

## ALGORITMA BACK PROPAGATION NEURAL NETWORK UNTUK PENGENALAN POLA KARAKTER HURUF JAWA

Nazla Nurmila, Aris Sugiharto, dan Eko Adi Sarwoko  
Prodi Ilmu Komputer Jurusan Matematika F. MIPA UNDIP

### Abstract

Back Propagation Neural Network (BPNN) is a type of algorithm in Neural Network that can be use for Javanese alphabets character recognition. Matlab 7.1 has been used as a software to support the program. The main purpose of this research is order to find out BPNN's training characteristic from each samples. On the other hand, this research also gives BPNN's accurancy value in Javanese alphabets character recognition.

The result of research shows that each part of the samples having different BPNN's characteristic based on the best training.

Keywords : NN, BPNN

### 1. Pendahuluan

Seiring dengan berkembangnya teknologi, komputer diharapkan memiliki kemampuan untuk mengerjakan segala sesuatu yang dapat dikerjakan oleh manusia. Agar komputer dapat bertindak seperti dan sebaik manusia, diperlukan beberapa metode untuk membekali komputer agar menjadi mesin yang pintar. Salah satunya yaitu dengan mengimplementasikan *Neural Network (NN)* pada komputer.

*NN* merupakan sebuah sistem pembelajaran terhadap penerimaan informasi yang memiliki kinerja layaknya sebuah jaringan syaraf pada manusia. *NN* diimplementasikan dengan menggunakan program komputer sehingga mampu menyelesaikan sejumlah proses perhitungan. Salah satu penggunaan *NN* adalah untuk pengenalan pola. Sistem pengenalan pola merupakan komponen penting dalam proses peniruan cara kerja sistem manusia

Dalam pengenalan pola karakter, banyak jenis karakter yang dapat dikenali melalui komputer dengan menggunakan berbagai algoritma. Pada tahun 2005, Reki Zamasari telah berhasil membuat program aplikasi pengenalan pola karakter huruf Jawa dengan algoritma *Kohonen Neural Network (KNN)*. Di tahun yang sama, Agung Nugroho berhasil membuat program aplikasi pengenalan pola karakter huruf Jawa dengan menggunakan algoritma *Learning Vector Quantization (LVQ)*. Algoritma dalam penelitian ini digunakan *Back Propagation*

*Neural Network (BPNN)* untuk mengenali pola karakter huruf Jawa.

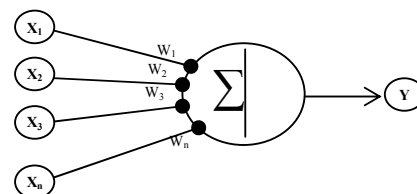
Karakter huruf Jawa yang terdiri atas 20 merupakan pola karakter kompleks. Hal ini disebabkan karena pola karakter tersebut memiliki banyak kemiripan antar setiap individu karakter.

Untuk mengetahui karakteristik *BPNN*, pelatihan dilakukan pada masing-masing bagian sampel. Data sampel terbagi atas 6 jenis, yaitu 1 sampel, 3 sampel, 5 sampel, 8 sampel, 10 sampel, dan 15 sampel. Selain itu, penelitian ini juga memberikan nilai keakuratan *BPNN* dalam mengenali pola karakter huruf Jawa

### 2. Tinjauan Pustaka

#### 2.1. *Neural Network (NN)*

*NN* 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 (*Rajasekaran, 2005*).



Gambar 1. Proses Komunikasi Antar Neuron

Pada Gambar 1 diperlihatkan bahwa *NN* terdiri atas satuan-satuan pemroses berupa *neuron*. *Y* sebagai *output* menerima *input* dari *neuron*  $X_1, X_2, X_3, \dots, X_n$  dengan bobot  $W_1, W_2, W_3, \dots, W_n$ . Hasil penjumlahan seluruh impuls *neuron* dibandingkan dengan nilai ambang tertentu melalui fungsi aktivasi  $f$  setiap *neuron*. Fungsi aktivasi digunakan sebagai penentu keluaran suatu *neuron*.

Salah satu metode pelatihan dalam *NN* adalah pelatihan terbimbing (*supervised learning*). Pada pelatihan terbimbing diperlukan sejumlah masukan dan target yang berfungsi untuk melatih jaringan hingga diperoleh bobot yang diinginkan.

