

# APLIKASI ALGORITMA FEED FORWARD BACKPROPAGATION PADA SISTEM KEAMANAN AKSES MENGGUNAKAN SIDIK JARI.

Ade Gafar Abdullah, Neni Fitriani, Jaja Kustija, Iwan Kustiawan

<sup>1</sup> Program Studi Pendidikan Teknik Elektro, FPTK UPI  
Jl. Dr. Setiabudhi 207 Bandung 40154  
email : [ade\\_gaffar@upi.edu](mailto:ade_gaffar@upi.edu)

## ABSTRAK

Makalah ini membahas aplikasi jaringan syaraf tiruan dengan arsitektur *multi layer* menggunakan algoritma belajar *supervised learning feed forward back propagation*. Jaringan ini menggunakan fungsi aktivasi *sigmoid biner* dengan *learning rate* 0.5 dan setting maksimum *epoch* dibatasi hingga 30000. Jaringan *feed forward back propagation* ini digunakan untuk melakukan pengenalan pola sidik jari manusia yang diaplikasikan untuk sistem keamanan akses. Proses pengolahan gambar (*image processing*) melewati tahapan *image acquisition*, *image preprocessing*, *image feature extraction* dan *image recognition*. Pada tahap *feature extraction* image diperkecil ukurannya menjadi *8x8 pixel*. Pola sidik jari yang diinputkan terdiri dari 12 posisi derajat kemiringan yaitu pada posisi  $0^{\circ}$ ,  $30^{\circ}$ ,  $60^{\circ}$ ,  $90^{\circ}$ ,  $120^{\circ}$ ,  $150^{\circ}$ ,  $180^{\circ}$ ,  $210^{\circ}$ ,  $240^{\circ}$ ,  $270^{\circ}$ ,  $300^{\circ}$ , dan  $330^{\circ}$ . dan uji coba dilakukan terhadap 4 sampel sidik jari yang berbeda. Hasil simulasi dan percobaan membuktikan bahwa tingkat ketelitian pengenalan pola tergantung pada jumlah epoch yang digunakan serta setting learning ratenya.

**Kata Kunci** : propagasi umpan balik, keamanan akses, pola sidik jari.

## ABSTRACT

*This paper discusses the application of neural networks with multi layer architecture using learning algorithm, namely supervised learning feed forward back propagation. This network uses function of the binary sigmoid activation with learning rate 0.5 and the maximum setting for epoch is limited to 30 000. This feed forward back propagation is used to perform human fingerprint pattern recognition which is applied for accessing security system. Image processing is done by the stages of image acquisition, image preprocessing, image feature extraction and image recognition. At this stage of feature extraction image will be reduced in size to 8x8 pixels. The pattern of input fingerprint consists of 12 degree tilt position exactly at positions  $0^{\circ}$ ,  $30^{\circ}$ ,  $60^{\circ}$ ,  $90^{\circ}$ ,  $120^{\circ}$ ,  $150^{\circ}$ ,  $180^{\circ}$ ,  $210^{\circ}$ ,  $240^{\circ}$ ,  $270^{\circ}$ ,  $300^{\circ}$ , dan  $330^{\circ}$ . The test is carried out on 4 samples of different fingerprints. Results of simulation and experiments prove that the accuracy level of the pattern recognition depends on the number of used epochs and the setting of learning rate.*

**Keywords:** *feedback propagation, access security, fingerprint pattern*

## I. PENDAHULUAN

Teknik identifikasi konvensional untuk mengenali identitas seseorang dengan menggunakan *password* atau kartu tidak cukup handal, karena system 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 system *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.

