

SIMULASI PENGENALAN KARAKTER MENGGUNAKAN NEURAL NETWORK PADA MATLAB

(SIMULATION OF CHARACTER RECOGNITION USING NEURAL NETWORK IN MATLAB)

Adi Sadli

STIKES Tanawali Persada Takalar

email: adisadli@gmail.com

ABSTRAK

Saat ini hampir semua aspek kehidupan telah tersentuh dengan komputerisasi. Salah satunya adalah bagaimana suatu sistem komputer bisa mengenali atau membaca karakter tertentu dari suatu citra digital. Misalnya saja bagaimana komputer bisa mengenali wajah seseorang atau menentukan tulisan tangan seseorang, mendiagnosa suatu penyakit dan lain-lain. Pengenalan pola (*pattern recognition*) merupakan salah satu cabang ilmu komputer yang dapat diartikan sebagai pengumpulan data-data mentah untuk dapat diklasifikasikan dengan maksud dan tujuan tertentu. Pengenalan pola ini bersifat conceptually driven processing yang berarti bahwa proses dimulai dari pembentukan konsep pada objek yang dijumpai (informasi dari memori).

Beberapa metode yang bisa digunakan untuk pengenalan pola adalah Jaringan Saraf Tiruan (JST), Metode Statistik, Metode Terstruktur dan lain sebagainya. Dengan JST, menganalogikan cara berfikir pada otak manusia. Jadi, informasi di proses sebagaimana otak manusia memproses sebuah informasi yang di dapat. Misalnya cara pengenalan wajah pada manusia. Sedangkan metode statistik berdasarkan hasil analisa yang sudah terkumpul. Misalnya pengenalan dan analisa penyakit pada manusia. Backpropagation Neural Network (BNN) mungkin adalah metode jaringan saraf tiruan yang paling banyak digunakan untuk pengenalan pola tertentu seperti tulisan tangan. Adapun Wavelet pada awalnya, transformasi *wavelet* digunakan untuk menganalisis sinyal bergerak (*non-stationary signals*). Sinyal bergerak ini dianalisis dalam transformasi *wavelet* dengan menggunakan teknik *multi-resolution analysis*. Secara umum teknik *multi-resolution analysis* adalah teknik yang digunakan untuk menganalisis frekuensi dengan cara frekuensi yang berbeda dianalisis menggunakan resolusi yang berbeda.

Penggunaan MATLAB *Neural NetworkToolbox* dalam sistem ini adalah merupakan teknik yang dipakai untuk memperoleh kesesuaian data antara data gambar masukan (*input*) dan data keluarannya (*output*) setelah melakukan beberapa pengujian (Train).

Kata Kunci : MATLAB, Pengenalan Pola, Image Processing, Jaringan Saraf Tiruan.

ABSTRACT

At present almost all aspects of life have been done by computerization. One of them is how a computer system can recognize or read certain characters from a digital image. For example, how a computer can recognize someone's face or determine someone's handwriting, diagnose a disease and others. Pattern recognition is one of the branches of computer science which can be interpreted as collecting raw data to be classified with specific intentions. This pattern recognition is conceptually driven processing which means that the process starts from the formation of concepts on objects encountered (information from memory).

Some methods that can be used for pattern recognition are Artificial Neural Networks (ANN), Statistical Methods, Structured Methods and so on. With ANN, it is analogous to the way of thinking in the human brain. So, information is processed as the human brain processes information obtained. For example, how to recognize faces in humans. While the statistical method is based on the results of the analysis that

has been collected. For example the introduction and analysis of diseases in humans. Backpropagation Neural Network (BNN) is probably the most widely used neural network method for recognition of certain patterns such as handwriting. As for Wavelet at first, wavelet transformations are used to analyze non-stationary signals. This mobile signal is analyzed in wavelet transformation using multi-resolution analysis techniques. In general, multi-resolution analysis techniques are techniques used to analyze frequencies by means of different frequencies analyzed using different resolutions.

