



Analisa Pengenalan Sidik Jari Berdasarkan Penjajaran Minutiae Dengan Algoritma Adaptif

Fuji Alfian

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: fujialfian025@gmail.com

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi : 07 November 2020

Revisi Akhir : 15 November 2020

Diterima : 20 November 2020

Diterbitkan Online : 28 November 2020

KATA KUNCI

Biometrik, Boundary, Minutiae, Adaptif

KORESPONDENSI

E-mail: fujialfian025@gmail.com

ABSTRACT

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 Biometrik. Adapun masalah Biometrik 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 minus. Jumlah dan letak *minutiae* pada setiap orang berbeda-beda sehingga dapat dijadikan pembeda antara orang yang satu dengan yang lain.

1. PENDAHULUAN

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 biometrik.

Biometrik 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 minus. Jumlah dan letak *minutiae* pada setiap orang berbeda-beda sehingga dapat dijadikan pembeda antara orang yang satu dengan yang lain.

Teknologi ini sangat efektif untuk memberikan perlindungan terhadap hal-hal yang penting dan juga menghindari penipuan karena tidak bisa hilang, sangat sulit ditiru, dan bukan merupakan sesuatu yang dapat dilupakan. Diantara seluruh biometri yang ada, pengenalan sidik jari adalah teknologi yang paling populer digunakan saat ini karena pemindai sidik jari sudah semakin murah dan bentuknya lebih kecil.

Menurut kamus besar bahasa indonesia algoritma adalah urutan logis pengambilan putusan untuk pemecahan masalah. Algoritma adalah logika, metode, dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan. Algoritma dapat juga diartikan sebagai urutan langkah secara sistematis dan logis. Dalam perkembangannya, algoritma banyak dipakai di bidang komputer.

Secara spesifik, pengertian algoritma adalah suatu metode khusus yang tepat dan terdiri dari serangkaian langkah yang terstruktur dan dituliskan secara matematis, yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer. Jadi berdasarkan definisi ini, dapat dikatakan algoritma merupakan langkah penyelesaian suatu masalah yang menghasilkan solusi dalam bentuk program komputer. Namun penting diketahui bahwa algoritma tidaklah tergantung oleh suatu bahasa pemrograman tertentu, artinya suatu algoritma harus dapat diwujudkan oleh bahasa pemrograman komputer apa pun.

2. METODOLOGI PENELITIAN

2.1 Sidik Jari (*Fingerprint*)

Menurut Ashbaugh (1991) Sidik jari atau *Dactyloscopy* adalah ilmu yang mempelajari sidik jari untuk keperluan pengenalan kembali identitas orang dengan cara mengamati garis yang terdapat pada guratan garis jari tangan dan telapak kaki. Fungsinya adalah untuk memberi gaya gesek lebih besar agar jari dapat memegang benda-benda lebih erat. Sidik jari manusia digunakan untuk keperluan identifikasi karena tidak ada dua manusia yang memiliki sidik jari persis sama. Hal ini mulai dilakukan pada akhir abad ke-19. Sidik jari kaki bayi juga diambil di rumah sakit untuk identifikasi bayi.

Sidik jari telah ditemukan pada tablet tanah liat di Babilon kuno, segel, dan tembikar. Mereka juga telah ditemukan di dinding kuburan Mesir dan Minoan, Yunani, dan keramik Cina - dan juga pada batu bata dan genteng di Babilon dan Roma. Beberapa dari sidik jari yang diendapkan secara tidak sengaja oleh para pekerja selama fabrikasi; kadang sidik jari berfungsi sebagai hiasan.

2.2 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:

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

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 ζ_x dan ζ_y . Nilai ζ_x dan ζ_y dapat dinyatakan dengan:

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

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

dengan F adalah frekuensi garis alur citra, k_x adalah konstanta untuk ζ_x , dan k_y adalah konstanta untuk ζ_y . Ukuran tapis dapat dinyatakan sebagai fungsi dari standar deviasi sebagai berikut:

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

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

Dengan w_x dan w_y adalah lebar dan tinggi penapis (*mask*) dari tapis Gabor, dan ζ_x dan ζ_y adalah standar deviasi sampul Gaussian sepanjang sumbu x dan y . Nilai lebar dan tinggi tapis ditetapkan 6 karena informasi berada diantara daerah $[-3, 3]$ 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. ANALISA DAN PEMBAHASAN

3.1. 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 penjajaran dan tahap pencocokan.

