

BAB VI

CONTOH APLIKASI JST UNTUK PENGENALAN POLA SIDIK JARI

Kompetensi :

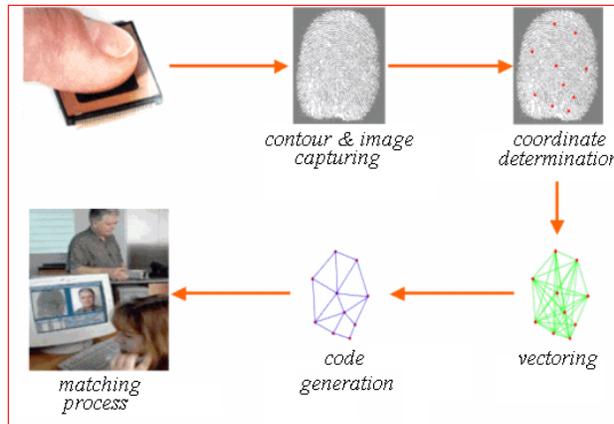
1. Mahasiswa memahami aplikasi JST

Sub Kompetensi :

1. Dapat melakukan pemrograman JST dengan Matlab.
2. Dapat mencoba kasus aplikasi JST untuk pengenalan pola sidik jari.
3. Dapat menerapkan JST untuk kasus rekayasa lainnya.

Pada bab ini akan dipaparkan contoh penggunaan Jaringan Syaraf Tiruan pada permasalahan pengenalan pola sidik jari manusia. Teknik identifikasi konvensional untuk mengenali identitas seseorang dengan menggunakan *password* atau kartu tidak cukup handal, karena sistem keamanan dapat ditembus ketika *password* dan kartu tersebut digunakan oleh pengguna yang tidak berwenang. Sistem autentifikasi data menggunakan pola sidik jari telah terbukti keakuratannya sebagai identifikasi bila dibandingkan dengan sistem *biometric* lainnya seperti retina mata atau DNA.

Dengan berkembangnya ilmu biometrik, akhirnya berkembang pula suatu sistem keamanan akses yang menggunakan karakteristik dari sidik jari sebagai *password* atau kunci untuk membuka akses keamanan tersebut. Berdasarkan hasil riset dari *internasional biometric group*, bahwa 48% industri biometrik lebih suka menggunakan aplikasi sidik jari untuk produk mereka daripada menggunakan identifikasi biometrik lainnya. Gambar dibawah ini memperlihatkan tahapan pengolahan data sistem pengenalan pola sidik jari untuk sistem keamanan akses.

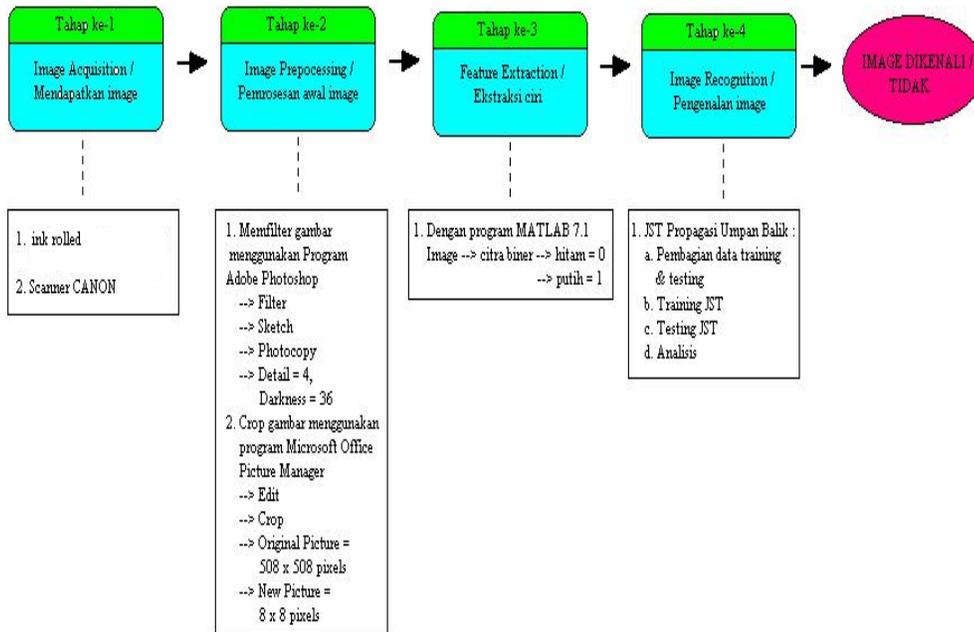


Gambar 1. Contoh aplikasi sidik jari pada sistem absensi karyawan.