The use of MATLAB Neural NetworkToolbox in this system is a technique used to obtain data compatibility between input data and output data after performing a number of tests.

Keywords: MATLAB, Pattern Recognize, Image Processing, Artificial Neural Networks

1. PENDAHULUAN

Pengenalan pola dalam istilah ilmu komputer terkhusus pada bidang pengolahan citra digital (image processing) merupakan teknik yang bertujuan untuk mengklasifikasikan citra digital yang telah diolah berdasarkan kesamaan atau kemiripan ciri yang dimilikinya. Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom, dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan elemen citra atau piksel (*picture element*) [1].

Pola adalah entitas yang terdefinisi dan dapat didefinisikan melalui ciri-cirinya (*feature*). [2] Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi sehingga pengelompokkan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Untuk pola huruf mempunyai ciri tinggi, tebal, titik sudut, lengkungan garis dan lain-lain [3]

Jaringan Saraf Tiruan (JST) adalah prosesor yang terdistribusi besar-besaran secara parallel yang dibuat dari unit proses sederhana, yang mempunyai kemampuan untuk menyimpan pengetahuan berupa pengalaman dan dapat digunakan untuk proses lain [5].

Jaringan saraf tiruan tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, ke dalam jaringan saraf tiruan dimasukkan pola-pola masukan (dan keluaran) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima [6].

Jaringan saraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan saraf biologis manusia, dengan asumsi bahwa:

- Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- Sinyal dikirimkan di antara neuron-neuron melalui penghubung-penghubung.
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.

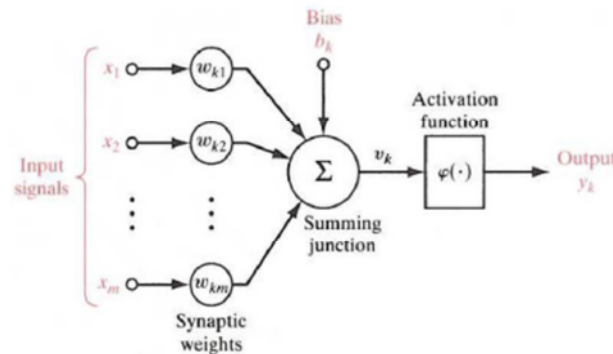
Untuk menentukan keluaran, Setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlah masukan yang diterima. Besarnya keluaran ini selanjutnya dibandingkan dengan suatu batas ambang.

Prinsip jaringan saraf tiruan (JST) ditentukan oleh tiga elemen dasar model saraf, yaitu:

1. Satu set dari sinapsis, atau penghubung yang masing-masing digolongkan oleh bobot atau kekuatannya.
2. Sebuah penambah untuk menjumlahkan sinyal-sinyal input. Ditimbang dari kekuatan sinaptik masing-masing neuron.

- Sebuah fungsi aktivasi untuk membatasi amplitudo output dari neuron. Fungsi ini bertujuan membatasi jarak amplitudo yang diperbolehkan oleh sinyal output menjadi sebuah angka yang terbatas.

Prinsip jaringan saraf tiruan secara sederhana digambarkan di bawah ini:



Gambar 1.1 Prinsip Dasar Jaringan Saraf Tiruan (JST)

Pada gambar di atas, Y menerima masukan dari neuron x_1 , x_2 , dan x_3 , dengan bobot hubungan masing-masing adalah w_1 , w_2 , dan w_3 . Ketiga impuls neuron yang ada dijumlahkan menjadi:

$$\text{Net} = x_1w_1 + x_2w_2 + x_3w_3$$

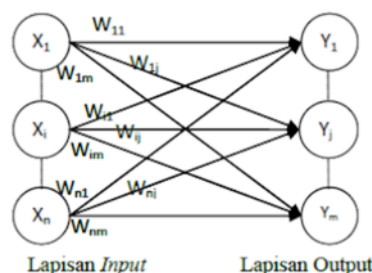
Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(\text{net})$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot [7].