Pada setiap kali pelatihan, suatu *input* diberikan ke jaringan. Jaringan akan memproses dan mengeluarkan keluaran. Selisih antara keluaran jaringan dengan target merupakan kesalahan yang terjadi. Jaringan akan memodifikasi bobot sesuai dengan kesalahan tersebut. *BPNN* merupakan metode yang menggunakan *supervised learning*.

## 2.2. Back Propagation Neural Network (BPNN)

*BPNN* merupakan algoritma pelatihan terbimbing yang mempunyai banyak lapisan. *BPNN* 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.

Syarat fungsi aktivasi dalam *BPNN* adalah bersifat kontinu, terdifferensial dengan mudah, dan merupakan fungsi yang tidak turun. Fungsi aktivasi yang dapat memenuhi ketiga syarat tersebut adalah logsig, tansig, dan purelin.

Metode pengenalan merupakan proses inialisasi data yang akan diolah selanjutnya oleh *BPNN*. Data yang akan dikenali disajikan dalam bentuk vektor. Masing-masing data mempunyai target yang disajikan juga dalam bentuk vektor. Target atau keluaran acuan merupakan suatu peta karakter yang menunjukkan lokasi dari vektor masukan.

Sedangkan metode pelatihan merupakan proses latihan mengenali data dan menyimpan pengetahuan atau informasi yang didapat ke dalam bobot-bobot (*Heaton, 2003*).

Terdapat 3 fase dalam pelatihan *BPNN*, yaitu fase maju (*feed forward*), fase mundur (*back propagation*), dan fase modifikasi bobot. Dalam fase *feed forward*, pola masukan dihitung maju dimulai dari lapisan *input* hingga lapisan *output*. Dalam fase *back propagation*, tiap-tiap unit *output* menerima target pola yang berhubungan dengan pola *input* untuk dihitung nilai kesalahan. Kesalahan tersebut akan dipropagasikan mundur. Sedangkan fase modifikasi bobot bertujuan untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang secara terus menerus hingga kondisi penghentian dipenuhi.

Berikut ini merupakan algoritma dalam pelatihan *BPNN* (*Kusumadewi, 2004*) :

<b>Langkah 0</b>	: $\hookrightarrow$ Inialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil). $\hookrightarrow$ Tetapkan : Maksimum Epoch, Target Error, dan Learning Rate ( ). $\hookrightarrow$ Inialisasi : Epoch = 0, MSE = 1.
<b>Langkah 1</b>	: $\hookrightarrow$ Kerjakan langkah-langkah berikut selama (Epoch < Maksimum Epoch) dan (MSE > Target Error) : $\hookrightarrow$ Epoch = Epoch + 1.
<b>Langkah 2</b>	: Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan langkah-langkah berikutnya.

### Fase I : Feed Forward

<p><b>Langkah 3</b> : Tiap-tiap unit input (<math>X_i, i=1,2,\dots,n</math>) menerima sinyal <math>X_i</math> dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).</p> <p><b>Langkah 4</b> : Tiap-tiap unit pada suatu lapisan tersembunyi (<math>Z_j, j=1,2,\dots,p</math>) menjumlahkan sinyal-sinyal input terbobot :</p> $Z_{in_j} = b1_j + \sum_{i=1}^n X_i W_{ij}$ <p>gunakan fungsi aktivasi untuk menghitung sinyal output :  <math>Z_j = f(Z_{in_j})</math>                  dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).</p> <p><b>Langkah 5</b> : Tiap-tiap unit output <math>Y_k (k=1,2,\dots,m)</math> menjumlahkan sinyal-sinyal input terbobot :</p> $Y_{in_k} = b2_k + \sum_{j=1}^p Z_j W_{jk}$ <p>gunakan fungsi aktivasi untuk menghitung sinyal output :  <math>Y_k = f(Y_{in_k})</math>                  dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).</p>
---