Sistem pengenalan identitas dengan sidik jari seperti yang dijelaskan pada gambar di atas merupakan teknologi pengenalan pola yang banyak dipakai oleh industri biometrik pada saat ini. Sistem tersebut masih mempunyai kelemahan, dimana *user* harus menempelkan jari pada bidang sensor secara lurus, sehingga seringkali dijumpai *user* berulang kali mengalami kegagalan akses. Untuk menangani permasalahan tersebut kita akan mencoba merancang suatu sistem pengenalan pola yang lebih cerdas, yaitu sistem tersebut dapat mengenali pola sidik jari meskipun *user* menempatkan posisi jarinya secara sembarang pada bidang sensor. Perkembangan algoritma kecerdasan buatan (*artificial intelligent*) yang semakin handal akan dicoba untuk studi ini yaitu dengan menggunakan algoritma *feed forward backpropagation*.

IV.1. Pengenalan Pola

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



Gambar 1. Tahapan pengenalan pola sidik jari

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 8x8 *pixel*. *Image* ini dipilih karena masih dapat mewakili ciri citra asli, sesuai dengan rumus level maksimum dekomposisi yang dibatasi oleh persamaan yang berkorelasi (Kanata, 2008:7). *Image* ini akan diubah menjadi citra biner yang disimpan dalam matriks dengan nilai 0 (hitam) dan 1 (putih). Proses perubahan citra menggunakan MATLAB ver.7.1. dengan menggunakan perintah :

```

Imdat =imread('sampel1(input).jpg');
imgray=rgb2gray(imdat);
imbw =im2bw(imgray)

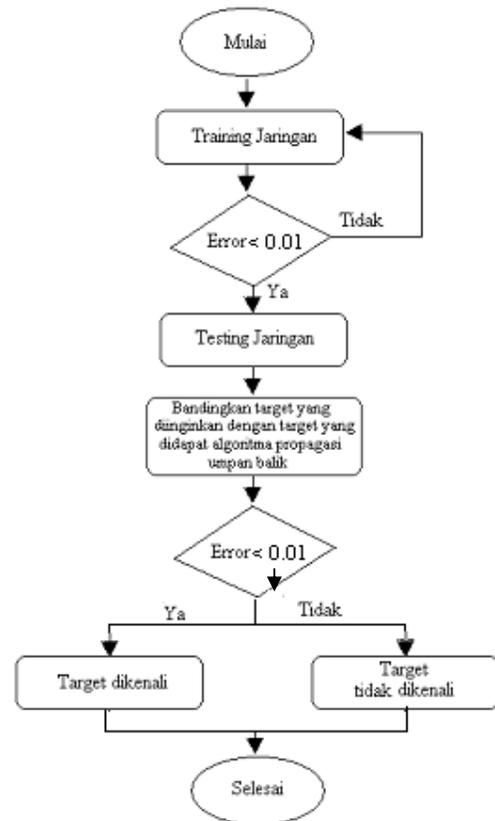
```

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.

Diagram alir (*flowchart*) pengenalan pola sidi umpan balik berbasis jaringan syaraf tiruan dapat dijelaskan pada gambar 3.2 disamping:

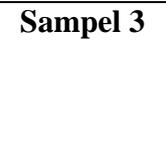


IV.2. Propagasi Umpan Balik Berbasis Jaringan Syaraf Tiruan Dengan Menggunakan Toolbox MATLAB ver.7.1

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).