Arsitektur Jaringan Saraf Tiruan

Pada jaringan saraf tiruan, neuron-neuron akan dikumpulkan dalam sebuah lapisan yang disebut dengan lapisan neuron (neuron layers). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan lainnya. Informasi yang didapatkan pada sebuah neuron akan disampaikan ke semua lapisan-lapisan yang ada, mulai dari lapisan masukan sampai dengan lapisan keluaran melalui lapisan tersembunyi (hidden layer). Pada jaringan saraf tiruan ini tiga lapisan bukanlah sebuah struktur umum karena beberapa jaringan saraf ada yang tidak memiliki lapisan tersembunyi.

[5] secara umum, ada tiga jenis arsitektur dari Jaringan Saraf Tiruan yaitu:

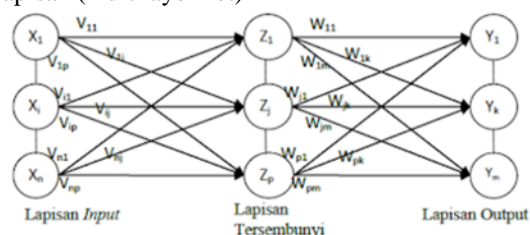
a. Jaringan dengan lapisan tunggal (single layer net)



Gambar 1.2. Jaringan dengan lapisan tunggal (single layer net)

Di dalam Jaringan Saraf Tiruan dengan satu layer, neuron-neuron diorganisasi dalam bentuk layer-layer. Dalam bentuk paling sederhana dari Jaringan Saraf Tiruan dengan satu layer, kita mempunyai sebuah input layer dari node sumber di mana informasi diproyeksikan ke output layer dari neuron tapi tidak bisa sebaliknya. Dengan kata lain, jaringan ini adalah tipe feed forward. Input layer dari node sumber tidak dihitung karena tidak ada perhitungan yang dilakukan.

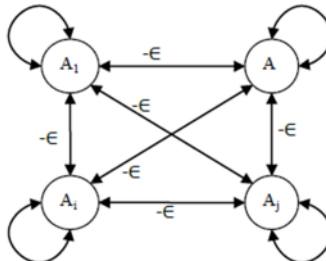
b. Jaringan dengan banyak lapisan (multilayer net)



Gambar 1.3. Jaringan dengan banyak lapisan (multilayer net)

Merupakan jaringan dengan satu atau lebih lapisan tersembunyi (hidden layer). Jaringan multi lapis ini memiliki kemampuan lebih dalam memecahkan masalah bila dibandingkan dengan jaringan lapis tunggal, namun pelatihannya mungkin lebih rumit. Pada beberapa kasus, pelatihan pada jaringan ini lebih baik karena memungkinkan bagi jaringan untuk memecahkan masalah yang tidak dapat diselesaikan jaringan berlapis tunggal karena jaringan tidak bisa dilatih untuk menampilkan secara benar.

c. Jaringan dengan lapisan kompetitif (competitive layer net)



Gambar 1.4. Jaringan dengan lapisan kompetitif (competitive layer net)

Bentuk lapisan kompetitif merupakan jaringan saraf tiruan yang sangat besar. Interkoneksi antar neuron pada lapisan ini tidak ditunjukkan pada arsitektur seperti jaringan yang lain. Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif atau sering pula disebut dengan prinsip winner takes all atau yang menanglah yang mengambil semua bagiannya.

2. METODE PENELITIAN

Penelitian ini menggunakan menggunakan metode deskriptif kuantitatif, Penelitian deskriptif kuantitatif merupakan penelitian dengan pendekatan menggunakan data sample atau populasi untuk mendapatkan informasi dari objek penelitian dimana sampel yang digunakan berbentuk angka [8]

Dalam pengenalan pola terbagi ke dalam dua fase proses pengenalan yaitu :

1. *Fase pelatihan*, pada fase ini beberapa contoh citra dipelajari untuk menentukan ciri yang akan digunakan dalam proses pengenalan serta prosedur klasifikasinya.
2. *Fase pengenalan*, pada fase ini citra diambil cirinya kemudian ditentukan kelas kelompoknya.