**Fase II : Back Propagation**

<p><b>Langkah 6</b> : <math>\zeta</math> Tiap-tiap unit output <math>Y_k (k=1,2,\dots,m)</math> menerima target pola yang berhubungan dengan pola input pelatihan, hitung informasi errornya :  <math>\delta = (t_k - Y_k) f'(Y_{in_k})</math>  <math>\varphi 2_{jk} = \delta_k Z_j</math>  <math>\beta 2_k = \delta_k</math></p> <p><math>\zeta</math> Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai ) :  <math>\Delta W_{jk} = \alpha \varphi 2_{jk}</math></p> <p><math>\zeta</math> Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai ) :  <math>\Delta b 2_k = \alpha \beta 2_k</math></p> <p><b>Langkah 7</b> : <math>\zeta</math> Tiap-tiap unit tersembunyi <math>Z_j (j=1,2,\dots,p)</math> menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya ) :  <math>\delta_{in_j} = \sum_{k=1}^m \delta 2_k W_{jk}</math></p> <p><math>\zeta</math> Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error :  <math>\delta 1_j = \delta_{in_j} f'(Z_{in_j})</math>  <math>\varphi 1_{ij} = \delta 1_j X_i</math>  <math>\beta 1_j = \delta 1_j</math></p> <p><math>\zeta</math> Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai ) :  <math>\Delta W_{ij} = \alpha \varphi 1_{ij}</math></p> <p><math>\zeta</math> Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai ) :  <math>\Delta b 1_j = \alpha \beta 1_j</math></p>
--

**Langkah 8** :  $\nabla$  Tiap-tiap unit output  $Y_k$  ( $k=1,2,\dots,m$ ) memperbaiki bias dan bobotnya ( $j=0,1,2,\dots,p$ ) :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk}$$

$$b_{2k}(\text{baru}) = b_{2k}(\text{lama}) + \Delta b_{2k}$$

$\nabla$  Tiap-tiap unit tersembunyi  $Z_j$  ( $j=1,2,\dots,p$ ) memperbaiki bias dan bobotnya ( $i=0,1,2,\dots,n$ ) :

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij}$$

$$b_{1j}(\text{baru}) = b_{1j}(\text{lama}) + \Delta b_{1j}$$

**Langkah 9** : Hitung MSE.

**2.3. Pemrograman BPNN dengan Matlab 7.1**

Matlab 7.1 mempunyai *toolbox NN* yang dapat digunakan dalam menyelesaikan permasalahan BPNN. Untuk menginisialisasi jaringan, perintah yang digunakan adalah *newff*. Format yang diberikan oleh *toolbox* Matlab 7.1 adalah sebagai berikut :

```
net=newff(minmax(P),[S1 S2...SN],{TF1 TF2...TFN},BTF,BLF,PF)
```

Setiap kali membentuk jaringan, Matlab 7.1 akan memberikan nilai bobot awal dengan bilangan random kecil. Bobot akan berubah setiap kali menginisialisasi jaringan. Akan tetapi, jika diinginkan memberi bobot tertentu, dapat menggunakan perintah *net.IW{i,j}* dan *net.LW{j,k}*.

Setelah menginisialisasi jaringan dan bobot, hasilnya kemudian disimulasikan dengan *input*

yang sama dengan *input* data pelatihan. Simulasi jaringan merupakan proses untuk menghitung keluaran jaringan. Perintah *sim* dalam Matlab 7.1 dapat digunakan untuk melakukan simulasi. Formatnya adalah  $Y = \text{sim}(\text{net},P)$  Sedangkan untuk melakukan pelatihan BPNN, digunakan fungsi *train* yang formatnya adalah  $\text{net} = \text{train}(\text{net}, P, T)$

**3. Pembahasan dan Hasil Penelitian**

**3.1. Persiapan Data Pelatihan**

Data pelatihan yang digunakan terbagi atas 6 bagian. Masing-masing bagian menunjukkan banyaknya jumlah sampel pola. Tujuan dari pembagian data pelatihan agar dapat membandingkan hasil pelatihan pada masing-masing bagian. Selain itu, untuk membandingkan nilai keakuratan proses pengujian berdasarkan hasil pelatihan. Keenam bagian tersebut akan ditunjukkan dalam Tabel 1.

Tabel 1. Pembagian Data Pelatihan

No.	Bagian	Sampel ke -														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.	1 Sampel	√														
2.	3 Sampel	√	√	√												
3.	5 Sampel	√	√	√	√	√										
4.	8 Sampel	√	√	√	√	√	√	√	√							
5.	10 Sampel	√	√	√	√	√	√	√	√	√	√					
6.	15 Sampel	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

