

# **Penerapan Jaringan Syaraf Tiruan metode Propagasi Balik Dalam Pengenalan Tulisan Tangan Huruf Jepang Jenis Hiragana dan Katakana**

**Erico Darmawan Handoyo<sup>1)</sup>, Lydia Wiguna Susanto<sup>2)</sup>**

Jurusan S1 Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri no. 65 Bandung 40164

email: <sup>1)</sup>khe\_wan\_xing@yahoo.com ; <sup>2)</sup>tsukiakiko@yahoo.com

## *Abstract*

*Artificial intelligence is a branch in computer science which tries to make machines act like human. Artificial neural network imitates human biological learning process, which also creates some foundations in pattern recognition. An application area of pattern recognition is in learning foreign languages that use computer to help human to understand the roles of new foreign characters, especially graphical characters, such as in Japanese. Our goal in this research is to build a system which would be able to recognize handwriting in Japanese Hiragana and Katakana characters, by using artificial neural network with back propagation learning mechanism.*

*Keywords: artificial neural network, back propagation, pattern recognizing, hiragana, katakana.*

## **1. Pendahuluan**

Banyaknya bangsa dan negara di dunia, menimbulkan keanekaragaman bahasa dan budaya. Hal tersebut membuat manusia mengalami kesulitan untuk saling mengerti maksud dan tujuan dari lawan bicaranya. Agar dapat memudahkan komunikasi antar bangsa dan antar budaya, dibentuklah bahasa internasional. Selain dengan adanya bahasa internasional, manusia juga berusaha untuk mempelajari bahasa dan budaya dari bangsa lain.

Berdasarkan data daftar bahasa yang digunakan menurut jumlah penutur asli, bahasa Jepang berada pada urutan ke-9 setelah bahasa Mandarin, Hindi, Spanyol, Inggris, Bengali, Arab, Rusia dan Portugis [Mof07]. Berdasarkan data 10 bahasa yang paling sering digunakan di internet, bahasa Jepang menduduki peringkat ke-4 setelah bahasa Inggris, Mandarin dan Spanyol [Int07]. Kedua data ini menggambarkan bahwa bahasa Jepang adalah penting dan digunakan secara internasional.

Terdapat berbagai cara bagi manusia untuk mempelajari bahasa-bahasa asing. Selain dengan mempelajarinya secara manual, seperti kursus dan berbicara aktif, juga dapat mempelajari bahasa asing secara digital, autodidak, melalui komputer dan internet. Mesin (komputer) perlu mengerti dan menguasai karakter serta huruf dalam bahasa asing tersebut, agar dapat membantu manusia dalam mempelajari

bahasa asing (Contoh: telepon seluler touchscreen sudah memiliki handwriting recognition, baik alphabet maupun huruf kanji mandarin. Contoh lain: Tablet PC).

Pemanfaatan komputer sebagai alat bantu manusia sangat diharapkan sampai tingkat kemampuan komputer tersebut dapat menggantikan keterbatasan yang dimiliki oleh manusia. Manusia dapat mengenali sebuah objek dengan menggunakan mata dan otaknya, tetapi bila mata dan otak tidak dapat bekerja dengan baik maka akan membuat pekerjaan manusia menjadi terhambat.

Kecerdasan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Secara umum, jaringan syaraf tiruan memiliki jaringan syaraf seperti manusia dan dapat melakukan proses pembelajaran seperti yang dilakukan oleh manusia. Aspek yang cukup penting yang mendasari berbagai teori dalam kecerdasan buatan adalah sistem pengenalan pola (*pattern recognizing*) yang merupakan bagian dari pengimplementasian jaringan syaraf tiruan secara praktis. Sistem pengenalan pola merupakan komponen penting dalam proses peniruan kemampuan inderawi manusia terutama penglihatan dan pendengaran.

Teknik pengenalan pola (*pattern recognition*) mengalami banyak kemajuan dan semakin dipakai untuk memecahkan suatu masalah. Teknik pengenalan pola dipakai untuk mengenali tulisan tangan, tanda tangan, gambar dan sebagainya. Berbeda dengan disiplin ilmu pengolahan citra yang dibatasi oleh penggunaan citra sebagai masukan maupun keluarannya, suatu aplikasi pengenalan pola bertujuan untuk melakukan proses pengenalan terhadap suatu objek (misalnya citra) ke dalam salah satu kelas tertentu, berdasarkan pola yang dimilikinya. Jaringan syaraf tiruan memiliki beberapa metode atau algoritma yang dapat digunakan untuk mengklasifikasikan atau mengenali suatu tipe pola khususnya dalam pengenalan pola tulisan tangan, salah satunya adalah propagasi balik.