Pendekatan yang digunakan dalam sistem pengenalan pola yaitu pendekatan secara statistik dan pendekatan secara sintatik atau struktural.

1. Pendekatan Statistik

Pendekatan ini menggunakan teori-teori ilmu peluang dan statistik. Ciri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya. Pola yang berbeda memiliki distribusi yang berbeda pula. Dengan menggunakan teori keputusan di dalam statistik, digunakan distribusi ciri untuk mengklasifikasi pola.

2. Pendekatan sintatik

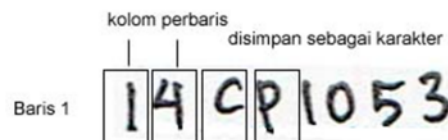
Pendekatan sintaktik adalah pendekatan untuk pengenalan pola menggunakan teori bahasa formal. Ciri-ciri yang terdapat pada suatu pola ditentukan primitif dan hubungan struktural antara primitif kemudian menyusun tata bahasanya. [2]

Adapun metode pengolahan citra yang digunakan yaitu :

- Segmentasi
Langkah pertama dalam analisis citra umumnya adalah melakukan proses segmentasi pada citra. Segmentasi digunakan untuk mengelompokkan komponen-komponen citra ke dalam suatu

kelompok berdasarkan kesamaan sifatnya. Proses segmentasi akan memisahkan objek atau bagian citra selanjutnya diisolasi untuk proses selanjutnya. Segmentasi adalah sebuah proses yang digunakan untuk memotong-motong gambar yang diproses menjadi beberapa bagian. Segmentasi merupakan proses mempartisi citra menjadi beberapa daerah atau objek. Segmentasi citra pada umumnya berdasar pada sifat *discontinuity* atau *similarity* dari intensitas piksel.

- Pendekatan *Discontinuity*
Mempartisi citra bila terdapat perubahan intensitas secara tiba-tiba (*edge based*)
- Pendekatan *Similarity*
Mempartisi citra menjadi daerah-daerah yang memiliki kesamaan sifat tertentu (*region based*). [4]



Gambar 2.1. Contoh Proses Segmentasi



Gambar 2.2. Keluaran Segmentasi

- Threshold
 - Threshold adalah suatu proses yang digunakan untuk menghasilkan citra biner yaitu citra dengan hanya dua warna yaitu : hitam dan putih. Operator ini memilih piksel yang memiliki nilai tertentu atau lingkup tertentu. Proses ini dapat dilakukan apabila kita telah mengetahui *brightness level* (atau *contrast*) dari gambar tersebut. Bentuk teknik thresholding ada 2 macam, yaitu *Uniform Thresholding* dan *Adaptive Thresholding*. Di dalam *uniform thresholding* metode yang digunakan adalah dengan menentukan suatu batas level, yang nantinya akan dipergunakan untuk menentukan warna piksel. Piksel yang levelnya lebih dari threshold level akan dirubah menjadi putih dan sebaliknya piksel yang levelnya ada di bawah dari *level threshold* akan dirubah menjadi hitam.
- Filter Noise Eliminator
 - Image filtering sering pula dinamakan proses penghalusan gambar yang utamanya digunakan untuk mengurangi efek-efek yang tidak diinginkan yang muncul dalam sebuah gambar digital sebagai akibat dari kesalahan sistem sampling atau sebab lain.
 - Pemfilteran adalah suatu cara untuk mengekstrasi bagian data tertentu dari suatu himpunan data, dengan menghilangkan bagian-bagian data yang tidak diinginkan. *Filter* dalam pengolahan citra (secara khusus disebut *filter* digital) dirancang untuk ‘menyaring’ informasi spektral, sehingga menghasilkan citra baru yang mempunyai variasi nilai spektral yang berbeda dari citra asli.
- Thinning
 - Thinning merupakan operasi morfologis yang digunakan untuk menghilangkan piksel *foreground* tertentu dari citra. Operasi morfologis biasanya digunakan untuk mengambil struktur atau bentuk dari citra atau mengidentifikasi sebuah

