tahap penjajaran, *minutiae* pada citra masukan disejajarkan dengan *minutiae* pada tiap citra *template* (citra di data acuan).



Gambar 5 Pengambilan titik-titik pada garis alur bukit *minutiae*^[6]

Lebih jelasnya, anggap R^d dan R^D sebagai kumpulan garis alur bukit (*ridge*) yang terhubung dengan *minutiae* pada citra masukan dan citra *template*. Algoritma penjajaran adalah sebagai berikut:

- 1) Untuk masing-masing garis alur anggota R^d , tampilkan dalam bentuk diskrit satu dimensi dan cocokkan dengan masing-masing garis alur anggota R^D menurut rumus sebagai berikut:^[6]

$$\Delta d = \frac{1}{L} \sum_{i=0}^L |d_i^d - d_i^D| \quad \dots\dots(11)$$

$$\Delta \delta = \frac{1}{L} \sum_{i=0}^L |\delta_i^d - \delta_i^D| \quad \dots\dots(12)$$

dengan L adalah jumlah titik pada dua garis alur, d_i^d adalah jarak dari titik i pada alur bukit R^d ke *minutiae* Q_j dan δ_i^d adalah sudut antara garis yang menghubungkan titik i dengan *minutiae* P_i dan orientasi dari *minutiae* P_i . d_i^D dan δ_i^D memiliki pengertian yang sama, tetapi untuk *minutiae* P_i . Jika nilai Δd dan $\Delta \delta$ lebih kecil dari nilai ambang yang ditentukan, maka kedua *minutiae* dapat dikatakan identik dan dilanjutkan ke langkah kedua, sebaliknya bila melebihi nilai ambang yang ditentukan maka kedua *minutiae* dianggap tidak identik dan dilanjutkan pencocokan dengan *minutiae* berikutnya.

- 2) Memperkirakan transformasi antara dua garis alur bukit. Vektor translasi $(\Delta x, \Delta y)^T$ antara dua garis alur yang disamakan dihitung dengan:^[2]

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x^d \\ y^d \end{pmatrix} - \begin{pmatrix} x^D \\ y^D \end{pmatrix} \quad \dots\dots(13)$$

$(x^d, y^d)^T$ dan $(x^D, y^D)^T$ adalah koordinat x dan y dari dua *minutiae*, yang terhubung dengan garis alur d dan D secara berurutan. Sudut rotasi $\Delta \theta$ antara dua garis alur dengan persamaan 14 berikut.^[2]

$$\Delta \theta = \frac{1}{L} \sum_{i=0}^L (\gamma_i - \Gamma_i) \quad \dots\dots(14)$$

dengan L adalah panjang minimal dari dua garis alur d dan D , γ_i dan Γ_i adalah sudut radian dari titik ke i pada garis alur terhadap *minutiae* referensi yang terhubung dengan dua garis alur d dan D secara

berurutan. Faktor skala antara citra masukan dan citra *template* dianggap sama dengan satu.

- 3) Menyatakan $(x^D, y^D, \theta^D)^T$ berdasarkan parameter transformasi yang telah ditentukan, sebagai *minutiae* referensi. Translasikan dan rotasikan keseluruhan N *minutiae* masukan terhadap *minutiae* referensi, sesuai dengan persamaan 15.^[2]

$$\begin{pmatrix} x_i^A \\ y_i^A \\ \theta_i^A \end{pmatrix} = \begin{pmatrix} \cos(\Gamma_i - \Delta\theta) & -\sin(\Gamma_i - \Delta\theta) & 0 \\ \sin(\Gamma_i - \Delta\theta) & \cos(\Gamma_i - \Delta\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i + \Delta x - x^D \\ y_i + \Delta y - y^D \\ \theta_i - \Delta\theta \end{pmatrix} \quad \dots\dots(15)$$

dengan $(x_i, y_i, \theta_i)^T$, ($i = 1, 2, \dots, N$), adalah *minutiae* masukan dan $(x_i^A, y_i^A, \theta_i^A)^T$ adalah *minutiae* yang telah disejajarkan.