Sistem pengenalan identitas dengan sidik jari yang banyak dipakai oleh industri biometrik pada saat ini masih memiliki kelemahan, dimana *user* harus menempelkan jari pada bidang sensor secara lurus, sehingga seringkali dijumpai *user* berulang kali mengalami

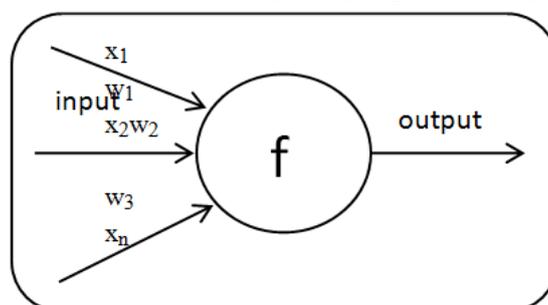
kegagalan akses. Untuk menangani permasalahan tersebut peneliti mencoba merancang suatu sistem pengenalan pola yang lebih cerdas, yaitu sistem tersebut dapat mengenali pola sidik jari meskipun *user* menempatkan posisi jarinya secara sembarang pada bidang sensor. Sistem pengenalan pola yang akan diujicobakan menggunakan teknologi *softcomputing* yaitu jaringan syaraf tiruan (*artificial neural network*) dengan algoritma *feed forward back propagation*.

## II. TINJAUAN PUSTAKA

### Jaringan Syarat Tiruan

Jaringan syaraf tiruan adalah suatu simulasi abstrak dari sistem syaraf dalam otak manusia yang mengandung sekelompok neuron yang terinterkoneksi satu sama lain. Menurut Haykin, JST didefinisikan bahwa sebuah neural network adalah prosesor yang terdistribusi paralel, terbuat dari unit-unit yang sederhana, dan memiliki kemampuan untuk menyimpan pengetahuan yang diperoleh secara eksperimental dan siap pakai untuk berbagai tujuan. Neural network ini meniru otak manusia dari sudut : 1) pengetahuan diperoleh oleh network dari lingkungan, melalui proses pembelajaran. 2) Kekuatankoneksi antar unit yang disebut synaptic weight, berfungsi untuk menyimpan pengetahuan yang telah diperoleh jaringan tersebut.

Pemodelan neuron tiruan yang membangun jaringan ini pertama kali diperkenalkan oleh McCulloch dan Pitts pada tahun 1943. Model neuron M-P ini menjadi awal perkembangan selanjutnya oleh pakar seperti John von Neuman, Marvin Minsky, Frank Rosenblatt dan lain-lain. McCulloch dan Pitts memperkenalkan model matematika yang merupakan penyederhanaan dari struktur sel syaraf yang sebenarnya (seperti pada gambar 1):



Gambar 1. Model neuron McCulloch dan Pitts

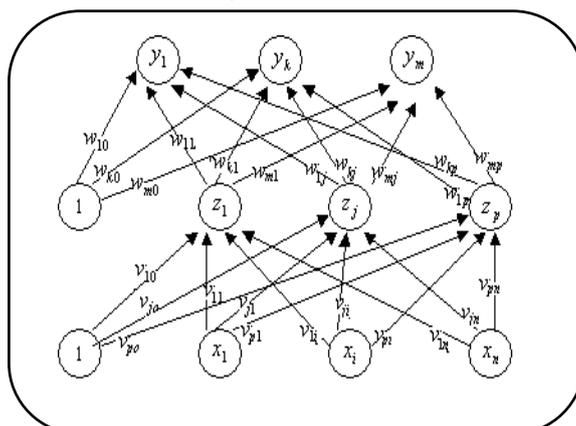
Signal  $x$  berupa vektor berdimensi  $n$  ( $x_1, x_2, \dots, x_n$ ) akan mengalami penguatan oleh sinapsis  $w$  ( $w_1, w_2, \dots, w_n$ ). Selanjutnya akumulasi dari penguatan tersebut akan mengalami transformasi oleh fungsi aktivasi  $f$ . Fungsi ini akan memonitor, bila akumulasi penguatan signal itu telah melebihi batas tertentu, maka sel neuron yang semula berada dalam kondisi “0” akan mengeluarkan sinyal “1”. Berdasarkan nilai output tersebut, sebuah neuron dapat berada dalam dua status : “0” atau “1”. Neuron tersebut dalam kondisi *firing* bila menghasilkan output bernilai “1”.