objek atau batas dari sebuah citra. Berperan penting pada aplikasi-aplikasi penglihatan pada mesin (*machine vision*) dan deteksi objek otomatis. Biasanya operasi morfologis dilakukan pada citra biner (hanya terdiri dari piksel 0 dan 1, hitam dan putih, dan lain-lain).

- Thinning dapat diterapkan dalam beberapa hal, terutama untuk *skeletonization*. Dalam hal ini, thinning dipakai untuk merapikan hasil dari deteksi sisi dengan menipiskan semua garis sampai lebarnya hanya satu piksel. Seperti halnya operasi morfologis lainnya maka thinning biasanya diterapkan pada citra biner dan menghasilkan citra biner yang lain sebagai hasil. Thinning dapat diaplikasikan untuk objek-objek yang terdiri dari garis-garis (lurus atau melengkung/kurva). Metode ini tidak berlaku untuk objek yang memiliki bentuk yang menutupi daerah luas. Thinning lebih banyak merupakan suatu proses penengah, untuk mempersiapkan objek agar bisa dianalisa. Proses-proses berikutnya menentukan sifat-sifat dari kerangka tersebut. Ada banyak algoritma thinning hanya saja *Template-Based Mark and Delete Thinning Algorithms*-lah yang sangat populer karena kehandalannya dan efektifitasnya. Tipe proses thinning ini menggunakan *templates*, dimana pasangan yang cocok dengan *template* pada citra menghapus tengah-tengah piksel.
- Algoritmanya bersifat iteratif, yang mengikis (*erodes*) lapisan luar dari piksel sampai tidak ada lagi lapisan yang dapat dihilangkan.

3. ANALISIS DAN PENGUJIAN

3.1 Spesifikasi Perangkat Keras

- a. Processor : CPU Min. 1.0 GHZ
- b. Memory : Minimum 2 MB RAM
- c. Harddisk : Kapasitas Min. 200 GB

3.2 Perangkat Lunak yang digunakan

Simulasi pengenalan karakter ini menggunakan aplikasi MATLAB Ver. R2013a

3.3 Hasil

Setelah melakukan proses ekstraksi citra digital yang meliputi *pre-processing* sampai *post-processing* untuk mendapatkan pola karakter dalam bentuk matriks. Keluaran dari proses pembacaan karakter ini merupakan fitur yang direpresentasikan dalam bentuk matriks Selanjutnya fitur ini akan menjadi masukan bagi jaringan *Neural Network* yang bertugas sebagai pengenalan karakter.

a	b	c	d	e	f	g	h	i
j	k	l	m	n	o	p	q	r
s	t	u	v	w	x	y	z	.
A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	=
0	1	2	3	4	5	6	7	8
9	?	-	:	+	-	=	/	*

Gambar 3.3.1 Contoh *Template* Karakter Huruf yang akan di ekstraksi menjadi matriks

Format fitur keluaran dari karakter telah ditetapkan berdasarkan posisi segmen pada karakter. Nilai pada fitur karakter akan bernilai 1 bila fitur tersebut ada dalam karakter, dan bernilai 0 untuk sebaliknya.

Untuk dapat melakukan proses pengenalan, jaringan *Neural Network (NN)* perlu mendapatkan training terlebih dahulu. Proses training jaringan NN dilakukan agar sistem pengenalan karakter dapat mengenali citra karakter dengan tepat. Masukan adalah sejumlah tertentu citra yang terdiri karakter huruf kecil dan huruf besar masing-masing diwakili dengan jumlah contoh citra yang diambil dari berbagai sumber. Sedangkan untuk tanda baca dan angka masing-masing diwakili dengan jumlah contoh citra.