3.4.2 Tahap Pencocokan

Diberikan $P = ((x_1^P, y_1^P, \theta_1^P)^T, \dots, (x_M^P, y_M^P, \theta_M^P)^T)$ menyatakan *minutiae* sebanyak M pada citra di data acuan dan $Q = ((x_1^Q, y_1^Q, \theta_1^Q)^T, \dots, (x_N^Q, y_N^Q, \theta_N^Q)^T)$ menyatakan kumpulan *minutiae* sebanyak N pada citra masukan yang disejajarkan dengan M menurut *minutiae* referensi yang ditentukan. Algoritma yang digunakan adalah sebagai berikut:

- 1) Mengubah masing-masing titik *minutiae* ke sistem koordinat polar mengacu pada *minutiae* referensi yang diperoleh pada proses penjajaran.^[4]

$$\begin{pmatrix} r_i \\ e_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i^* - x^{ref})^2 + (y_i^* - y^{ref})^2} \\ \tan^{-1} \left(\frac{y_i^* - y^{ref}}{x_i^* - x^{ref}} \right) \\ \theta_i^* \end{pmatrix} \quad \dots\dots(16)$$

$(x_i^*, y_i^*, \theta_i^*)^T$ adalah koordinat dari *minutiae* terjajar, $(x^{ref}, y^{ref}, \theta^{ref})^T$ adalah koordinat dari *minutiae* referensi, dan $(r_i, e_i, \theta_i)^T$ adalah indeks *minutiae* pada sistem koordinat polar (r_i menyatakan jarak radian, e_i menyatakan sudut radian dan θ_i menyatakan orientasi *minutiae* mengacu pada *minutiae* referensi).

- 2) Menyatakan *minutiae* masukan dan *template* dalam koordinat polar sebagai untai simbolik dengan menggabungkan masing-masing *minutiae* menurut urutan besar sudut radian.^[4]

$$P_p = ((r_1^P, e_1^P, \theta_1^P)^T, \dots, (r_M^P, e_M^P, \theta_M^P)^T) \quad \dots\dots(17)$$

$$Q_p = ((r_1^Q, e_1^Q, \theta_1^Q)^T, \dots, (r_N^Q, e_N^Q, \theta_N^Q)^T) \quad \dots\dots(18)$$

dengan $(r_i^P, e_i^P, \theta_i^P)$ dan $(r_i^Q, e_i^Q, \theta_i^Q)$ merupakan radius, sudut radian, dan orientasi *minutiae* ternormalisasi mengacu pada *minutiae* referensi.

- 3) Mencocokkan untai P_p dan Q_p yang dihasilkan dengan algoritma pencocokan untai.
- 4) Memilih jumlah *minutiae* cocok terbesar untuk menentukan nilai kecocokan. Hitung nilai kecocokannya.

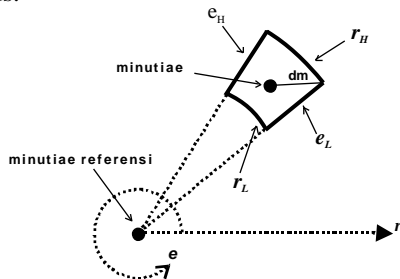
Beberapa algoritma pencocokan untaian sudah dikembangkan. Biasanya, pencocokan untaian dilakukan dengan menempatkan suatu pembatas yang mengitari *minutiae*. Apabila ada sepasang *minutiae* berada di dalam pembatas maka kedua *minutiae* dapat dikatakan sama atau cocok. Penggunaan pembatas dapat menghasilkan toleransi kesalahan tertentu.

Pembatas yang berukuran tetap (fixed sized boundary) memiliki ukuran sudut radian dan radius tetap. Ukuran pembatas dinyatakan sebagai berikut.^[6]

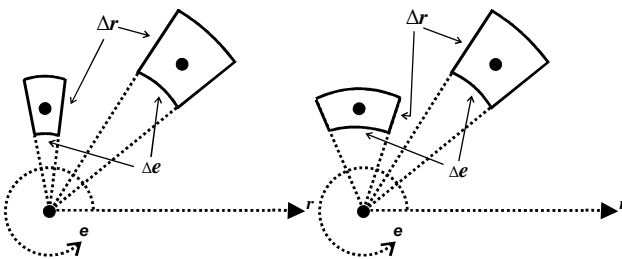
$$\Delta e = e_H - e_L \quad \dots\dots\dots(19)$$

$$\Delta r = r_H - r_L \quad \dots\dots\dots(20)$$

e_H dan e_L menyatakan batas atas dan bawah sudut radian pembatas, sedangkan r_H dan r_L menyatakan batas atas dan bawah radius pembatas. Ukuran pembatas dinyatakan dengan Δe dan Δr , yaitu ukuran sudut radian dan radius.