### Algoritma *Feed Forward Backpropagation*

Algoritma *feed forward backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot- bobot yang terhubung dengan neuron – neuron yang ada pada lapisan tersembunyinya. *Feed forward backpropagation* menggunakan *error output* untuk mengubah nilai bobot – bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron – neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensiasi.

*Feed forward backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar 2 adalah arsitektur *back propagation* berbasis jaringan syaraf

tiruan dengan  $n$  buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit keluaran (Jek, 2005:98).



Gambar 2. Arsitektur *Feed Forward Back Propagation*

$v_{ji}$  merupakan bobot garis dari unit masukan  $x_i$  ke unit layer tersembunyi  $z_j$  ( $v_{j0}$  merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layer tersembunyi  $z_j$ ).  $w_{kj}$  merupakan bobot dari unit layer tersembunyi  $z_j$  ke unit keluaran  $y_k$  ( $w_{k0}$  merupakan bobot dari bias di layer tersembunyi ke unit keluaran  $z_k$ ) (Jek, 2005:98).

### Pelatihan *Feed Forward Back Propagation*

Pelatihan *feed forward backpropagation* meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layer masukan hingga layer keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layer keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi (Jek, 2005:100).

#### ▪ Fase I: Propagasi Maju

Selama propagasi maju, sinyal masukan ( $= x_i$ ) dipropagasikan ke layer tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layer tersembunyi ( $= z_j$ ) tersebut selanjutnya dipropagasikan maju lagi ke layer tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ( $= y_k$ ).

Berikutnya, keluaran jaringan ( $= y_k$ ) dibandingkan dengan target yang harus dicapai ( $= t_k$ ). Selisih dari  $t_k$  terhadap  $y_k$  yaitu ( $t_k - y_k$ ) adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

#### ▪ Fase II: Propagasi Mundur

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_k$  ( $k = 1, 2, \dots, m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor  $\delta_j$  ( $j = 1, 2, \dots, p$ ) di setiap unit di layer tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layer di bawahnya. Demikian seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

▪ **Fase III: Perubahan Bobot**

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas  $\delta_k$  yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahanyang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Algoritma pelatihan untuk jaringan dengan satu layer tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut :

**Langkah 0** : Inisialisasi semua bobot dengan bilangan acak kecil.

**Langkah 1** : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 8.

**Langkah 2** : Untuk setiap pasang data pelatihan, lakukan langkah 3 – 8.

**Fase I : Propagasi maju**

**Langkah 3:** Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.

**Langkah 4** : Hitung semua keluaran di unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ ).

$$z\_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}}$$

**Langkah 5** : Hitung semua keluaran jaringan di unit  $y_k$  ( $k = 1, 2, \dots, m$ ).

$$y\_net_k = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y\_net_k) = \frac{1}{1 + e^{-y\_net_k}}$$

**Fase II : Propagasi mundur**

**Langkah 6** : Hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran  $y_k$  ( $k = 1, 2, \dots, m$ ).

$\delta_k = (t_k - y_k) f'(y\_net_k) = (t_k - y_k) y_k (1 - y_k)$   $\delta_k$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya ( langkah 7 ).