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° (Ket : dapat dilihat pada lampiran)	1 1 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° (Ket : dapat dilihat pada lampiran)	2 2 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° (Ket : dapat dilihat pada lampiran)	3 3 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° (Ket : dapat dilihat pada lampiran)	4 4 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,meanp, stdp, 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, dan 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
```

Proses Tes (*Testing*)

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

```
% Data input & target
P = load ('sampell(input).dat');
T = load ('sampell(target).dat');
```

2. Melakukan simulasi jaringan.

```
% Simulasi jaringan
Qn = trastd(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')
```

Proses Pengenalan Pola Sidik Jari

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

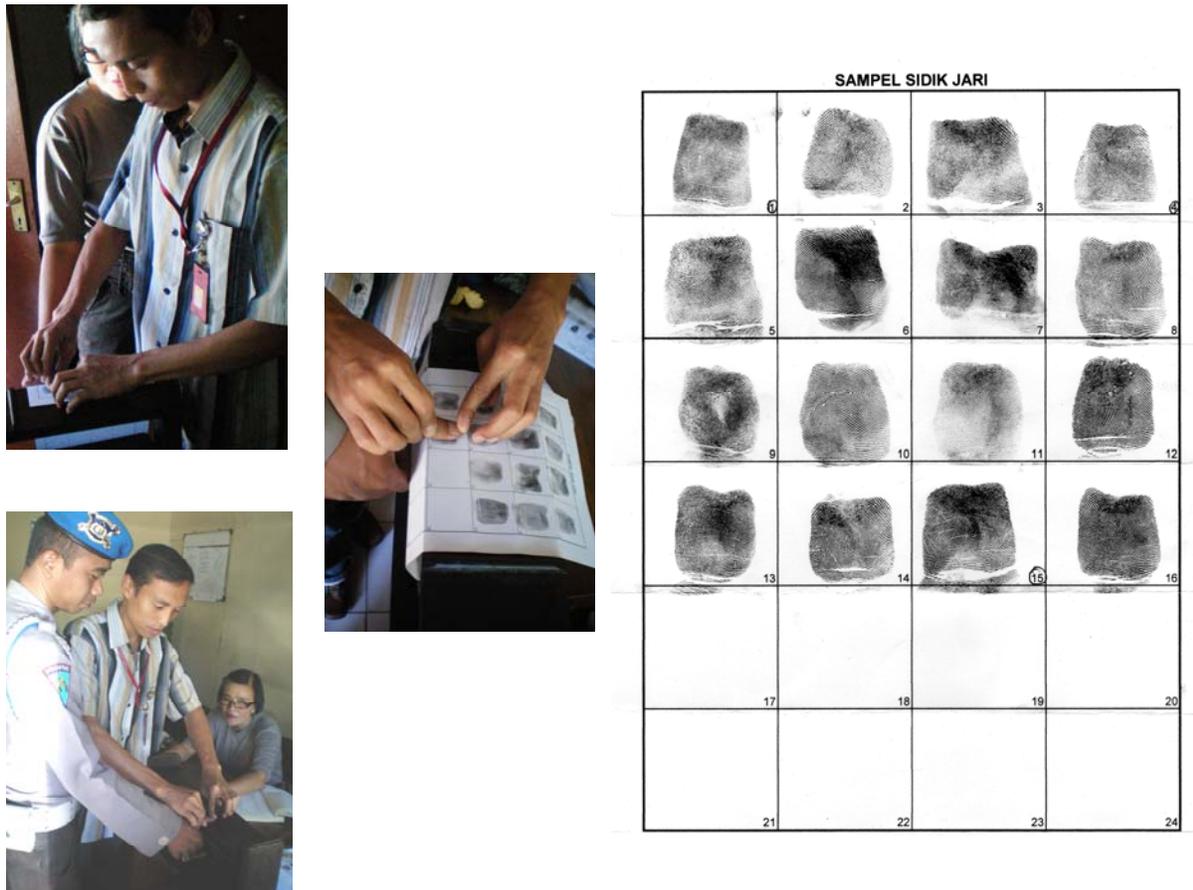
```
disp('-----');
disp('                Data sidik jari                ');
disp('-----');
y=b';
x=sum(y)/4;
disp(['target yang didapat algoritma propagasi umpan balik berbasis jaringan
syaraf tiruan = '                num2str(x)]);
a=TQ';
b=sum(a)/4;
disp(['target yang diinginkan                = 'num2str(b)]);
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;
```

VI.3. Hasil Penelitian

Untuk mengenali pola sidik jari menggunakan propagasi umpan balik berbasis jaringan syaraf tiruan terdapat 4 tahapan yaitu *image acquisition*, *image preprocessing*, *image feature extraction* dan *image recognition*. Sehingga hasil dan pembahasan untuk hal tersebut tercantum dalam pembahasan berikut ini.