Gambar 6 Pembatas dan parameternya



(a). Pembatas berukuran tetap (b). Pembatas adaptif
Gambar 7 Pembatas berukuran tetap dan adaptif^[6]

Suatu pembatas adaptif (*adaptive boundary*) memiliki ukuran sudut dan radius yang berbeda pada setiap *minutiae* menurut besar radius *minutiae* tersebut. Jika radius suatu *minutiae* lebih besar, maka pembatasnya akan memiliki ukuran radius yang lebih besar dan ukuran sudut yang lebih kecil. Perbedaan antara pembatas berukuran tetap dan adaptif diperlihatkan pada gambar 7. Parameter pembatas adaptif dinyatakan sebagai berikut.^[6]

$$\Delta r = \begin{cases} r_b & \text{jika } r_s < r_b \\ r_s & \text{jika } r_b < r_s < r_a \\ r_a & \text{jika } r_s > r_a \end{cases} \quad \dots\dots\dots(21)$$

$$r_s = \frac{r}{\alpha} \quad \dots\dots\dots(22)$$

$$\Delta e = \begin{cases} e_b & \text{jika } e_s < e_b \\ e_s & \text{jika } e_b < e_s < e_a \\ e_a & \text{jika } e_s > e_a \end{cases} \quad \dots\dots\dots(23)$$

$$e_s = \frac{\beta}{r^2} \quad \dots\dots\dots(24)$$

dengan r_b , r_a , e_b , dan e_a adalah nilai minimum dan maksimum Δe dan Δr yang merupakan ukuran sudut dan radius pembatas, sedangkan r adalah radius *minutiae* terhadap *minutiae* referensi. α dan β adalah tetapan yang ditentukan terlebih dahulu. Dalam menentukan nilai-nilai parameter pembatas perlu diperhatikan bahwa jarak terjauh antara *minutiae* dan pembatas yaitu dm seperti terlihat pada gambar 6 tidak boleh lebih dari nilai rerata jarak antar alur bukit (D). Hal ini karena pada saat ekstraksi *minutiae*, dua *minutiae* dengan jarak lebih dari D merupakan *minutiae* yang berbeda.

Tujuan digunakannya pembatas adaptif daripada pembatas berukuran tetap adalah untuk mengatasi masalah perubahan bentuk citra (*deformation*). Ketika radius suatu *minutiae* kecil, sedikit perubahan bentuk pada citra akan berakibat perubahan besar terhadap sudut radian sementara perubahan pada radiusnya kecil. Dengan demikian, pada kasus ini Δe pembatas sebaiknya lebih besar dan nilai Δr sebaiknya lebih kecil. Pada keadaan lain, ketika radius dari *minutiae* besar, sedikit perubahan pada sudut radian akan mengakibatkan perubahan besar pada posisi *minutiae* sementara radiusnya dapat mengalami perubahan besar karena akumulasi perubahan dari seluruh daerah antara *minutiae* tersebut dengan *minutiae* referensi. Dengan demikian, pada kasus ini nilai Δe pembatas sebaiknya lebih kecil dan nilai Δr sebaiknya lebih besar.

Pada proses pencocokan *minutiae*, penentuan dua *minutiae* yang dipasangkan identik atau tidak digunakan rumus sebagai berikut.^[6]

$$\text{kondisi} = \begin{cases} \text{benar} & \text{jika} \begin{cases} r_L < r < r_H \\ e_L < e < e_H \\ \Delta\theta_m < \varepsilon \end{cases} \\ \text{salah} & \text{, untuk lainnya} \end{cases} \quad \dots\dots\dots(25)$$

dengan r dan e adalah radius dan sudut radian *minutiae* terhadap *minutiae* referensi, sedangkan $\Delta\theta_m$ dan ε adalah beda orientasi antara kedua *minutiae* dan nilai ambangnya. Jumlah *minutiae* cocok terbesar digunakan untuk menentukan nilai kecocokan. Nilai kecocokan S , dihitung menurut persamaan 26.^[3]

$$S = \frac{\text{jumlah pasang } \textit{minutiae} \text{ cocok}}{\text{jumlah pasang } \textit{minutiae} \text{ yang dicocokkan}} \times 100\% \quad \dots\dots\dots(26)$$

Nilai kecocokan atau derajat kecocokan ini dijadikan sebagai penentu apakah kedua sidik jari sama atau tidak dengan mengacu pada nilai ambang yang ditentukan.

