

HUBUNGAN JUMLAH *INPUT LAYER* DAN *OUTPUT LAYER* *NEURAL NETWORK* TERHADAP TINGKAT AKURASI SISTEM *HANDWRITING RECOGNITION* DENGAN METODE *BACKPROPAGATION*

Harjono, Didik Warasto

Politeknik Pratama Mulia Surakarta

bangjont@gmail.com

Abstrak

Karakter pola yang unik dari setiap huruf tulisan tangan diterjemahkan dalam suatu vektor tertentu yang terdiri dari titik awal, titik percabangan dan titik akhir. Neural Network Back-Propagation diaplikasikan pada proses training dan klasifikasi, untuk mencari karakter-karakter yang memiliki pola vektor yang serupa. Input dari sistem berupa file gambar tulisan tangan.

Pengujian yang dilakukan bertujuan untuk mengetahui seberapa besar pengaruh jumlah input layer dan output layer dalam Neural Network terhadap tingkat akurasi hasil recognize dengan jumlah hidden layer konstan, 64. Pengujian proses training dilakukan dengan karakter dari font Arial ukuran 20. Sedangkan pengujian recognition dilakukan dengan mengaplikasikan ke-81 skenario pengujian terhadap suatu gambar tulisan tangan. Gambar tulisan tangan yang digunakan berisikan huruf abjad besar dan kecil dengan penulisan terpisah untuk setiap karakternya.

Dari uji statistik (Pearson Correlation) yang dilakukan diperoleh hasil bahwa input layer dan output layer sangat berpengaruh terhadap tingkat akurasi, yaitu 0.733 untuk input layer dan 0.480 untuk output layer. Sedangkan pengaruh keduanya secara bersamaan terhadap tingkat akurasi adalah 0.876 atau 87.6 %, dan 12.4 % dipengaruhi oleh faktor lain yang tidak diukur dalam penelitian ini.

Kata Kunci: *Neural Network Back-Propagation, input layer, output layer, akurasi recognize, tulisan tangan.*

1. Pendahuluan

Penelitian di bidang *Handwriting Recognition* (pengubahan tulisan tangan ke bentuk teks ketikan) terus dikembangkan, terutama untuk meningkatkan akurasinya. Berbagai metode telah banyak dikembangkan seperti metode ekstrasi fitur, metode *moment*, fitur filter, Gabor, *wavelet*, dan lain-lain dengan memanfaatkan konsep *Artificial Neural Networks* (Choudhary & Rishi, 2011; Gorgel & Oztas, 2007).

Neural network telah banyak digunakan untuk melakukan analisis *recognize* gambar dan dokumen (Plamondon & Srihari, 2000). Akurasi dan jumlah waktu yang dibutuhkan untuk melakukan *recognize* sangat tergantung pada metode yang digunakan dan juga dipengaruhi oleh jumlah setiap *layer neural network* (*input layer, hidden layer dan output layer*) (Pal & Singh, 2010). Tingkat akurasi *Handwriting Recognition* masih bisa ditingkatkan, terutama untuk

karakter-karakter dengan bentuk dasar yang sama, yaitu q-g, C-G-c, e-c, K-k, D-0-Q-o, j-i, I (i besar)-l (L kecil), P-p, S-s, U-u-V-v, W-w, X-x, Y-y, Z-z (Pal & Singh, 2010).

Perbaikan tingkat akurasi sangat dibutuhkan untuk bisa lebih memaksimalkan hasil sistem *Handwriting Recognition* dan tentu saja dengan mempertimbangkan lamanya waktu yang dibutuhkan. Sehingga pada akhirnya nanti, penggunaan sistem komputer akan semakin mudah, terutama untuk pekerjaan tulis-menulis dan *input* data (Pal & Singh, 2010).

Penelitian yang dilakukan Pal dan Singh (2010) menghasilkan kesimpulan bahwa tingkat akurasi berbanding lurus dengan jumlah *layer* neuron simetris pada sistem *Character Recognition Neural Network Fuzzy Logic*. Dan jumlah masing-masing *layer* (*input layer*, *output layer* dan *hidden layer*) dapat ditentukan melalui *trial-error* untuk mendapatkan performa terbaik, dan tergantung pada problem yang akan diselesaikan.