Algoritma ini melakukan dua tahap perhitungan, yaitu: perhitungan maju untuk menghitung galat (*error*) antara keluaran aktual dan target; dan perhitungan mundur yang mempropagasikan balik galat (*error*) tersebut untuk memperbaiki bobot-bobot sinaptik pada semua neuron yang ada. Dengan kata lain, kita dapat memasukkan acuan yg akan digunakan sesuai kebutuhan.

## **2. Tujuan**

Tujuan utama penulisan ini ialah untuk membuat perangkat lunak yang dapat mengenali tulisan tangan huruf Jepang jenis Hiragana dan Katakana. Pembuatan perangkat lunak ini akan menggunakan jaringan syaraf tiruan dengan metode propagasi balik. Perangkat lunak ini akan menerima input berupa gambar melalui *mouse/digitizer*, kemudian akan menggunakan jaringan syaraf tiruan dengan metode propagasi balik untuk mengenali huruf Katakana dan Hiragana tersebut. Selain itu juga dapat membuka gambar tertentu, kemudian akan menggunakan

jaringan syaraf tiruan dengan metode propagasi balik untuk mengenali huruf Katakana dan Hiragana tersebut.

### 3. Skenario Sistem

Aplikasi yang akan dibuat mempunyai (skenario penggunaan) dengan batasan – batasan sebagai berikut :

1. Menerapkan jaringan syaraf tiruan metode propagasi balik.
2. Tidak menggabungkan jenis huruf Hiragana dan Katakana dalam proses pengenalan huruf.
3. Menerima *input* gambar dari *mouse/digitizer*.
4. Menerima *input* gambar dari file gambar yang dipilih pengguna.
5. Hanya mengenali satu buah huruf dalam satu kali proses pengenalan huruf.
6. Tidak menyediakan fasilitas untuk penambahan sample huruf baru.
7. Menampilkan *output* berupa gambar, pelafalan suku kata dan persentase kemiripannya.

### 4. Tinjauan Pustaka

#### 4.1 Hiragana dan Katakana

Bahasa Jepang mempunyai 5 huruf vokal yaitu /a/, /i/, /u/, /e/, dan /o/. Tulisan bahasa Jepang berasal dari tulisan bahasa China (kanji) yang diperkenalkan pada abad keempat Masehi[3]. Sebelum ini, masyarakat Jepang tidak mempunyai sistem penulisan sendiri. Bahasa Jepang ditulis dengan menggunakan kombinasi dari tiga huruf yang berbeda, yaitu : huruf Kanji yang berasal dari China, huruf Hiragana, dan huruf Katakana. Kedua aksara terakhir ini disebut “Kana” dan keduanya terpengaruhi fonetik bahasa Sansekerta. Hal ini masih bisa dilihat dalam urutan aksara Kana. Selain itu, ada pula sistem alih aksara yang disebut romaji.

**Tabel1. Huruf Hiragana**

		ひらがな Hiragana								
Seion	あ	a	い	i	う	u	え	e	お	o
	か	ka	き	ki	く	ku	け	ke	こ	ko
	さ	sa	し	shi	す	su	せ	se	そ	so
	た	ta	ち	chi	つ	tsu	て	te	と	to
	な	na	に	ni	ぬ	nu	ね	ne	の	no
	は	ha	ひ	hi	ふ	fu	へ	he	ほ	ho
	ま	ma	み	mi	む	mu	め	me	も	mo
	や	ya			ゆ	yu			よ	yo
	ら	ra	り	ri	る	ru	れ	re	ろ	ro
	わ	wa							を	wo
ん	n									

*Sumber : [http://www.saiga-jp.com/japanese\\_language.html](http://www.saiga-jp.com/japanese_language.html)*

Tabel2. Huruf Katakana

かたかな Katakana										
Seion	ア	a	イ	i	ウ	u	エ	e	オ	o
	カ	ka	キ	ki	ク	ku	ケ	ke	コ	ko
	サ	sa	シ	shi	ス	su	セ	se	ソ	so
	タ	ta	チ	chi	ツ	tsu	テ	te	ト	to
	ナ	na	ニ	ni	ヌ	nu	ネ	ne	ノ	no
	ハ	ha	ヒ	hi	フ	fu	ヘ	he	ホ	ho
	マ	ma	ミ	mi	ム	mu	メ	me	モ	mo
	ヤ	ya			ユ	yu			ヨ	yo
	ラ	ra	リ	ri	ル	ru	レ	re	ロ	ro
	ワ	wa							ヲ	wo
	ン	n								