IV. Pengujian dan Analisis

Dalam pengujian, sidik jari dari ibu jari tangan kanan 11 responden diambil dengan masing-masing dilakukan pengambilan sebanyak 23 kali dengan posisi ibu jari yang bervariasi saat pengambilannya. Hasilnya berupa citra sidik jari digunakan sebagai data acuan dan data uji. Untuk masing-masing responden, sebanyak 8 buah citra digunakan sebagai data acuan, sedangkan 15

buah citra lainnya digunakan sebagai data uji. Persentase pengenalan dihitung dengan persamaan 27 berikut.

$$\text{Persentase pengenalan} = \frac{\text{jumlah data uji dikenali benar}}{\text{jumlah data uji}} \times 100\% \quad \dots\dots\dots(27)$$

Nilai parameter pembatas berukuran tetap ditetapkan sebesar 5 derajat untuk ukuran sudut dan 10 piksel untuk ukuran radiusnya. Untuk pembatas adaptif ukuran sudut ditetapkan berinterval antara 5 sampai 20 derajat, sedangkan ukuran radius ditetapkan berinterval antara 2 sampai 10 piksel dengan nilai konstanta β sebesar 1000 dan α sebesar 7. Nilai ambang Δd , $\Delta \delta$, ϵ , dan S masing-masing ditetapkan 0,05 piksel, 0,2 radian, 0,2 radian, dan 50 %. Hasil pengujian disajikan pada tabel 1 berikut.

Tabel 1 Hasil pengujian pengenalan program terhadap citra uji

Responden	Persentase Pengenalan (%)	
	Program dengan Pembatas Berukuran Tetap	Program dengan Pembatas Adaptif
ADI	86,67	93,33
BRIAN	100	100
FUAD	80	86,67
LEGAWA	73,33	93,33
LUKMAN	66,67	93,33
PASKAH	66,67	93,33
PUNTOZ	40	93,33
REZA	53,33	86,67
TEGAR	60	93,33
VALEN	66,67	86,67
ZAINI	53,33	93,33
Rata-Rata	67,88	92,12

Dari tabel tersebut diketahui bahwa sistem pengenalan sidik jari yang menggunakan pembatas adaptif dalam proses pencocokan *minutiae* memiliki tingkat pengenalan yang lebih tinggi daripada sistem pengenalan sidik jari yang menggunakan pembatas berukuran tetap. Hal ini menunjukkan bahwa dengan menggunakan pembatas adaptif lebih tahan terhadap masalah lokalisasi *minutiae* yang tidak tepat dan deformasi citra sidik jari. Terdapat beberapa hal yang mempengaruhi tingkat pengenalan, yakni:

1) Derau

Derau sangat mempengaruhi hasil pengenalan sidik jari. Hal ini berkaitan dengan proses ekstraksi *minutiae*. Derau yang tidak hilang saat proses penapisan berpotensi merubah struktur alur bukit sidik jari sehingga dapat menimbulkan kesalahan dalam lokalisasi *minutiae*.

2) Jumlah *minutiae*

Jumlah *minutiae* dari citra sidik jari yang diujikan juga berpengaruh pada hasil pengenalan. Jumlah *minutiae* dari citra uji yang terlalu sedikit dapat mengakibatkan terjadinya pengenalan yang salah, artinya citra uji tersebut dikenali sebagai sidik jari seseorang padahal orang tersebut bukanlah pemilik sidik jari yang diujikan.

3) Penentuan ukuran pembatas

Penentuan ukuran pembatas harus tepat agar proses pencocokan *minutiae* dapat berjalan optimal. Ukuran pembatas yang terlalu besar dapat mengakibatkan sepasang *minutiae* yang tidak sama dianggap sama dalam proses pencocokannya sehingga dapat menimbulkan kesalahan dalam pengenalan. Sebaliknya, ukuran pembatas yang terlalu kecil tidak dapat mengkompensasi masalah lokalisasi *minutiae* yang tidak tepat dan masalah deformasi citra sehingga dapat mengakibatkan citra uji tidak dikenali.