1. Hasil *Image Acquisition*

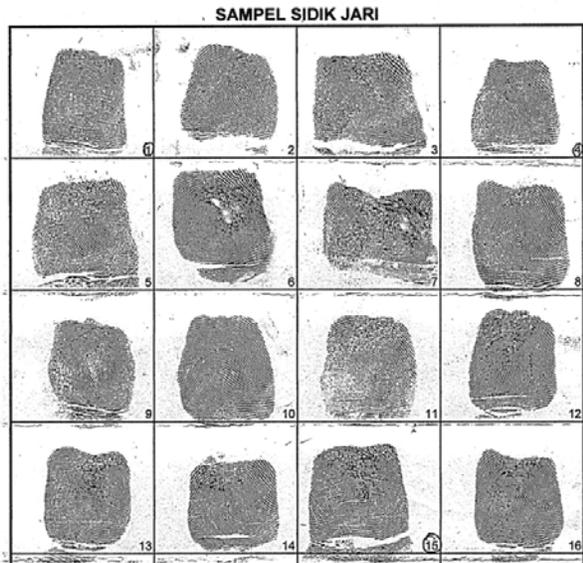
Sampel sidik jari diambil pada tanggal 4 Mei 2009 bertempat di Kapolresta Kota Bandung Tengah, menggunakan tinta sidik jari (*fingerprint ink*).



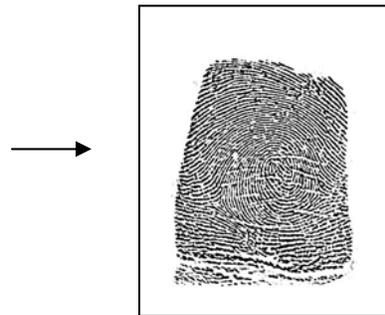
Gambar 2. Hasil sidik jari dengan metode ink rolled

4.2 Hasil *Image Preprocessing*

Image yang didapat kemudian di filter dan dicrop (di potong) agar mempermudah dalam mengidentifikasi karakteristik dari masing – masing sampel.

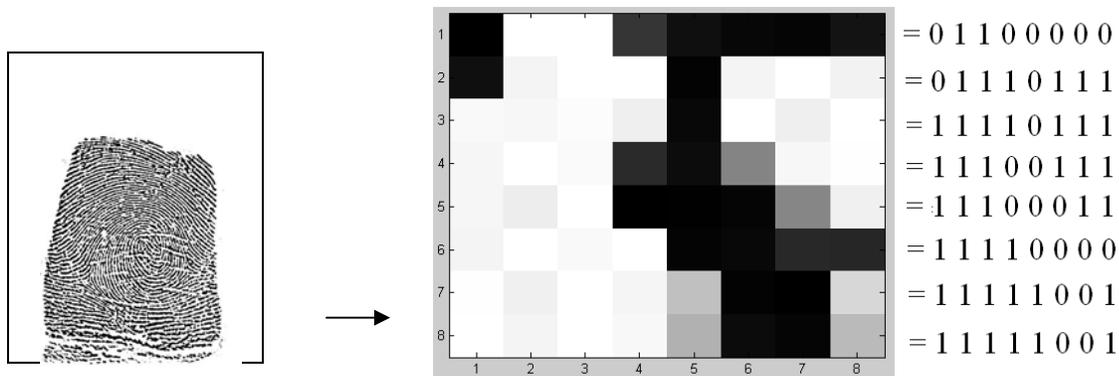


Gambar 3. Memfilter image dengan Program Adobe Photoshop



Gambar 4. Dimension 508x508

4.3 Hasil Image Feature Extraction



- Dimension image = 508 x 508

- Dimension image = 8 x 8

- Image yang akan dibinerisasi.

Gambar 5. Tahapan Image Feature Extraction

Hasil *image preprocessing* akan diperkecil ukuran *pixel*nya karena jumlah datanya yang terlalu besar untuk dijadikan *input*, sehingga *image* diperkecil menjadi 8x8 *pixel*. *Image* ini

dipilih karena masih dapat mewakili ciri citra asli, sesuai dengan rumus level maksimum dekomposisi yang dibatasi oleh persamaan yang berkorelasi (Kanata, 2008:7)

Hasil Image Recognition

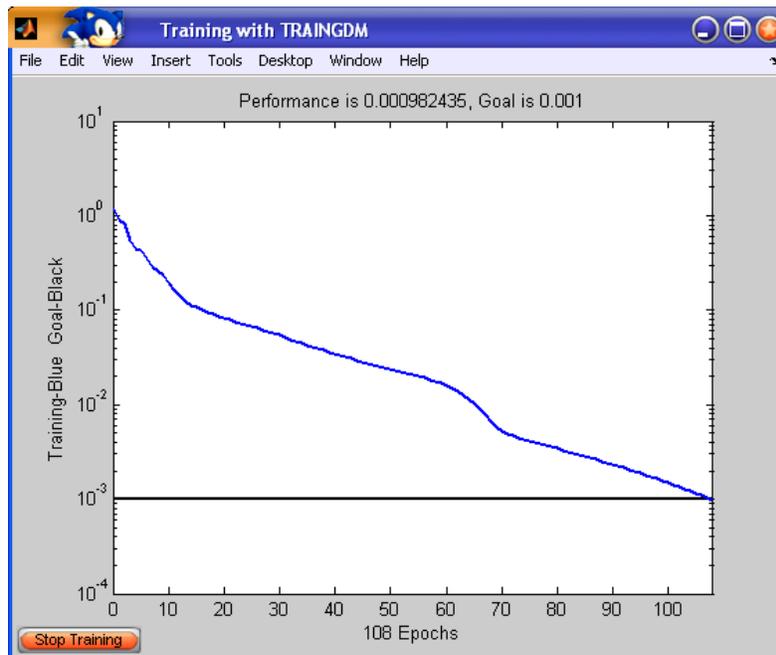
Hasil Proses Pelatihan (Training)

Karena bobot awal training adalah random maka saat dilakukan *training* lagi (tanpa adanya perubahan konstruksi JST) akan didapat bobot *training* yang berbeda, sehingga hasil yang didapat pun akan berbeda pula. Dibawah ini hasil dari beberapa percobaan *training*:

Tabel 1 Hasil percobaan *training*

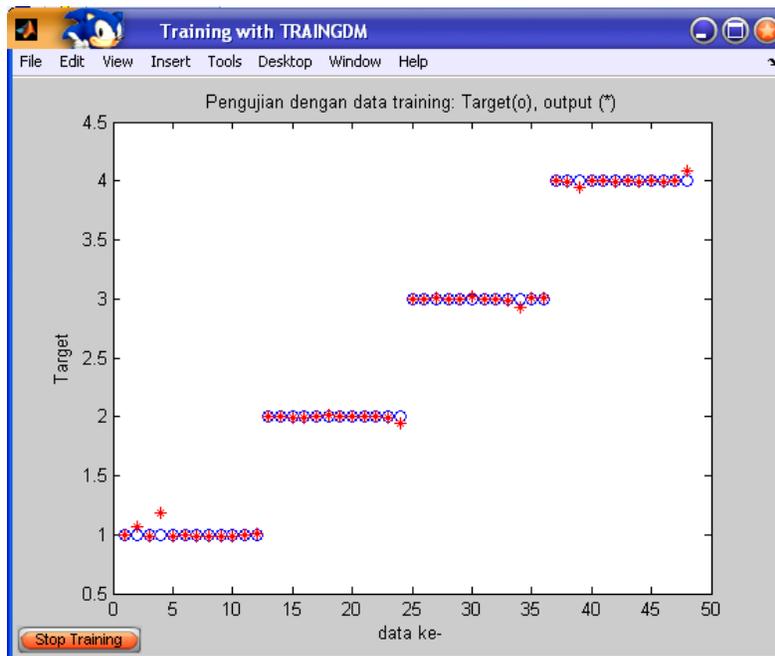
Percobaan Ke-	Jumlah Epoch	Besar error
1	149	0.000980178
2	179	0.000992874
3	131	0.00992383
4	116	0.000999847
5	127	0.000998413