Dalam penelitian ini, sistem yang akan diuji adalah *Handwriting Recognition* dengan metode *backpropagation*. Diharapkan akan terlihat hubungan yang signifikan antara jumlah *input layer* dan *output layer neural network* terhadap tingkat akurasi hasil *recognize*.

Lingkup penelitian dibatasi pada tulisan tangan dimana setiap karakternya terpisah satu dengan yang lain, dengan *font* Arial ukuran 20 sebagai karakter acuan. Karakter-karakter yang ditulis tangan adalah karakter angka dan karakter huruf besar dan kecil dari A sampai Z (Pal & Singh, 2010).

2. Dasar Teori

2.1 *Image Processing*

Image processing adalah suatu metode yang digunakan untuk memproses atau memanipulasi gambar dalam bentuk 2 dimensi (Theodoridis & Koutroumbas, 2003). *Image processing* dapat juga dikatakan segala operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar. Pada umumnya, objektifitas dari *image processing* adalah mentransformasikan atau menganalisis suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas. Secara umum tahapan pengolahan citra digital meliputi akuisisi citra, peningkatan kualitas citra, segmentasi citra, representasi dan uraian, pengenalan dan interpretasi (Gomes & Velho, 1997).

2.2 Segmentasi Citra

Segmentasi bertujuan untuk memilih dan memisahkan suatu objek dari keseluruhan citra. Pada tahap ini dilakukan penapisan dengan filter median untuk menghilangkan derau yang biasanya muncul pada frekuensi tinggi pada spektrum citra. Ada 3 tipe segmentasi, yaitu:

- a. *classification-based* : segmentasi berdasarkan kesamaan suatu ukuran dari nilai *pixel*.
- b. *edge-based* : mencari garis yang ada pada gambar untuk digunakan sebagai pembatas dari tiap segmen.
- c. *region-based* : segmentasi berdasarkan kumpulan *pixel* yang memiliki kesamaan (tekstur, warna atau tingkat warna abu-abu) dimulai dari suatu titik ke titik-titik lain yang ada di sekitarnya.

2.3 Threshold

Threshold adalah suatu tahapan dimana suatu *image* dilakukan pengolahan *pixel* atau menghilangkan beberapa *pixel* dan juga mempertahankan beberapa *pixel* sehingga menghasilkan suatu citra baru hasil sortir *pixel* yang telah dilakukan. *Threshold* dilakukan agar mempermudah dalam proses identifikasi ataupun perbandingan dari dua atau lebih citra. Dalam melakukan *threshold*, dibutuhkan citra dalam bentuk 8 bit dan 2 *channel* atau *grayscale*. Setelah itu citra *grayscale* ini diubah menjadi 2 bit atau *black-and-white*.

Misal pada sebuah gambar, $f(x,y)$ tersusun dari objek yang terang pada sebuah *background* yang gelap. *Gray-level* milik objek dan milik *background* terkumpul menjadi 2 grup yang dominan. Jika ditentukan nilai *threshold* adalah T , maka semua *pixel* yang memiliki nilai $> T$ disebut titik objek, yang lain disebut titik *background*. Proses ini disebut *thresholding*. Sebuah gambar yang telah di-*threshold* $g(x,y)$ dapat didefinisikan sesuai Persamaan 1 (Gomes & Velho, 1997).

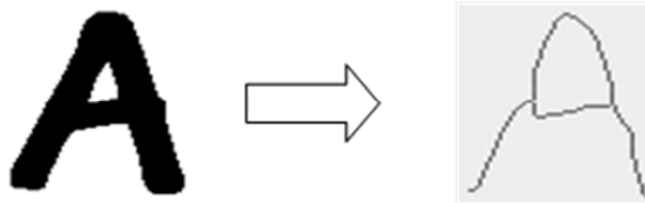
$$g(x,y) = \begin{cases} 1 & \text{jika } f(x,y) > T \\ 0 & \text{jika } f(x,y) \leq T \end{cases} \dots\dots\dots(1)$$

Ketepatan nilai *threshold* sangat mempengaruhi hasil segmentasi. Nilai *threshold* ditentukan dengan mempertimbangkan warna dominan *background* dan warna karakter. Nilai *threshold* yang terlalu rendah dapat menghasilkan segmentasi yang tidak bersih dari *noise*, sedangkan jika terlalu tinggi akan dapat menghilangkan beberapa informasi yang sebenarnya diperlukan (Gomes & Velho, 1997).