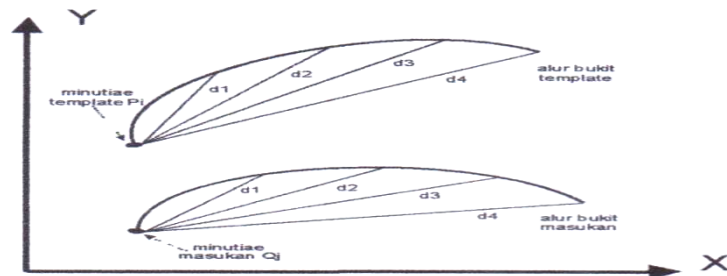
3.3.1 Tahapan Prosessing

Preprocessing diaplikasikan untuk gambar sidik jari dalam urutan *enhancement* feature sidik jari, tahapan proses ini mengacu berdasarkan penelitian dari Hong, Wan, dan Jain. Tahapan yang termasuk dalam sistem ini ialah : gambar ternormalisasi, *orientation estimation*, filtering, binerisasi, dan thinning, dimana masukkan proses berupa gambar asli

yang discan menggunakan scanner (epson type 660). Dan hasil akhirnya berupa penggambaran *ridge* skeleton yang siap disimpan sebagai *template* dalam database yang dimiliki.

3.4.2 Tahap Penjajaran

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



Gambar 3.3 Pengambilan titik-titik pada garis alur bukit *minutiae*

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:

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

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

dengan L adalah jumlah titik pada dua garis alur, adalah jarak dari titik i pada alur bukit R^d ke *minutiae* Q_j dan adalah sudut antara garis yang menghubungkan titik dengan *minutiae* P_i dan orientasi dari *minutiae* P_i . Dan memiliki pengertian yang sama, tetapi untuk *minutiae* P_i . Jika nilai dan 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

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

$(x_d, y_d)^T$ 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 berikut.

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

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.

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

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

3.4.2 Tahap Pencocokan

Diberikan $P = ((, ,)^T, \dots, (, ,)^T)$ menyatakan *minutiae* sebanyak M pada citra didata acuan dan $Q = ((, ,)^T, \dots, (, ,)^T)$ 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.

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

$(x_i^*, y_i^*, \theta_i^*)$ adalah koordinat dari *minutiae* terajar, $(x^{ref}, y^{ref}, \theta^{ref})^T$ 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.

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

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

Dengan (r^p, e^p, θ^p) dan (r^q, e^q, θ^q) merupakan radius, sudut radian dan orientasi *minutiae*.

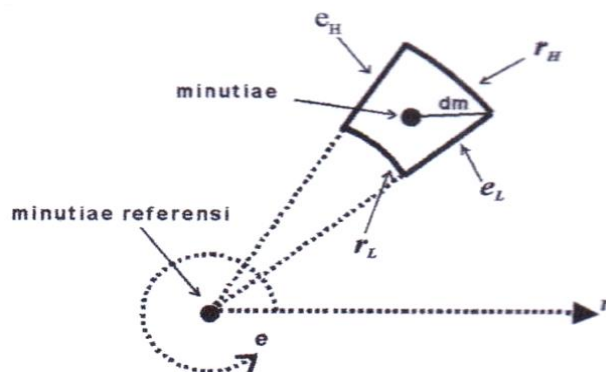
3. Mencocokkan untai P_p dan Q_p yang dihasilkan dengan algoritma pencocokan untai.
4. Memilih jumlah *minutiae* cocok terbesar untuk menentukan nilai kecocokan.. Hitung nilai kecocokannya.