6	272	0.000993525
7	213	0.000997468
8	318	0.000982992
9	137	0.000995935
10	108	0.000982435

Dari hasil percobaan di atas maka hasil dari percobaan ke- 10 yang akan ditampilkan pada grafik di bawah ini



Grafik 6. Grafik hasil proses pelatihan (training)

Grafik diatas menunjukkan bahwa mse didapat pada epoch ke- 108 dari 30000 epoch yang ditentukan di awal, dengan nilai mse senilai 0.000982435 dari 0.001 yang ditentukan.



Grafik 7. Perbandingan target yang diinginkan (T) dengan target hasil pelatihan algoritma propagasi umpan balik (H).

Hasil dari perbandingan target yang diinginkan (T) dengan target hasil pelatihan propagasi umpan balik berbasis jaringan syaraf tiruan (H).

Tabel 2. Perbandingan target yang diinginkan (T) dengan target hasil pelatihan propagasi umpan balik berbasis jaringan syaraf tiruan (H).

T	H	T	H	T	H	T	H
1	0.995663	2	2.001279	3	3.001602	4	3.998563
1	1.063432	2	2.004135	3	2.991353	4	3.994768
1	0.982419	2	1.990801	3	3.003588	4	3.940941
1	1.183388	2	1.99699	3	2.997532	4	4.001962
1	0.980387	2	2.001608	3	2.995913	4	4.000954
1	1.002046	2	2.009302	3	3.014674	4	3.994935
1	0.981966	2	2.001156	3	3.001445	4	3.999775
1	0.983033	2	2.000225	3	2.994774	4	3.995281
1	0.990984	2	1.999403	3	2.990348	4	3.997246
1	0.987137	2	2.00631	3	2.927237	4	3.991658
1	0.991465	2	1.994478	3	3.006144	4	4.005958
1	1.00732	2	1.947462	3	3.011519	4	4.090609

Untuk menganalisa hasil tabel diatas, dapat dilakukan analisis regresi. Pada MATLAB menyediakan fungsi *postreg* untuk melakukan evaluasi ini. Dan hasil yang didapat adalah

>> m1 = 0.9964

Penjelasan : **m** adalah gradient garis hasil regresi linear. Apabila *output* jaringan tepat sama dengan targetnya, maka gradient ini akan bernilai 1. Dari hasil **m** yang didapat oleh MATLAB maka *output* jaringan hampir tepat sama dengan targetnya.