Masukan sistem berupa fitur-fitur dari karakter yang direpresentasikan dengan matrik

Target merupakan keluaran yang diharapkan dari hasil training.

Keluaran sistem berupa sebuah jaringan *neural network* yang telah ditraining. Sehingga jaringan ini nantinya dapat digunakan pada sistem pengenalan karakter dengan cara mengenali fitur masukan sistem pengenalan karakter untuk dicocokkan dengan contoh fitur yang telah melalui proses training.

Sebagai contoh didapatkan matriks dari hasil pengenalan karakter yaitu [9]:

$x = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8];$

Dan ada juga punya matriks lain, sebut matriks t :

$t = [0 \ 0.84 \ 0.91 \ 0.14 \ -0.77 \ -0.96 \ -0.28 \ 0.66 \ 0.99];$

Matriks x adalah input untuk JST dan t adalah targetnya. Kita akan mengajari x agar mencapai hasil seperti t .

Membuat Arsitektur jaringannya, untuk simulasi ini yang digunakan adalah Algoritma Jaringan Syaraf Tiruan (JST) *Backpropagation*

Code Perintah :

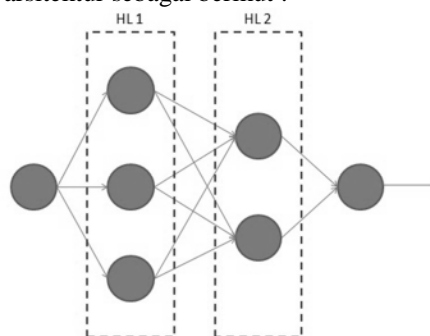
```
net = newff(x,t,[S1 S2...S(N-1)],{xF1 xF2...xFNI},BtF,BLF,tF,ItF,OtF,DDF)
```

atau

```
net = newff(x,t,[S1 S2...S(N-1)])
```

Penjelasan:

x dan t adalah input dan Target JST, sedangkan $S1 \ S2...S(N-1)$ adalah jumlah neuron dalam layer JST. Asumsinya akan membentuk arsitektur sebagai berikut :



Gambar 3.2 Arsitektur JST

Membuat program JST:

Ketik pada **command windows** di **MATLAB** :

```
% input
x = [0 1 2 3 4 5 6 7 8];
% target
t = [0 0.84 0.91 0.14 -0.77 -0.96 -0.28 0.66 0.99];
% bangun JST
newff(x,t,[3 2]);
```

secara sintaksis di tuliskan $JST = newff(x,t,[3 \ 2]);$

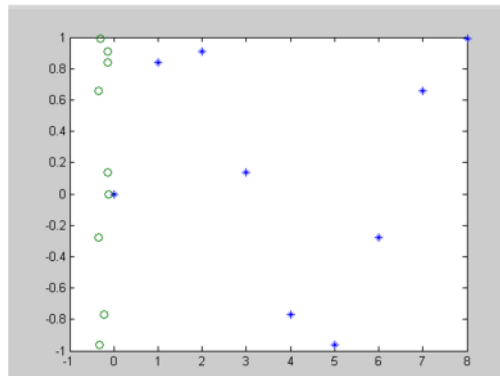
Mensimulasikan JST, karena pada saat ini Matlab sudah membangkitkan nilai-nilai bias dan bobot awal pada tiap-tiap koneksi dan neuron.

Sintax:

```
y = sim(JST,x);
```

Hasilnya dapat diplot dengan perintah berikut:

```
plot(x,t,'*',y,t,'o');
```



Gambar 3.3 Hasil Ujicoba

Target yang ingin dicapai adalah lambang bintang (*), dan hasil simulasi diperlihatkan dengan tanda lingkaran (o).