V. Penutup

5.1 Kesimpulan

Dari hasil penelitian dan pembahasan dapat disimpulkan bahwa :

- 1) Tingkat pengenalan sistem menggunakan proses pencocokan *minutiae* dengan pembatas berukuran tetap adalah sebesar 67,88%. Pengenalan terhadap sidik jari responden terbesar adalah 100% yaitu pada citra Brian, sedangkan yang terkecil sebesar 40% pada citra Puntoz.
- 2) Tingkat pengenalan sistem menggunakan proses pencocokan *minutiae* dengan pembatas adaptif adalah sebesar 92,12%. Pengenalan terhadap sidik jari responden terbesar adalah 100% yaitu pada citra Brian, sedangkan yang terkecil sebesar 86,67 yaitu pada citra Fuad, Reza, dan Valen.
- 3) Penggunaan pembatas adaptif membuat sistem lebih tahan terhadap masalah akuisisi data yang tidak baku, terbukti dengan pembatas adaptif tingkat pengenalan sistem lebih tinggi bila dibandingkan dengan menggunakan pembatas berukuran tetap.
- 4) Ukuran pembatas yang terlalu besar dapat mengakibatkan sepasang *minutiae* yang tidak sama dianggap sama dalam proses pencocokannya sehingga dapat menimbulkan kesalahan dalam pengenalan. Sebaliknya, ukuran pembatas yang terlalu kecil tidak dapat mengkompensasi masalah lokalisasi *minutiae* yang tidak tepat dan masalah deformasi citra sehingga dapat mengakibatkan citra uji tidak dikenali.
- 5) Jumlah *minutiae* yang sedikit dapat menimbulkan kesalahan dalam pengenalan. Misalkan hanya ada dua *minutiae* pada salah satu citra yang dicocokkan, maka dengan sepasang *minutiae* cocok berarti derajat kecocokan sudah 50%. Dengan demikian rentan terjadinya kesalahan dalam pengenalan.

5.2 Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

- 1) Penelitian dapat dilanjutkan dengan membuat sistem pengenalan sidik jari secara langsung dengan proses akuisisi yang terhubung langsung dengan sistem.
- 2) Perlu dilakukan penelitian lebih lanjut mengenai penggunaan informasi pada alur bukit sidik jari untuk sistem pengenalan sidik jari, misalkan dengan menggabungkan fitur lokal dan global untuk proses pengenalannya.

DAFTAR PUSTAKA

- [1] Ardisasmita, M. Syamsa. "Pengembangan Model Matematika Untuk Analisis Sistem Identifikasi Sidik Jari Otomatis". Pusat Pengembangan Teknologi Informatika dan Komputasi, BATAN.
- [2] Hong, Lin. Automatic personal Identification Using Fingerprints. Department of Computer Science, Michigan State University, 1998.
- [3] Jain, Anil K. "Fingerprint Recogniton". <http://biometrics.cse.msu.edu>_Diakses Juli 2010.
- [4] Jain, Anil K dan Lin Hong. On-Line Fingerprint Verification. Department of Computer Science, Michigan State University, 1996.
- [5] Kovesi, Peter. "MATLAB and Octave Functions for Computer Vision and Image Processing". <http://www.csse.uwa.edu.au>. Diakses Juli 2010.
- [6] Luo, Xiping, Jie Tian, dan Yan Wu. "A Minutiae Matching Algorithm in Fingerprint Verification". AILAB, Institute of Automation, The Chinese Academy of Sciences, Beijing, 2000.
- [7] Munir, Rinaldi. Pengolahan Citra Digital dengan pendekatan Algoritmik. Informatika, Bandung, 2004.
- [8] Murmu, Neeta. Fingerprint Recognition. Department of Electrical Engineering, National Institute of Technology, 2008.
- [9] Thay, Raymond. Fingerprint Image Enhancement and Minutiae Extraction. School of Computer Science and Software Engineering, University of Western Australia, 2003.
- [10] _____. Image Processing Toolbox™ 7 User's Guide. MathWorks Inc, 2010.

Menyetujui,
Dosen Pembimbing I,

Achmad Hidayatno, S.T., M.T.
NIP. 19691221 199512 1 001

Dosen Pembimbing II,

Imam Santoso, S.T., M.T.
NIP. 19701203 199702 1 001

BIODATA



Tri Legawa, lahir di kota Semarang pada tanggal 12 Desember 1987. Telah menjalani pendidikan di Taman Kanak-kanak YWKA Semarang, Sekolah Dasar Negeri Bendungan 2 Semarang, Sekolah Lanjutan Tingkat Pertama Negeri 3 Semarang, Sekolah Menengah Umum Negeri 3 Semarang. Dan sekarang tengah menyelesaikan pendidikan Strata Satu di konsentrasi Elektronika dan Telekomunikasi, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang, Indonesia.