>> a1 = 0.0100

Penjelasan : **a** adalah titik perpotongan dengan sumbu y. Apabila *output* jaringan tepat sama dengan targetnya, maka perpotongan dengan sumbu y ini akan bernilai 0. Dari hasil **a** yang didapat oleh MATLAB maka *output* jaringan hampir tepat sama dengan targetnya.

>> r1 = 0.9995

Penjelasan : **r** adalah koefisien korelasi antara *output* jaringan dan target. Apabila *output* jaringan tepat sama dengan targetnya, maka koefisien korelasi ini akan bernilai 1. Dari hasil **r** yang didapat oleh MATLAB maka *output* jaringan hampir tepat sama dengan targetnya.

Penjelasan Hasil Proses Pelatihan (*Training*)

Seperti yang terlihat dari hasil pelatihan diatas, dapat disimpulkan bahwa pada proses pelatihan (*training*) *output* jaringan hampir tepat sama dengan targetnya.

Hasil Proses *Testing* :

Tabel 4.3 Hasil uji propagasi umpan balik berbasis jaringan syaraf tiruan.

No	Input	Target	Output JST	Error	Keputusan
1	Sampel 1.dat	Sampel 1	1.0178	0.01783	Dikenali
2	Sampel 1.dat	Sampel 2	1.0185	0.98154	Tidak dikenali
3	Sampel 1.dat	Sampel 3	1.0001	1.9999	Tidak dikenali
4	Sampel 1.dat	Sampel 4	1.0024	2.9976	Tidak dikenali
5	Sampel 1(0°,90°,180°,270°).dat	Sampel 1	0.99947	0.00053	Dikenali
6	Sampel 1(0°,90°,180°,270°).dat	Sampel 2	0.99914	1.0009	Tidak dikenali
7	Sampel 1(0°,90°,180°,270°).dat	Sampel 3	1.0317	1.9683	Tidak dikenali
8	Sampel 1(0°,90°,180°,270°).dat	Sampel 4	1.0126	2.9874	Tidak dikenali
9	Sampel 1(30°,60°,120°,150°).dat	Sampel 1	1.0231	0.0231	Dikenali
10	Sampel 1(30°,60°,120°,150°).dat	Sampel 2	1.0168	0.9832	Tidak dikenali
11	Sampel 1(30°,60°,120°,150°).dat	Sampel 3	1.0029	1.9971	Tidak dikenali