2.4 Skeletonizing

Skeletonizing atau *thinning* adalah proses untuk membuang *pixel-pixel* ekstra dan menghasilkan gambar yang lebih sederhana. Tujuan dari *skeletonizing* adalah membuat gambar yang lebih sederhana sehingga gambar tersebut dapat dianalisis lebih lanjut dalam hal bentuk dan kecocokannya maupun untuk dibandingkan dengan gambar lainnya untuk dikenali.

Terdapat beberapa metode yang biasa digunakan, namun dalam penelitian ini yang digunakan adalah metode yang dibuat oleh Zang dan Suen (1984). Ide dasarnya adalah untuk menentukan apakah sebuah *pixel* dapat dierosi hanya dengan melihat 8 tetangga dari *pixel* tersebut (Canuto, 2001). Contohnya dapat dilihat pada Gambar 1.



Gambar 1 Proses *Skeletonizing*

2.5 Feature Extraction

Feature extraction adalah sebuah proses untuk mengubah data *input* menjadi sebuah set *feature* yang dapat disebut dengan *feature vector*. Dalam tahap *feature extraction*, setiap karakter diwakilkan oleh sebuah *feature vector* yang akan menjadi identitasnya. Tujuan utama dari *feature extraction* adalah untuk mengambil beberapa set fitur yang dapat memaksimalkan tingkat pengenalan hanya dengan elemen yang lebih sedikit (Canuto, 2001).

Tulisan tangan memiliki sifat alami yaitu tidak teratur dan dapat berubah-ubah sehingga sangat susah dalam mendapatkan fiturnya. Metode *feature extraction* dibagi menjadi 3, yaitu:

- *Statistical*
Fitur ini mewakili sebuah gambar karakter dengan distribusi *statistical* dari titik-titik yang dapat menangani variasi gaya sampai batas tertentu.
- *Structural*
Karakter dapat diwakilkan dengan fitur *structural* dengan toleransi tinggi atas distorsi dan variasi gaya. Tipe fitur ini dapat menyimpan pengetahuan tentang struktur dari objek atau menyediakan pengetahuan seperti komponen-komponen yang membentuk objek tertentu.

- *Global transformation and moments*

Global transformation menggunakan hasil kalkulasi *Fourier transform* dari kontur gambar.

2.6 Fungsi Aktivasi

Fungsi aktivasi bertindak sebagai fungsi pengontrol, seperti pada keluaran dari sebuah neuron, sehingga keluaran dari neuron dalam jaringan saraf adalah antara nilai-nilai tertentu (biasanya 0 dan 1, atau -1 dan 1). Secara umum, ada tiga tipe fungsi aktivasi, yaitu:

- Fungsi Ambang (*Threshold Function*), seperti terlihat pada Persamaan 2, mengambil nilai 0 jika masukan yang dijumlahkan kurang dari nilai ambang tertentu, dan nilai 1 jika masukan yang dijumlahkan lebih besar atau sama dengan nilai ambang.

$$\varphi(v) = \begin{cases} 1 & \text{jika } v \geq 0 \\ 0 & \text{jika } v < 0 \end{cases} \dots\dots\dots (2)$$

- Fungsi Piecewise-Linear, seperti terlihat pada Persamaan 3, selain membawa nilai 0 atau 1, juga dapat membawa nilai di antaranya, tergantung pada faktor penguatan (*amplification*) pada bagian tertentu dalam operasi linear.

$$\varphi(v) = \begin{cases} 1 & \text{jika } v \geq \frac{1}{2} \\ v & \text{jika } -\frac{1}{2} < v < \frac{1}{2} \\ 0 & \text{jika } v \leq -\frac{1}{2} \end{cases} \dots\dots\dots (3)$$