Beberapa algoritma pencocokan untai 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:

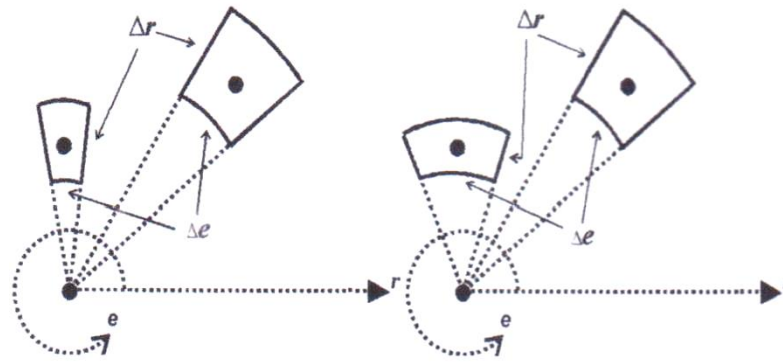
$$\Delta e = e_H - e_L \dots\dots\dots(19)$$

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

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 3.4 Pembatas dan parameternya



(a). Pembatas berukuran tetap

(b). Pembatas adaptif

Gambar 3.5 Pembatas berukuran tetap dan adaptif

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.

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

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

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

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

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 d_m 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.

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

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.

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

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

4. IMPLEMENTASI

4.1 Pengenalan Pola

Secara umum, pengenalan pola terdiri dari 4 langkah, yaitu :

Tahap ke-1 (*Image Acquisition*) : adalah proses mendapatkan data *image*. Pengambilan data dilakukan dengan metode *ink rolled* (tinta ditetaskan pada permukaan *stam pad* kemudian jari yang telah dikenai tinta dicapkan ke kertas dari ujung bawah kuku sampai pangkal jari). Setelah itu kertas di scan menggunakan *scanner canon*. Karena pengambilan *image* diambil secara langsung maka sering terjadi *trial and error*.

Tahap ke-2 (*Image Preprocessing*) : Pada tahap ini *image* diberi proses lain seperti penghilangan *noise/derau*, penajaman *image*, pemotongan *image*, dll. Program yang dipergunakan untuk memfilter *image* yaitu *Program Adobe Photoshop* dan *Microsoft Office Picture Manager*. Keluaran dari tahap ini adalah *image* tersegmentasi yang akan digunakan untuk proses selanjutnya dalam penelitian. *Image* tersegmentasi adalah *image* yang sudah dipisahkan dari *image* awal.

Tahap ke-3 (*Feature Extraction*) : Di tahap ini, *image* akan diekstrak untuk mendapatkan nilai-nilai yang merepresentasikan ciri spesifik dari *image* tersebut. *Image* dari tahap ke-2 akan diperkecil ukuran *pixel*nya karena jumlah datanya yang terlalu besar untuk dijadikan *input*, sehingga *image* diperkecil menjadi 8×8 *pixel*. *Image* ini dipilih karena masih dapat mewakili ciri citra asli, sesuai dengan rumus level maksimum dekomposisi yang dibatasi oleh persamaan yang berkorelasi (Kanata, 2008). *Image* ini akan diubah menjadi citra biner yang disimpan dalam matriks dengan nilai 0 (hitam) dan 1 (putih). Proses perubahan citra menggunakan MATLAB ver.6.1. dengan menggunakan perintah

```
Imdat = imread('sampel1(input).jpg');
```

```
imgray=rgb2gray(imdat);
```

```
imbw = im2bw(imgray)
```

Data yang dihasilkan (*imbw*) disimpan dalam file *.dat*

Tahap ke-4 (Image Recognition) :

Di tahap ini, vektor ciri *image* akan diklasifikasikan. Keluaran dari tahap ini adalah klasifikasi *image*, yang kemudian dapat secara langsung untuk mengenali suatu *image*. Metode yang digunakan adalah propagasi umpan balik berbasis jaringan syaraf tiruan.