Hitung suku perubahan bobot  $w_{kj}$  (yang akan dipakai nanti untuk merubah bobot  $w_{kj}$ ) dengan laju perceptron  $\alpha \cdot \Delta w_{kj} = \alpha \delta_k z_j$ ;  $k = 1, 2, \dots, m$ ;  $j = 0, 1, \dots, p$

**Langkah 7** : Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ ).

$$\delta\_net_j = \sum_{k=1}^m \delta_k w_{kj}$$

Faktor  $\delta$  unit tersembunyi :

$$\delta_j = \delta\_net_j f'(z\_net_j) = \delta\_net_j Z_j (1 - Z_j)$$

Hitung suku perubahan bobot  $v_{ji}$  (yang akan dipakai nanti untuk merubah bobot  $v_{ji}$ ).

$$\Delta v_{ji} = \alpha \delta_j x_i ; j = 1, 2, \dots, p ; i = 0, 1, \dots, n$$

**Fase III : Perubahan bobot**

**Langkah 8** : Hitung semua perubahan bobot.

Perubahan bobot garis yang menuju ke unit keluaran :

$$W_{kj}(\text{baru}) = W_{kj}(\text{lama}) + \Delta W_{kj} \quad (k = 1, 2, \dots, m ; j = 0, 1, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi :

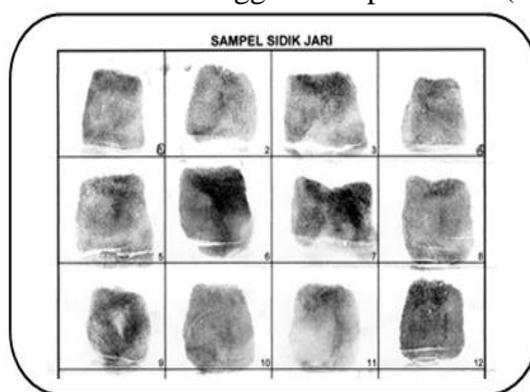
$$V_{ji}(\text{baru}) = V_{ji}(\text{lama}) + \Delta v_{ji}(j = 1, 2, \dots, p ; I = 0, 1, \dots, n)$$

Setelah selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju ( langkah 4 dan 5 ) saja yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada langkah 6 dan 7.

### III. METODOLOGI

Makalah ini membahas tentang penggunaan algoritma feed forward backpropagation yang diaplikasikan untuk pengenalan pola sidik jari sebagai salah satu teknologi biometrik untuk keamanan akses. Untuk membelajarkan pola sidik jari ke dalam algoritma JST, diperlukan 4 tahap pengolahan image :

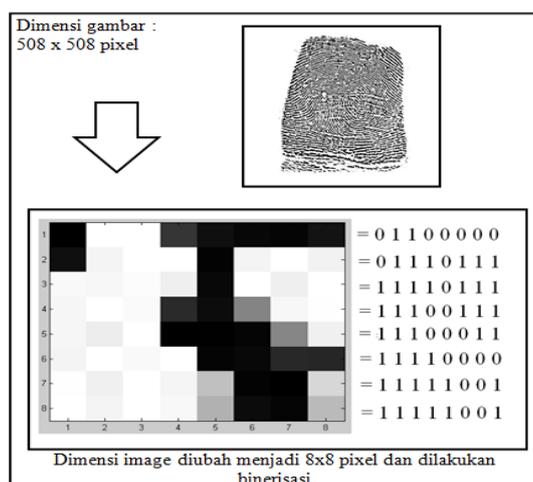
**Tahap ke-1 (*Image Acquisition*)** : adalah proses mendapatkan data *image*. Pengambilan data dilakukan dengan metode *ink rolled* (tinta ditetaskan pada permukaan *stam pad* kemudian jari yang telah dikenai tinta dicapkan ke kertas dari ujung bawah kuku sampai pangkal jari). Setelah itu kertas di scan menggunakan pemindai (*scanner*).



Gambar 3. Hasil sidik jari dengan metode ink rolled

**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. *Image* tersegmentasi adalah *image* yang sudah dipisahkan dari *image* awal.