**3.2. Pengolahan Citra**

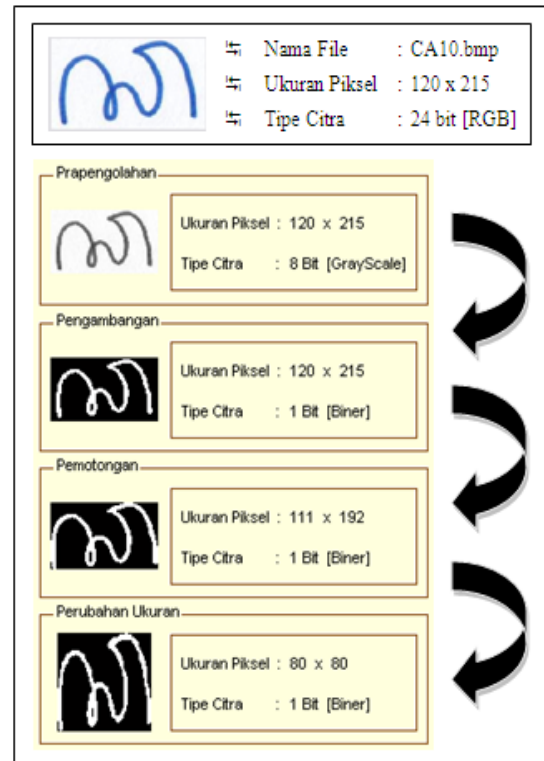
Pola karakter yang digunakan seringkali memiliki tingkat kualitas yang rendah sehingga

tidak dapat menghasilkan suatu *output* yang dikenali dengan baik oleh komputer. Agar pola karakter dapat direpresentasikan dengan baik, proses pengolahan citra dapat digunakan untuk memperbaiki kualitas citra. Selain itu, untuk dapat diproses oleh *BPNN*, data pelatihan yang digunakan harus memiliki tipe citra dan ukuran *pixel* yang sama.

Tahap-tahap dan contoh pada proses pengolahan citra yang digunakan dalam merepresentasikan pola karakter huruf Jawa terlihat pada Gambar 2 dan Gambar 3.



Gambar 2. Tahap-Tahap Proses Pengolahan Citra



Gambar 3. Contoh Proses Pengolahan Citra Pola Karakter Huruf Jawa

### 3.3. Pelatihan Pola Karakter Huruf Jawa dengan *BPNN*

Pelatihan pola karakter huruf Jawa dilakukan secara berulang-ulang dengan menggunakan data pelatihan, jumlah *neuron* dan parameter jaringan berupa *learning rate* dan momentum yang berbeda-beda. Sedangkan untuk fungsi pelatihan, fungsi evaluasi, fungsi aktivasi, dan parameter jaringan lainnya selain *learning rate* dan momentum bersifat tetap.

Tujuan dari pelatihan yang berulang-ulang ini adalah untuk mendapatkan karakteristik *BPNN* yang terbaik sehingga *BPNN* tersebut dapat mempelajari pola yang diberikan dengan baik.

Adapun batasan-batasan dalam melakukan pelatihan pola karakter huruf Jawa ini antara lain adalah :

1. Data pelatihan menggunakan 1 sampel, 3 sampel, 5 sampel, 8 sampel, 10 sampel, dan 15sampel.

2. Fungsi pelatihan yang digunakan adalah  $\text{traingdx}$ .
3. Fungsi evaluasi yang digunakan adalah  $\text{sse}$ .
4. Banyaknya jumlah *neuron* yang digunakan adalah 5, 15, dan 25.
5. Fungsi aktivasi yang digunakan pada lapisan tersembunyi dan lapisan output kedua-duanya adalah  $\text{logsig}$ .
6. Jumlah epoch maksimum adalah 3000.
7. Pilihan untuk nilai learning rate adalah 0.01, 0.03, dan 0.05.
8. Nilai momentum yang digunakan adalah 0.5, 0.7, dan 0.9
9. Nilai yang diberikan untuk kenaikan dan penurunan learning rate masing-masing adalah 1.05 dan 0.7.
10. Proses pelatihan dihentikan apabila sudah tercapai goal yang diinginkan, yaitu 0.1.
11. Untuk setiap data pelatihan dilakukan beberapa kali pelatihan dan diambil 1 yang terbaik untuk dijadikan sebagai acuan proses pengujian.
12. Hasil terbaik dari pelatihan ditentukan berdasarkan nilai squared error terkecil dan berakhir epoch yang paling kecil.

### 3.4. Pengujian Hasil Pelatihan

Langkah pertama dalam melakukan proses pengujian adalah mempersiapkan data

pengujian. Data pengujian tersusun atas 20 kelas pola karakter huruf Jawa. Masing-masing kelas mempunyai 15 sampel pola. Sehingga terdapat 300 pola yang digunakan untuk mempersiapkan data pengujian. Data pelatihan yang telah terbagi atas 6 bagian dan masing-masing telah memiliki hasil pelatihan akan diikuti sertakan dalam mempersiapkan data pengujian.