4.2 Proses Pelatihan (Training)

1. Sebelum melakukan simulasi (*running*) pada *command window matlab*, terlebih dahulu dibuat instruksi-instruksi (*syntax*) dalam *M-file editor*. Dan untuk menghapus semua data dan *syntax* pada *command window* dituliskan instruksi *clear*, yang diakhiri dengan tanda titik koma (seperti: *clear*).
2. Masukkan data input dan target. Pada penelitian ini, JST didesain dan dilatih untuk mengenali 4 pola sidik jari. Masing – masing pola diwakili oleh nilai Boolean kisi 12x64. Yang terdiri dari 768 vektor *input* dan 12 vektor target. Masing-masing vektor *output* mewakili karakteristik sebuah sampel. Sebagai contoh, sampel 1 diwakili oleh target 1 (untuk alasan kemudahan, karena sampel 1 adalah sampel yang diambil pertama kali).

Tabel 1. Proses Pelatihan (Training)

No	Sampel ke -	Input	Target
1	Sampel 1	Sudut sidik ibu jari terhadap sensor :0°,30°,60°,90°,120°,150°,180°,210°,240° ,270°,300°,330°	1 1 1 1 1 1 1 1 1 1 1 1
2	Sampel 2	Sudut sidik ibu jari terhadap sensor :0°,30°,60°,90°,120°,150°,180°,210°,240° ,270°,300°,330°	2 2 2 2 2 2 2 2 2 2 2 2
3	Sampel 3	Sudut sidik ibu jari terhadap sensor :0°,30°,60°,90°,120°,150°,180°,210°,240° ,270°,300°,330°	3 3 3 3 3 3 3 3 3 3 3 3
4	Sampel 4	Sudut sidik ibu jari terhadap sensor :0°,30°,60°,90°,120°,150°,180°,210°,240° ,270°,300°,330°	4 4 4 4 4 4 4 4 4 4 4 4

3. Data-data yang diperoleh disimpan dalam file .dat lalu dimuat ke *M-File*.

Data input & target

```
P = load ('sampel semua.dat');
```

```
T = load ('target.dat');
```

4. Proses awal sebelum dilakukan pelatihan, data input dan target harus dinormalisasi. Proses normalisasi dapat dilakukan dengan bantuan mean dan deviasi standar. Fungsi *prestd* akan membawa data ke bentuk normal dengan mean = 0 dan deviasi standar = 1.

Preprocessing

```
[pn,mean,stdev,tn,meant,stdt] = prestd(P,T);
```

5. Membangun jaringan dengan algoritma propagasi umpan balik berbasis jaringan syaraf tiruan, instruksi yang digunakan *newff*. Membangun jaringan dengan propagasi umpan balik berbasis jaringan syaraf tiruan
`net = newff(minmax(pn),[10 5 1],{'tansig' 'logsig' 'purelin'},'traingdm')`

6. Melihat bobot awal input, lapisan dan bias.

Melihat bobot-bobot awal input, lapisan,dan bias

```
BobotAwal_Input           = net.IW{1,1}
BobotAwal_Bias_Input      = net.b{1,1}
BobotAwal_Lapisan1       = net.LW{2,1}
BobotAwal_Bias_Lapisan1  = net.b{2,1}
BobotAwal_Lapisan2       = net.LW{3,2}
BobotAwal_Bias_Lapisan2  = net.b{3,1}
```

7. Seting maksimum epoh, target error, learning rate, momentum dan epoh show.

Set max epoh, target error, learning rate, momentum dan epoh show

```
net.trainParam.epochs = 30000,-
net.trainParam.goal   = 1e-3;
net.trainParam.lr      = 0.5;
net.trainParam.mc      = 0.7;
net.trainParam.show    = 250;
```

8. Melakukan pembelajaran/pelatihan terhadap data input dan target.

Melakukan pembelajaran

```
[net,tr] = train(net,pn,tn); pause
```

9. Melihat bobot akhir input, lapisan.

Melihat bobot akhir input, lapisan, bias

```
BobotAkhir_Input = net.IW{1,1} BobotAkhir_Bias_Input =
net.b{1,1} BobotAkhir_Lapisan1 = net.LW{2,1}
BobotAkhir_Bias_Lapisan1 = net.b{2,1} BobotAkhir_Lapisan2 =
net.LW{3,2} BobotAkhir_Bias_Lapisan2 = net.b{3,1}
```