Sumber : [http://www.saiga-jp.com/japanese\\_language.html](http://www.saiga-jp.com/japanese_language.html)

#### 4.2 Pengolahan Citra Digital dan Digitalisasi Citra Digital

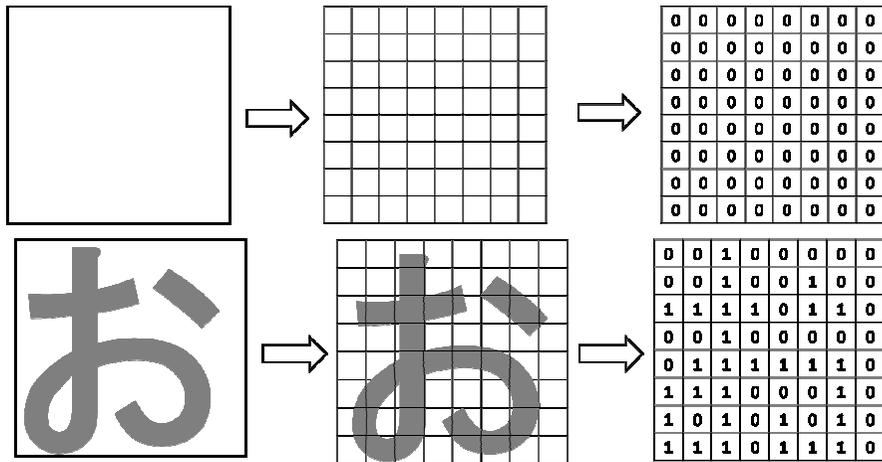
Pengenalan pola (*pattern recognition/image representation*) merupakan salah satu bidang studi yang berkaitan dengan citra di bidang komputer. Dalam proses pengenalan pola, mesin (komputer) akan mengelompokkan data numerik dan simbolik, tujuannya ialah untuk mengenali suatu objek dalam citra. Mesin (komputer) mencoba meniru sistem visual manusia untuk bisa mengenali objek. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra.

Komputer memiliki cara pandang tersendiri terhadap suatu citra, biasa disebut sebagai computer vision. Berbeda dengan citra konvensional yang misalnya dengan melalui proses fotografis seperti pada foto dapat dihasilkan suatu citra nyata yang langsung dapat dinikmati oleh indera penglihatan, citra pada komputer harus melalui beberapa tahapan yang cukup rumit. Tahapan-tahapan tersebut dapat digambarkan sebagai suatu rangkaian proses dari proses akuisisi data, manipulasi data, visualisasi data, serta proses penyimpanan data.

Suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit agar dapat diolah dengan komputer. Representasi citra dari kontinu menjadi nilai-nilai diskrit disebut digitalisasi. Citra yang dihasilkan dari proses representasi tersebut akan berupa citra digital. Citra digital akan berbentuk persegi panjang dan dimensi ukurannya dinyatakan sebagai tinggi x lebar. Citra digital yang berukuran N x M biasanya dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut :

$$\begin{bmatrix} (0,0) & \dots & (0,M-1) \\ \vdots & \vdots & \vdots \\ (N-1,0) & \dots & (N-1,M-1) \end{bmatrix} \quad \text{(Persamaan 1)}$$

Proses digitalisasi citra ada dua macam, yaitu digitalisasi spasial (penerokan/sampling) dan digitalisasi intensitas (kuantisasi). Dalam proses digitalisasi spasial, sebuah citra kontinu diterok pada grid yang berbentuk bujursangkar.

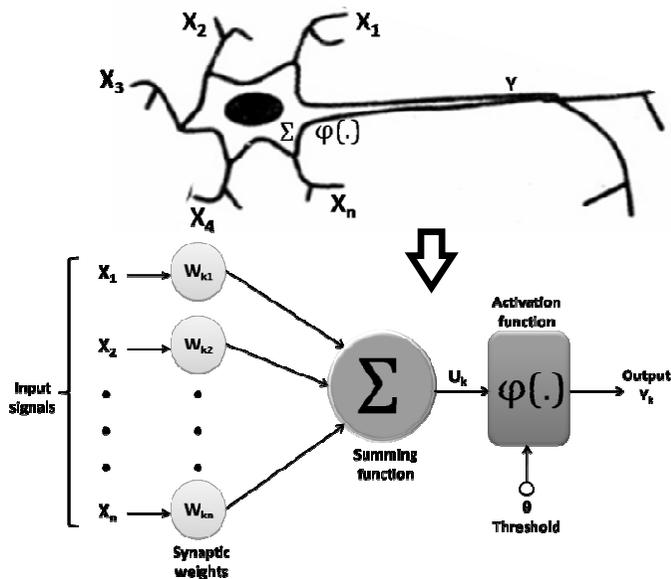


Gambar 1. Digitalisasi spasial (sampling)