Langkah kedua adalah menentukan pola karakter yang akan diuji. Dalam hal ini, tidak ada aturan khusus dalam menentukan pola karakter mana yang akan diuji terlebih dahulu.

Langkah selanjutnya adalah proses pengolahan citra. Proses yang dilakukan yaitu prapengolahan, pengembangan, pemotongan, dan perubahan ukuran *pixel*.

Sedangkan proses pengenalan pola merupakan proses untuk menentukan nilai keakuratan *BPNN* dalam mengenali pola karakter huruf Jawa. Pola karakter berupa data pengujian akan disimulasikan berdasarkan hasil pelatihan. Pola karakter yang memiliki nilai simulasi terbesar merupakan pola karakter yang dikenali oleh *BPNN*.

Tabel 2 merupakan tabel seluruh hasil pengujian.

Tabel 2. Nilai Keakuratan Pengujian Pola Karakter Huruf Jawa

No.	Nilai Keakuratan (%)					
	1 sampel	3 sampel	5 sampel	8 sampel	10 sampel	15 sampel
1.	100	100	100	100	100	100
2.	10	100	100	100	100	100
3.	0	100	100	95	95	95
4.	5	55	100	100	100	100
5.	30	25	100	100	95	100
6.	10	50	55	100	100	100
7.	15	50	50	100	100	95
8.	5	35	50	100	100	100
9.	5	80	85	85	100	100
10.	15	70	70	90	100	100
11.	100	90	100	95	100	100
12.	5	40	45	70	80	100
13.	10	70	90	90	100	100
14.	15	65	75	85	75	95
15.	15	55	30	70	65	100
Rata-Rata Data Pelatihan	100	100	100	99.375	99	99
Rata-Rata Diluar Data Pelatihan	17.143	57.083	65	83.571	84	-
Rata-Rata Seluruh Data	22.667	65.667	76.667	92	94	99

Terlihat pada Tabel 2, sampel yang diarsir warna biru merupakan data pengujian yang digunakan pada saat melakukan proses pelatihan. Dengan kata lain, arsiran warna biru pada Tabel 2 menunjukkan banyaknya sampel yang digunakan pada saat pelatihan.

### 3.5. Analisis Hasil Pelatihan dan Pengujian Pola Karakter Huruf Jawa

Terdapat 6 bagian dalam pelatihan pola karakter huruf Jawa. Masing-masing bagian

memiliki karakteristik *BPNN* yang berbeda-beda. Hasil pelatihan yang ditunjukkan mempunyai nilai yang berbeda-beda pula. Berdasarkan hasil pelatihan tersebut, diambil 1 yang terbaik untuk dijadikan sebagai acuan proses pengujian. Seluruh karakteristik *BPNN* dengan hasil pelatihan terbaik pada masing-masing bagian sampel beserta hasil pengujiannya ditunjukkan dalam Tabel 3.

Tabel 3. Karakteristik Pelatihan BPNN Terbaik dan Hasil Pengujian

No.	Data Sampel (i)	Karakteristik BPNN			Rata-Rata Pengujian (%)		
		Jumlah Neuron	LR	Momentum	Data Pelatihan (Data Sampel i)	Di luar Data Pelatihan (Data Sampel i+1 s/d 15)	Seluruh Data (Data Sampel 1 s/d 15)
1.	1 Sampel	25	0.03	0.7	100	17.143	22.667
2.	3 Sampel	15	0.01	0.5	100	57.083	65.667
3.	5 Sampel	15	0.03	0.9	100	65	76.667
4.	8 Sampel	25	0.01	0.5	99.375	83.571	92
5.	10 Sampel	15	0.01	0.5	99	84	94
6.	15 Sampel	25	0.01	0.5	99	-	99
Rata-Rata					99.563	61.359	75

Berdasarkan Tabel 3, dapat ditunjukkan bahwa :