2.7 Proses Belajar

Belajar merupakan suatu proses dimana parameter-parameter bebas *Neural Network* diadaptasikan melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada. Berdasarkan algoritma pelatihannya, maka *Neural Network* terbagi menjadi dua, yaitu (Canuto, 2001):

- *Supervised Learning*

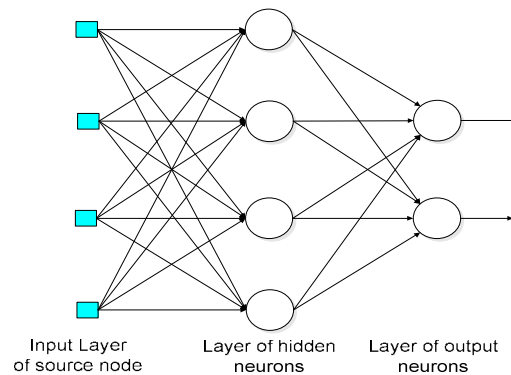
Metode belajar ini memerlukan pengawasan dari luar atau pelabelan data sampel yang digunakan dalam proses belajar. Jaringan belajar dari sekumpulan pola masukan dan keluaran. Vektor masukan dimasukkan ke dalam jaringan dan akan menghasilkan vektor keluaran yang selanjutnya dibandingkan dengan vektor target. Selisih kedua vektor tersebut menghasilkan *error* yang digunakan sebagai dasar untuk mengubah matriks koneksi sehingga *error* semakin mengecil pada siklus berikutnya.

- *Unsupervised Learning*

Metode belajar ini menggunakan data yang tidak diberi label dan tidak memerlukan pengawasan dari luar. Data disajikan kepada *Neural Network* dan membentuk kluster internal yang mereduksi data masukan ke dalam kategori klasifikasi tertentu.

2.8 *Neural Network Back Propagation*

Neural Network Back Propagation merupakan salah satu teknik pembelajaran/pelatihan *supervised learning* yang paling banyak digunakan dalam edukatif. Gambar 2 mengilustrasikan bagaimana arsitektur dari *Neural Network Back Propagation*.



Gambar 2 *Neural Network Back Propagation*

Algoritma pelatihan *Neural Network Back Propagation* adalah sebagai berikut (Theodoridis & Koutroumbas, 2003):

1. Definisi masalah, misalkan matriks masukan (P) dan matriks target (T).
2. Inisialisasi, menentukan bentuk jaringan dan menetapkan nilai-nilai bobot sinaptik ($W1$ dan $W2$) dan *learning rate* (lr).
3. Pelatihan Jaringan
 - a. Perhitungan Maju
 - Masing-masing unit masukan mendapatkan sinyal masukan berupa x_j dan kemudian sinyal-sinyal tersebut diteruskan ke unit-unit tersembunyi.
 - Masing-masing sinyal yang diterima dari unit masukan kemudian dikalikan dengan bobot.
 - Kemudian dihitung menggunakan fungsi aktivasi yang digunakan, misal: sigmoid tangen hiperbola.
 - Melakukan kembali langkah nomor 3.a. untuk masing-masing unit pada unit keluaran.

b. Penyebaran balik (*backpropagation*)

- Masing-masing unit keluaran menerima pola target sesuai dengan pola keluaran yang diinginkan dan kemudian dihitung *error*-nya menggunakan Persamaan 4.