**Tahap ke-3 (*Feature Extraction*)** : Pada 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  $8 \times 8$  *pixel*. *Image* ini dipilih karena masih dapat mewakili ciri citra asli, sesuai dengan rumus level maksimum dekomposisi yang dibatasi oleh persamaan yang berkorelasi (Kanata, 2008:7). *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



Gambar 4. Tahapan Image Feature Extraction

**Tahap ke-4 (Image Recognition)** : Pada tahap ini, vector ciri *image* akan diklasifikasikan. Keluaran dari tahap ini adalah klasifikasi *image*, yang kemudian dapat digunakan secara langsung untuk mengenali suatu *image*. Algoritma yang digunakan adalah *feed forward*.

Langkah – langkah diatas dapat dibangun dari data – data masukan pola sidik jari dengan posisi pola sidik jari yang berbeda. Data input yang diambil terdiri dari empat pola input dan memiliki 12 posisi derajat kemiringan. Diantaranya, posisi  $0^0$ ,  $30^0$ ,  $60^0$ ,  $90^0$ ,  $120^0$ ,  $150^0$ ,  $180^0$ ,  $210^0$ ,  $240^0$ ,  $270^0$ ,  $300^0$ , dan  $330^0$ . Dari Masing – masing data input tersebut di buat matriks ukuran 8 x 8.

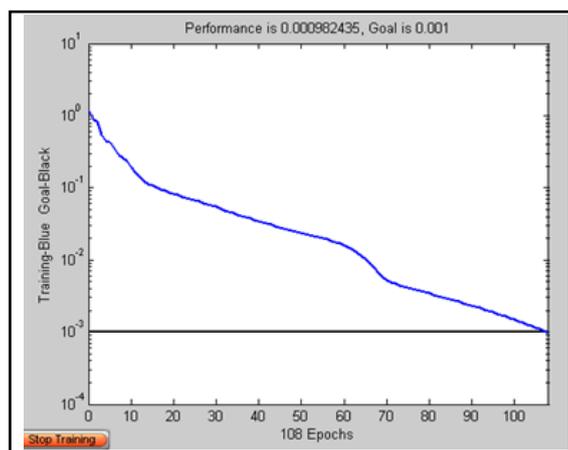
#### IV. HASIL DAN PEMBAHASAN

Akurasi pengenalan pola pada proses pelatihan ditentukan oleh setting bobot awal. Bobot awal diujicobakan secara random dan dari hasil 10 kali percobaan didapat hasil yang optimum.

Tabel 1. Hasil optimasi bobot awal

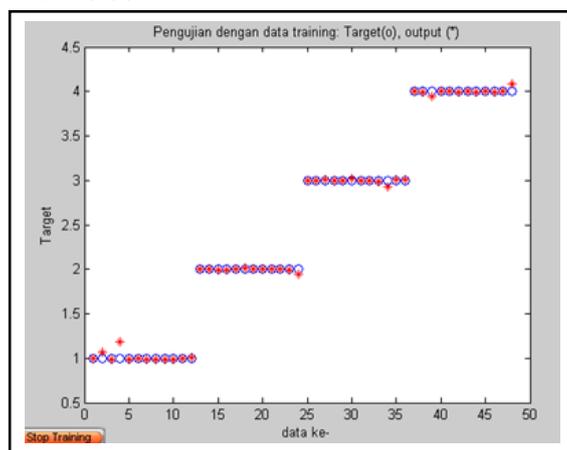
Percobaanke-	Jumlah Epoch	Besar error
1	149	0.000980178
2	179	0.000992874
3	131	0.00992383
4	116	0.000999847
5	127	0.000998413
6	272	0.000993525
7	213	0.000997468
8	318	0.000982992
9	137	0.000995935
10	108	0.000982435

Hasil optimasi bobot awal pada proses pelatihan menghasilkan jumlah epoch error terkecil sehingga nilai yang didapat dijadikan acuan untuk proses ujicoba.



Gambar 5. Grafik hasil pelatihan

Gambar 5 memperlihatkan jumlah epoch minimum dari hasil optimasi dan didapatkan *mean square error (mse)* sebesar 0.000982435 sehingga telah memenuhi syarat dari *mse* yang sudah ditentukan yaitu sebesar 0.001.