10. Melakukan simulasi dan hasilnya didenormalisasi pada data aslinya

Melakukan simulasi an = sim(net,pn)

```
a = poststd(an,meant,stdt)
```

```
H = [(1:size(P,2))' a] sprintf('%2d %5.2f\n',H)
```

11. Melakukan evaluasi terhadap output jaringan, yaitu data hasil pelatihan dengan target.

Melakukan evaluasi terhadap output jaringan [m1,a1,r1]=postreg(a,T

12. Menampilkan hasil pelatihan dalam bentuk grafik.

Menampilkan hasil pelatihan dalam bentuk grafik.

```
plot([1:size(P,2)]',T,'bo',[1:size(P,2)]',a,'r*')
```

```
title('Pengujian dengan data training: Target(o), output (*')
```

```
xlabel('data ke-');ylabel('Target');pause
```

4.3. Proses Tes (Testing)

1. Masukan data input dan target. Data-data yang diperoleh disimpan dalam file .dat lalu dimuat ke *M-File*.

```
% Data input & target
```

```
P = load ('sampel1(input).dat');
```

```
T = load ('sampel1(target).dat');
```

2. Melakukan simulasi jaringan.

```
% Simulasi jaringan
```

```
Qn = trstd(Q,meanp,stdp)
```

```
bn = sim(net,Qn)
```

```
b = poststd(bn,meant,stdt)
```

3. Menampilkan hasil perbandingan antara target yang didapat melalui metode propagasi umpan balik dengan target yang diinginkan dalam bentuk grafik.

```
%grafik
```

```
[b' TQ'];
```

```
plot(b,'b-*')
```

```
hold
```

```
plot(TQ,'r-o')
```

4.3.4 Proses Pengenalan Pola Sidik Jari

Menampilkan hasil dari proses testing, apakah target dikenali atau tidak.

```
disp('-----');
-
```

```
disp('                               Data sidik jari                               ');
```

```
disp('-----');
```

```
y=b';
```

```
x=sum(y)/4;
```

```
syaraf tiruan = '                               num2str(x)];
```

```
a=TQ';
```

```
b=sum(a)/4;
```

```
disp(['target yang diinginkan                               = 'num2str(b)]);
```

```
c=b-x;
```

```
e=c;
```

```
disp(['besar error                               = 'num2str(e)]);
```

```
if e>0.01
```

```
disp('                               Kesimpulan : target tidak dikenali                               ');
```

```
elseif e<=0.01
```

```
    disp('                                Kesimpulan : target dikenali                                ');
end;
```

5. KESIMPULAN

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.

1. 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.
2. Kesalahan dalam identifikasi sidik jari dipengaruhi oleh posisi jari saat pengambilan gambar sidik jari, banyaknya noise pada gambar sidik jari, jumlah dan posisi titik akhir yang mirip.

REFERENCES

- [1] http://eprints.undip.ac.id/32049/1/Tri_Legawa_dkk.pdf
- [2] <https://msaepulanwarstudent.wordpress.com/2016/04/28/analisa-artikel-sidik-jari-fingerprint-dan-pengertian-biometric/>. Analisaartikel sidik jari
- [3] <http://te.ugm.ac.id/~bsutopo/pri.pdf>
- [4] <https://aa12650204236.wordpress.com/2013/12/30/algoritma-routing/>
- [5] <http://belajarsidikjari.weebly.com/>
- [6] <http://a12650204219.blogspot.co.id/2013/12/algoritma-routing.html>
- [7] https://www.researchgate.net/publication/280698049_Automated_fingerprint_identification_system_dengan_teknik_minutiae-based/
- [8] <http://download.portalgaruda.org/article.php?article=146541&val=1028&title=Desain%20dan%20Implementasi%20Sistem%20Absensi%20Fingerprint%20di%20Jaringan%20Kampus%20dan%20Terintegrasi%20Dengan%20Sistem%20Informasi%20Terpadu%20UNSRAT>
- [9] <http://windahry.blogspot.com/2017/01/flowchart-sistem-cerdas-presensi.html>
- [10] https://www.researchgate.net/figure/Activity-diagram-Automated-Fingerprint-for-Military-Organisation_fig2_313371173