$$\partial_k = y_k - t_k \dots\dots\dots(4)$$

dimana $k = 0$ sampai n , sebagai indeks neuron *output*

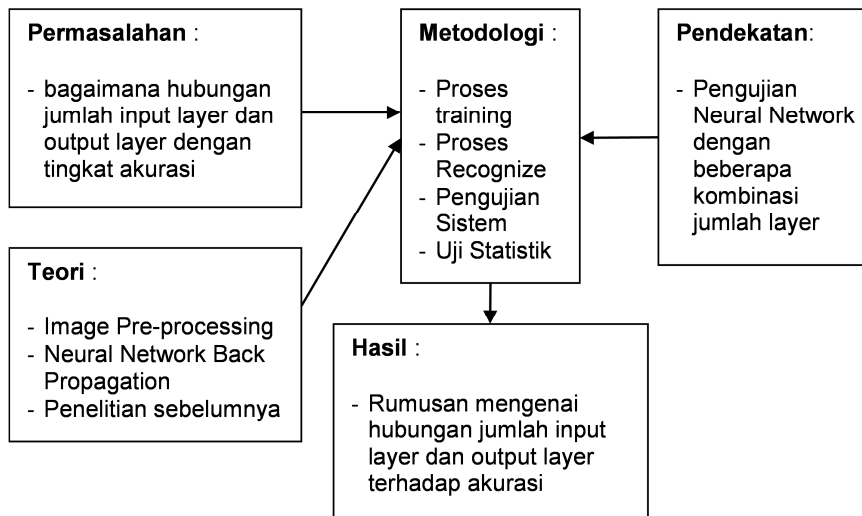
- Sebar balik ke unit-unit terdahulunya menggunakan Persamaan 5.

$$\partial_j = \varphi'(v_j)w_{kj}\partial_k \dots\dots\dots(5)$$

3. Pembahasan

3.1 Metodologi

Skema kerangka pikir dalam penelitian ini dapat dilihat pada Gambar 3.



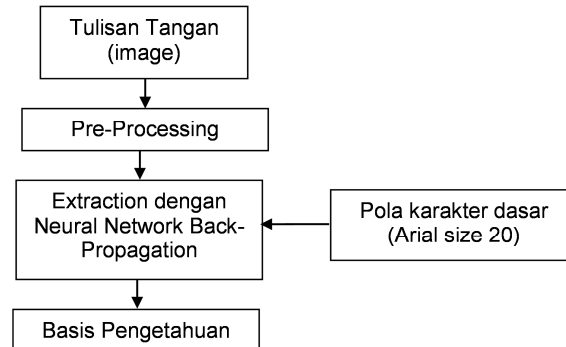
Gambar 3 Skema kerangka pikir

Sistem perangkat lunak yang dikembangkan terdiri dari dua modul utama. Kedua modul tersebut adalah modul *training* dan modul *recognize*.

3.2 Modul Training

Modul *training* dimaksudkan untuk melatih sistem mengenali karakter-karakter huruf dan angka dengan menggunakan *Neural Network Back-Propagation*. *Input* pada modul ini pola dasar yang berupa karakter tulisan tangan sebagai bahan pelatihan. Dari *input* yang berupa gambar ini selanjutnya dibuat matriks yang mencerminkan pola-pola dari setiap huruf yang akan dikenali serta merupakan inputan bagi tahap berikutnya yakni *Neural Network Back-Propagation*. Pada tahap akhir ini sebuah jaringan syaraf tiruan akan dibangun guna pengambilan keputusan akhir terhadap *input*-an. Hasil dari modul *training*

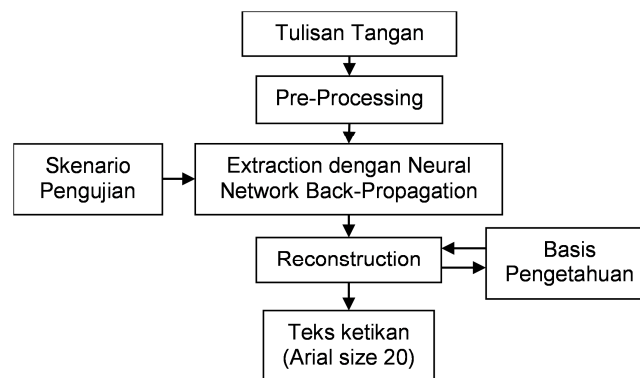
ini adalah sebuah basis pengetahuan mengenai pengenalan pola-pola setiap karakter dari tulisan tangan. Secara garis besar diagram alur modul *training* dapat dilihat pada Gambar 4.