Gambar 6. Perbandingan target yang diinginkan (T) dengan target hasilpelatihan (H).

Gambar 6 memperlihatkan perbandingan antara target yang diinginkan dengan target hasil pelatihan. Secara visual terlihat bahwa target yang diinginkan (o) secara umum dapat dikenali oleh target pelatihan (\*) meskipun ada 6 buah target pelatihan (\*) yang tidak menempel tepat di target yang diinginkan (o). Ujicoba terhadap sampel yang berjumlah 4 sidik jari yang berbeda diklasifikasikan terhadap 12 posisi sudut sidik jari yaitu 0°, 30°, 60°, 90°, 120°, 150°, 180°, 210°, 240°, 270°, 300° dan 330°. Hasil uji coba terhadap 4 sampel tersebut dapat dilihat pada tabel 2 di bawah ini :

Tabel 2. Hasil pengujian terhadap 4 sampel

No	Input	Error Akurasi (%)
1	Sampel 1	0.01783
2	Sampel 2	0.0082
3	Sampel 3	0.0005
4	Sampel 4	0.0005

Hasil pengujian memberikan kesimpulan bahwa hasil pengenalan pola sidik jari terhadap sampel cukup akurat dengan rata-rata kesalahan pengenalan pola sebesar 0,006758 %. Nilai rata-rata kesalahan dapat diperbaiki dengan melakukan optimasi terhadap nilai bobot awal, jumlah epoch dan setting learning rate.

## V. KESIMPULAN

1. Algoritma *feed forward backpropagation* telah terbukti dapat digunakan untuk pengenalan pola sidik jari dan memiliki *error recognition* yang sangat kecil.
2. Algoritma *feed forward backpropagation* memiliki kelebihan dalam proses belajarnya dikarenakan memiliki arsitektur jaringan *multi layer*. Tetapi berdampak pada proses komputasi yang cukup lama tergantung dari batas error dan jumlah epoch yang ditentukan dalam simulasi.
3. Akurasi pengenalan pola dapat ditentukan dengan mengatur *learning rate* yang tepat serta melakukan optimasi jumlah epoch.

## VI. DAFTAR PUSTAKA

- Koichi Ito, Ayumi Morita, 2005, Takafumi Aoki, Tatsuo Higuchi, Hiroshi Nakajima, and Koji Kobayashi, A Fingerprint Recognition Algorithm Using Phase-Based Image Matching for Low-Quality Fingerprints, IEEE.
- K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, and T. Higuchi, "A fingerprint matching algorithm using phaseonly correlation," *IEICE Trans. Fundamentals*, vol. E87-A, no. 3, pp. 682-691, Mar. 2004.
- Muslim, 2006, Fingerprint Pattern Recognition Using Distance Method Algorithm, Proceeding of the 2<sup>nd</sup> IMT-GT Regional Conference on Mathematics and Applications University Sains Malaysia, Penang, June 13-15.
- S.M. Ali, M.S. Al-Zewary, A New Fast Automatic Technique For Fingerprints Recognition And Identification, *Journal of Islamic Academy of Sciences* 10:2, 55-60, 1997
- Siang, Jong Jek. (2004). *Jaringan Syaraf Tiruan dan Pemogramannya dengan MATLAB*, Yogyakarta: Andi.
- Susmikanti, Mike, (2005). *Identifikasi Huruf Menggunakan Metode Pembelajaran Perceptron Dalam Jaringan Neural*, Prosiding Semiloka Teknologi Simulasi dan Komputasi serta Aplikasi, Pusat Pengembangan Teknologi Informasi dan Komputasi, BATAN.
- W.F Leung, S.H Leung, W.H Lau and Andrew Luk, 1999, Fingerprint Recognition Using Neural Network, IEEE.