### 4.3 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (*Artificial Neural Network (ANN)*), atau juga disebut *Simulated Neural Network (SNN)*, atau umumnya hanya disebut *Neural Network (NN)*, adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. Jaringan syaraf tiruan merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut neuron, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi.

Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan terdapat hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima, melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar di bawah ini menunjukkan struktur neuron pada jaringan syaraf.



Gambar 2. Struktur neuron jaringan syaraf

*Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya, demikian seterusnya.

Pada Gambar 2.3 terlihat serangkaian aliran sinyal masukan  $X_1, X_2, \dots, X_n$  yang direpresentasikan oleh sebuah neuron. Sebuah neuron bisa memiliki banyak masukan dan hanya satu keluaran yang bisa menjadi masukan bagi neuron-neuron yang lain. Aliran sinyal masukan ini dikalikan dengan suatu penimbang (bobot sinapsis)  $W_{k1}, W_{k2}, \dots, W_{kn}$  dan kemudian dilakukan penjumlahan terhadap semua masukan yang telah diboboti tadi. Hasil penjumlahan ini disebut keluaran dari *the linear combiner*  $U_k$ .

Secara matematis neuron k dapat digambarkan melalui persamaan berikut:

$$U_k = \sum_{j=1}^n W_{kj} X_j \text{ dan } Y_k = \varphi(U_k - \theta_k) \quad (\text{Persamaan 2})$$

dimana :

- $X_1, X_2, \dots, X_n$  : sinyal input
- $W_{k1}, W_{k2}, \dots, W_{kn}$  : bobot sinaptik dari neuron k
- $U_k$  : linear combiner output
- $\theta_k$  : threshold (diterapkan secara eksternal)

$\phi(\cdot)$  : fungsi aktivasi  
 $Y_k$  : sinyal output

Fungsi aktivasi yang dinotasikan dengan  $\phi(\cdot)$  mendefinisikan nilai *output* dari suatu neuron dalam level aktivasi tertentu berdasarkan nilai *output* pengkombinasi linier  $U_k$ . Fungsi aktivasi yang digunakan yaitu fungsi sigmoid logistik:

$$\phi(v) = \frac{1}{1+e^{-v}} \quad \text{(Persamaan 3)}$$

dimana :

$\phi(v)$  : fungsi aktivasi  
 $e$  : konstanta bernilai 2,718281828...  
 $v$  : sinyal output

#### 4.4 Algoritma Propagasi Balik

1. Definisikan masalah, misalkan matriks masukan (P) dan matriks target (T).
2. Inisialisasi, menentukan arsitektur jaringan, nilai ambang MSE (*Mean Square Error*) sebagai kondisi berhenti, *learning rate*, serta menetapkan nilai-nilai bobot sinaptik melalui pembangkitan nilai acak dengan interval nilai sembarang. Kita bisa membangkitkan nilai acak dalam interval [-1,+1] atau [-0.5,+0.5] ataupun lainnya. Tidak ada aturan yang baku mengenai interval ini.
3. Pelatihan jaringan :

- a. Perhitungan Maju

Dengan menggunakan bobot-bobot yang telah ditentukan pada inisialisasi awal (W1), dapat menghitung keluaran dari lapisan dalam berdasarkan persamaan berikut :

$$A1 = \frac{1}{1+e^{-(W1 \cdot P + B1)}} \quad \text{(Persamaan 4)}$$

Hasil keluaran lapisan dalam (A1) digunakan untuk mendapatkan keluaran dari lapisan keluaran, dengan persamaan berikut :

$$A2 = W2 * A1 + B2 \quad \text{(Persamaan 5)}$$

Keluaran dari jaringan (A2) dibandingkan dengan target yang diinginkan. Selisih nilai tersebut adalah galat (*error*) dari jaringan, seperti pada persamaan berikut :

$$SSE = \sum \sum E^2 \quad \text{(Persamaan 6)}$$

- b. Perhitungan Mundur

Nilai galat (*error*) yang didapat digunakan sebagai parameter dalam pelatihan. Pelatihan akan selesai jika galat (*error*) yang diperoleh sudah dapat diterima. Galat (*error*) yang didapat dikembalikan lagi ke lapis-lapis yang berada di depannya. Selanjutnya, neuron pada lapis tersebut akan memperbaiki nilai-