Gambar 4 Diagram blok modul *training*

3.3 Modul Recognize

Modul *recognize* dimaksudkan untuk melakukan pengubahan dari tulisan tangan ke bentuk teks ketikan dengan jenis *font* Arial ukuran 20. Untuk mengenali karakter-karakter tulisan tangan yang berupa *file* gambar dengan format BMP. Pada modul ini juga dilakukan proses klasifikasi dan penulisan ulang karakter-karakter yang berhasil dikenali dan diklasifikasikan dengan *font* Arial ukuran 20. Pengubahan tersebut didasarkan pada Basis Pengetahuan yang merupakan hasil dari modul *training*. *Input* yang digunakan pada modul kedua adalah *file* gambar tulisan. Proses yang dilakukan pada modul kedua adalah seperti terlihat pada Gambar 5.



Gambar 5 Diagram alur modul *recognize*

3.4 Skenario Pengujian

Skenario pengujian digunakan supaya arah dan tujuan penelitian tidak bergeser dari rumusan masalah yang sudah ditetapkan. Berdasarkan penelitian yang dilakukan Pal dan Singh (2010), bahwa tingkat akurasi berbanding lurus dengan jumlah *layer* neuron simetris pada sistem *Character Recognition - Neural*

Network Back Propagation maka disusun sebuah skenario pengujian untuk mendapatkan hubungan yang signifikan antara jumlah *layer input* dan jumlah *layer output* dengan tingkat akurasi. Tabel 1 memperlihatkan skenario pengujian.

Tabel 1 Skenario Pengujian

Skenario	<i>input layer</i>	<i>output layer</i>	Skenario	<i>input layer</i>	<i>output layer</i>
1	100	100	46	225	100
2	100	121	47	225	121
3	100	144	48	225	144
4	100	169	49	225	169
5	100	196	50	225	196
6	100	225	51	225	225
7	100	256	52	225	256
8	100	289	53	225	289
9	100	324	54	225	324
10	121	100	55	256	100
11	121	121	56	256	121
12	121	144	57	256	144
13	121	169	58	256	169
14	121	196	59	256	196
15	121	225	60	256	225
16	121	256	61	256	256
17	121	289	62	256	289
18	121	324	63	256	324
19	144	100	64	289	100
20	144	121	65	289	121
21	144	144	66	289	144
22	144	169	67	289	169
23	144	196	68	289	196
24	144	225	69	289	225
25	144	256	70	289	256
26	144	289	71	289	289
27	144	324	72	289	324
28	169	100	73	324	100
29	169	121	74	324	121
30	169	144	75	324	144
31	169	169	76	324	169
32	169	196	77	324	196
33	169	225	78	324	225
34	169	256	79	324	256
35	169	289	80	324	289
36	169	324	81	324	324
37	196	100			
38	196	121			
39	196	144			
40	196	169			
41	196	196			
42	196	225			
43	196	256			
44	196	289			
45	196	324			

4. Pengujian

4.1 Proses Training

Karakter standar yang digunakan dalam proses *training* adalah huruf abjad besar dan kecil dari *font* Arial ukuran 20. Basis pengetahuan yang sudah

dibuat berjumlah 81 buah, yang merupakan kombinasi dari jumlah *input layer* dan jumlah *output layer* dengan jumlah *hidden layer* konstan yaitu 64. Basis pengetahuan ini selanjutnya akan digunakan sebagai dasar dalam proses *recognize* terhadap karakter-karakter pola tulisan tangan.

Parameter-parameter yang digunakan pada proses *training* adalah sebagai berikut:

1. Learning rate : 0,001
2. Derajat Inialisasi : 0,5
3. Error Threshold : 1

Sedangkan kondisi yang menyebabkan proses *training* selesai (*Stop Condition*) adalah:

1. Target Classification Error : -1
2. Target Squared Error : 0,01
3. Maximum Epochs : 10000

4.2 Proses *Recognize*

Dari data informasi pengujian proses *recognition* yang dilakukan terlihat bahwa jumlah *input layer* dan *output layer* sangat mempengaruhi tingkat akurasi *recognition*. Semakin besar jumlah masing-masing *layer*, ada kecenderungan tingkat akurasi juga semakin meningkat.