12	Sampel 1(30°,60°,120°,150°).dat	Sampel 4	1.0438	2.9562	Tidak dikenali
13	Sampel 1(210°,240°,300°,330°).dat	Sampel 1	1.0095	0.0095	Dikenali
14	Sampel 1(210°,240°,300°,330°).dat	Sampel 2	1.0077	0.9923	Tidak dikenali
15	Sampel 1(210°,240°,300°,330°).dat	Sampel 3	1.0076	1.9924	Tidak dikenali
16	Sampel 1(210°,240°,300°,330°).dat	Sampel 4	1.007	2.993	Tidak dikenali
17	Sampel 2.dat	Sampel 1	1.992	0.9919	Tidak dikenali
18	Sampel 2.dat	Sampel 2	1.992	0.0082	Dikenali
19	Sampel 2.dat	Sampel 3	2.0047	0.9953	Tidak dikenali
20	Sampel 2.dat	Sampel 4	2.0056	1.9944	Tidak dikenali
21	Sampel 3.dat	Sampel 1	2.9994	1.9994	Tidak dikenali
22	Sampel 3.dat	Sampel 2	2.9963	0.9963	Tidak dikenali
23	Sampel 3.dat	Sampel 3	2.9995	0.0005	Dikenali
24	Sampel 3.dat	Sampel 4	3.0021	0.9978	Tidak dikenali
25	Sampel 4.dat	Sampel 1	3.9929	2.9989	Tidak dikenali
26	Sampel 4.dat	Sampel 2	4.0019	2.0019	Tidak dikenali
27	Sampel 4.dat	Sampel 3	3.9955	0.9955	Tidak dikenali
28	Sampel 4.dat	Sampel 4	4.0005	0.0005	Dikenali

Penjelasan Hasil Proses Pengujian (*Testing*)

Pada proses testing diatas, JST dilatih dalam 2 kondisi yaitu :

1. JST di uji untuk mengenali pola sidik jari dari berbagai sudut.
2. JST di uji untuk mengenali pola sidik jari dengan mengganti target yang seharusnya dengan target dari input yang lain. Contoh : data pada sampel 1 hanya akan mengenali output dengan

target 1, maka jika input nya sesuai dengan output maka sistem akan mengenali, tetapi apabila input nya tidak sesuai dengan output maka sistem tidak akan mengenali pola tersebut. Dari hasil pengujian (tabel 3) didapatkan bahwa dari 28 pengujian , sistem berhasil mengidentifikasi sidik jari tanpa terdapat suatu kesalahan.