Hasilnya akan berbeda untuk tiap simulasi, karena MATLAB akan membangkitkan bilangan acak. Dengan kata lain, bobot dan bias yang dihasilkan akan berbeda, sehingga hasil akhirnya akan berbeda pula.

Melatih JST agar mencapai target yang diinginkan.

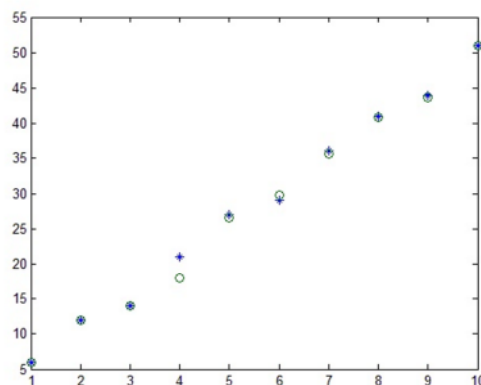
Sintak yang digunakan sebagai berikut:

```
JST.trainParam.epochs = 5;
```

```
JST = train(JST,x,t);
```

Maksud sintaks ini adalah melatih JST sebanyak 5 iterasi (epoch) maju-mundur. Kemudian akan muncul jendela *nntraintool*.

Agar JST kita lebih pintar, kita harus melatihnya lebih banyak data lagi. Sampai didapat hasil yang mendekati.



Gambar 3.4 Hasil Ujicoba yang diharapkan

4. KESIMPULAN

1. Untuk mendapatkan hasil yang lebih baik, harus dilakukan pengujian dengan menggunakan lebih banyak uji/training data.
2. Tampilan simulasi pengujian yang digunakan adalah model pengujian dengan menggunakan perintah Neural Network pada MATLAB dimana hasil pengujian/train pengenalan karakternya tidak diperlihatkan dalam bentuk GUI (Graphical User Interface) hanya dalam bentuk grafik.

5. SARAN

Berdasarkan hasil dari analisa, hasil belum sepenuhnya memuaskan sehingga penulis memberi saran untuk pengembangan lebih lanjut antara lain:

1. Perlu dicoba metode ekstraksi ciri (pengenalan pola) yang lain sehingga didapatkan ciri yang benar-benar mewakili suatu pola, dengan demikian dapat meningkatkan prosentase kebenaran pengenalan pola
2. Program dapat juga dikembangkan dengan metode pengenalan yang lain, misalnya Metode bayes, Mahalanobis distance atau juga dengan jaringan syaraf tiruan, untuk mengurangi kesalahan pengenalan.

DAFTAR PUSTAKA

- [1]. Gonzales, Rafael C and Woods. 1992. Digital Image Processing. Addison-Wesley Publishing Company.
- [2]. Munir, Rinaldi. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Penerbit Informatika. 2004
- [3]. Suryarini (repository. gunadarma.ac. id : 8000 / Image Processing_Suryarini_Widodo(6) 253_258_780.pdf)
- [4]. Murti, Aniarti, DR.2004. Segmentasi Citra,
<http://www.cs.ui.ac.id/WebKuliah/citra/2004/segmentasi.ppt>. 20 April 2011
- [5]. Haykin. S. 2009. Nural Networks and Learning Machines. United State of America: Pearson.
- [6]. Puspitaningrum, Diyah. 2006. Pengantar Jaringan Syaraf Tiruan. Yogyakarta: Andi.
- [7]. Siang, JJ. 2004. Aplikasi Jaringan Syaraf Tiruan dan Pemrograman Menggunakan MATLAB. Yogyakarta: Andi.
- [8]. Sugiyono, Statistik Untuk Penelitian, Bandung, Jawa barat: Alfabeta, 2011
- [9]. Mathwork, <http://www.mathworks.com/help/deeplearning/ref/train.html>; jsessionid=ed607c9948d33138261f503d581d;