Dari uji Regresi Berganda diketahui bahwa nilai R adalah 0.876. Hal ini menunjukkan bahwa secara bersamaan, variable jumlah *input layer* dan *output layer* mempunyai pengaruh terhadap tingkat akurasi cukup kuat, yaitu sebesar 0,876 atau 87,6%. Sedangkan 12,4% dipengaruhi oleh faktor lain yang tidak diukur dalam penelitian ini.

Dari uji statistik (*Pearson Correlation*) yang dilakukan diperoleh hasil bahwa *input layer* dan *output layer* sangat berpengaruh terhadap tingkat akurasi, yaitu 0,733 untuk *input layer* dan 0,480 untuk *output layer*. Sedangkan dari uji relasi berganda, diperoleh hasil bahwa hubungan tingkat akurasi dengan jumlah *input layer* dan *output layer* adalah seperti pada Persamaan 6.

$$Y = 4.826 + 0.59 IL + 0.39 OL \dots\dots\dots(6)$$

Keterangan:

Y : Tingkat Akurasi

IL : *Input Layer*

OL : *Output Layer*

Hasil tersebut adalah pengujian dengan jumlah *Hidden Layer* konstan, yaitu 64, dan jumlah *Input Layer (IL)* minimal 100 dan maksimal 324. Sedangkan jumlah *Output Layer (OL)* minimal 100 dan maksimal 324.

5. Penutup

Kesimpulan yang bisa diambil dari penelitian adalah bahwa jumlah *input layer* dan jumlah *output layer* sangat menentukan tingkat akurasi sistem *handwriting recognition*. Semakin besar jumlah *input layer* maka tingkat akurasi juga akan meningkat. Begitu juga untuk jumlah *output layer*. Secara bersamaan, variable jumlah *input layer* dan *output layer* mempunyai pengaruh terhadap tingkat akurasi cukup kuat, yaitu sebesar 0,876 atau 87,6%.

Pengujian dengan jumlah *Hidden Layer* konstan (64), jumlah *Input Layer (IL)* dan *Output Layer (OL)* antara 100 dan 324 menunjukkan hasil bahwa:
$$Y = 4.826 + 0.59 IL + 0.39 OL .$$

Untuk mendapatkan hasil yang lebih akurat, sangat perlu dilakukan penelitian dengan jumlah *input layer* dan *output layer* mulai dari 1 sampai jumlah yang lebih dari 324, atau lebih besar lagi. Selain itu, juga perlu dilakukan penelitian dengan menggunakan jumlah *hidden layer* yang tidak konstan.

Daftar Pustaka

- Canuto, A. M. P., 2001. Combining Neural Networks and Fuzzy Logic for Applications in Character Recognition. *Doctoral dissertation*. Canterbury, UK: University of Kent.
- Choudhary, A. & Rishi, R., 2011. Improving The Character Recognition Efficiency of Feed Forward BP Neural Network. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(1), pp. 85-96.
- Gomes, J. & Velho, L., 1997. *Image Processing For Computer Graphics*. New York, USA: Springer Science+Business Media.
- Gonzalez, R. C. & Woods, R. E., 2008. *Digital Image Processing*, 3rd Edition. New Jersey, USA: Prentice Hall International Inc.
- Gorgel, P., & Oztas, O., 2007. Handwritten Character Recognition System Using Artificial Neural Networks. *Istanbul University Journal of Electrical & Electronic Engineering*, 7(1), pp. 309-313.
- Kala, R., Vazirani, H., Shukla, A. & Tiwari, R., 2010. Offline Handwriting Recognition Using Genetic Algorithm. *International Journal of Computer Science Issues (IJCSI)*, 7(2), pp. 16-25.

- Pal, A. & Singh, D., 2010. Handwritten English Character Recognition Using Neural Network. *International Journal of Computer Science & Communication*, 1(2), pp. 141-144.
- Plamondon, R., & Srihari, S. N., 2000. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), pp. 63-84.
- Pradeep, J., Srinivasan, E. & Himavathi, S., 2011. Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(1), pp. 27-38.
- Theodoridis, S. & Koutroumbas, K., 2003. *Pattern Recognition*, 2nd Edition. San Diego, USA: Academic Press.
- Zhang, T. Y. & Suen, C. Y., 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. *Communications of the ACM*, 27(3), pp. 236-239.