1. Masing-masing bagian sampel memiliki karakteristik BPNN yang berbeda-beda untuk mendapatkan hasil pelatihan yang paling baik.
2. Semakin banyak data pelatihan, semakin baik nilai keakuratan BPNN dalam mengenali seluruh pola karakter huruf Jawa, baik data pelatihan yang ikut diuji maupun yang tidak ikut diuji.  
Hal ini terlihat pada rata-rata nilai keakuratan seluruh data pengujian yang bernilai 22.667%, 65.667%, 76.667%, 92%, 94%, dan 99%.
3. Semakin banyak data pelatihan, semakin kecil nilai keakuratan BPNN dalam mengenali dirinya sendiri. Data pengujian ini merupakan data pelatihan. Akan tetapi, nilai keakuratan tersebut tidak menunjukkan hasil yang terlalu signifikan.  
Hal ini terlihat pada rata-rata nilai keakuratan data pelatihan yang bernilai 100%, 100%, 100%, 99.375%, 99%, dan 99%.
4. Semakin banyak data pelatihan, semakin baik BPNN mengenali pola karakter huruf Jawa yang tidak ikut dilatih dalam proses pelatihan.  
Hal ini terlihat pada rata-rata nilai keakuratan di luar data pelatihan yang

memberikan 17.143%, 57.083%, 65%, 83.571%, dan 84%.

#### 4. Kesimpulan

Penelitian ini menunjukkan bahwa masing-masing bagian sampel memiliki karakteristik BPNN yang berbeda-beda untuk mendapatkan hasil pelatihan yang paling baik.

Sedangkan rata-rata keakuratan BPNN dalam mengenali pola karakter huruf Jawa adalah sebesar 99.563% untuk data sampel berupa data pelatihan, 61.359% untuk data sampel diluar data pelatihan, dan 75% untuk data sampel data pelatihan dan di luar data pelatihan.

#### 5. Daftar Pustaka

- [1] P. Nugraha dan A. B. Mutiara, “Metode Ekstraksi Data untuk Pengenalan Huruf dan angka Tulisan Tangan dengan Menggunakan Jaringan Syaraf Buatan Balik”, <http://www.gunadarma.ac.id>, 2004. (diakses terakhir pada tanggal 12 Maret 2007 jam 15.56 WIB).
- [2] Heaton, J., “Introduction to Neural Network with Java”, <http://www.heatonresearch.com/articles/6/page2.html>, 2003. (diakses terakhir pada tanggal 28 Agustus 2007 jam 14.35 WIB). Kusumadewi, Sri, “Artificial Intelligence



- (Teknik dan Aplikasinya)", Graha Ilmu, Yogyakarta, 2003.
- [3] Kusumadewi, Sri, "Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK", Graha Ilmu, Yogyakarta, 2004.
- [4] Munir, Rinaldi, "Pengolahan Citra Digital dengan pendekatan algoritmik", Informatika, Bandung, 2004.
- [5] Murni, Aniati, "Pengantar Pengolahan Citra", Elex Media Komputindo, Jakarta, 1992.
- [6] Nugroho, Agung, "Pengenalan Pola Karakter Huruf Jawa menggunakan Jaringan Syaraf Tiruan LVQ (Learning Vector Quantization)", Tugas Akhir-Perpustakaan F. MIPA UNDIP, Semarang, 2005.
- [7] Nur Yusuf Oktavia, Ahmad Zakky, Riyanto Sigit, Miftahul Huda, "Aplikasi Jaringan Syaraf Tiruan untuk Pengenalan Huruf pada Pengolahan Citra Digital Berbasis Web", Makalah Tugas Pendahuluan Proyek Akhir, Politeknik Elektronika Negeri Surabaya-Institut Teknologi Sepuluh Nopember Surabaya, 2002.
- [8] Prabowo, Anindito, "Perbandingan antara Metode Kohonen Neural Network dengan Metode Learning Vector Quantization pada Pengenalan Pola Tandatangan", Tugas Akhir-Perpustakaan F. MIPA UNDIP, Semarang, 2007.
- [9] Rajasekaran S.,GA. Vijayalakshmi Pai, "Neural Network, Fuzzy Logic and Genetic Algorithms", Prentice-Hall of India, New Delhi, 2005.
- [10] Samudera, Istyawan, "Pencocokan Citra Sidik Jari dengan Jaringan Syaraf Tiruan Back Propagation dan Wavelet menggunakan GUI Matlab 6.5", Tugas Akhir-Perpustakaan F. MIPA UNDIP, Semarang, 2005.
- [11] Siang, Jong Jek, "Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab", Andi, Yogyakarta, 2005.
- [12] Sugiharto, Aris, "Pemrograman GUI dengan MATLAB", Andi, Yogyakarta, 2006.
- [13] Zamasari, Reki, "Pengenalan Pola Karakter Aksara Jawa dengan Algoritma Kohonen Neural Network", Tugas Akhir-Perpustakaan F. MIPA UNDIP, Semarang, 2005.
- [14] [www.mathworks.com](http://www.mathworks.com) (diakses terakhir pada tanggal 23 September 2007 jam 15.25 WIB).