nilai bobotnya. Perhitungan perbaikan bobot diberikan pada persamaan berikut :

$$\begin{aligned}
 D2 &= (1 - A2^2) * E \\
 D1 &= (1 - A1^2) * (W2 * D2) \\
 dW1 &= dW1 + (lr * D1 * P) \\
 dB1 &= dB1 + (lr * D1) \\
 dW2 &= dW2 + (lr * D2 * P) \\
 dB2 &= dB2 + (lr * D2)
 \end{aligned}
 \tag{Persamaan 7}$$

c. Perbaikan Bobot Jaringan

Setelah neuron-neuron mendapatkan nilai-nilai yang sesuai dengan kontribusinya pada galat (*error*) keluaran, maka bobot-bobot jaringan akan diperbaiki agar galat (*error*) dapat diperkecil. Perbaikan bobot jaringan dilakukan dengan persamaan berikut :

$$\begin{aligned}
 TW1 &= W1 + dW1 \\
 TB1 &= B1 + dB1 \\
 TW2 &= W2 + dW2 \\
 TB2 &= B2 + dB2
 \end{aligned}
 \tag{Persamaan 8}$$

d. Presentasi Bobot Jaringan

Bobot-bobot yang baru, hasil perbaikan, digunakan kembali untuk mengetahui apakah bobot-bobot tersebut sudah cukup baik bagi jaringan. Baik bagi jaringan berarti bahwa dengan bobot-bobot tersebut, galat yang akan dihasilkan sudah cukup kecil. Pemakaian nilai bobot-bobot yang baru diperlihatkan pada persamaan-persamaan berikut :

$$\begin{aligned}
 TA1 &= \frac{1}{1 + e^{-(TW1 * P + TB1)}} \\
 TA2 &= TW2 * TA1 + TB2 \\
 TE &= T - TA2 \\
 TSSE &= \sum \sum TE^2
 \end{aligned}
 \tag{Persamaan 9}$$

Kemudian bobot-bobot sinapsis jaringan diubah menjadi bobot-bobot baru :

$$\begin{array}{llll}
 W1 = TW1 & B1 = TB1 & W2 = TW2 & B2 = TB2 \\
 A1 = TA1 & A2 = TA2 & E2 = TE2 & SSE = TSSE
 \end{array}$$

Keterangan :

- Wn = Nilai bobot
- TWn = Nilai bobot baru
- An = Nilai *input*
- TAn = Nilai *input* baru
- Bn = Nilai *output*
- TBn = Nilai *output* baru
- En = Nilai galat
- TEn = Nilai galat baru
- SSE = *Sum Square Error*

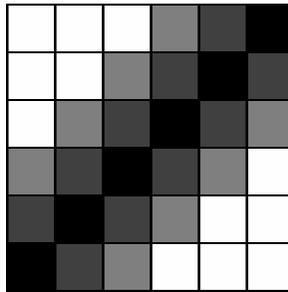
TSSE = *Sum Square Error* baru

- Langkah-langkah diatas adalah untuk satu kali siklus pelatihan (satu epoch). Biasanya, pelatihan harus diulang-ulang lagi hingga jumlah siklus tertentu atau telah tercapai SSE (*SumSquare Error*) atau MSE (*Mean Square Error*) yang diinginkan.  
Hasil akhirnya merupakan bobot-bobot W1, W2, B1 dan B2.

#### **4.5 Penerapan Metode Propagasi Balik pada Pengenalan Huruf**

Berikut ini akan dipaparkan proses metode propagasi balik dalam mengenali huruf. Ukuran yang digunakan sebagai contoh saat ini adalah 6 x 6 pixel. Ukuran ini diambil karena jika kurang dari 6x6 pixel, data yg diperlukan untuk perhitungan kurang mencukupi. Sebaliknya, jika lebih dari 6x6 pixel, sebenarnya akan lebih baik, tapi akan menyebabkan perhitungannya yang sangat panjang.

- Tentukan citra yang akan digunakan sebagai referensi /acuan pembelajaran (Gambar 2.6).



Gambar 3. Citra referensi (acuan pembelajaran)

- Ubah citra menjadi rangkaian data numerik, sehingga dapat diolah dalam proses pengenalan huruf. Gambar A mewakili matriks warna dengan nilai RGB yang telah diolah menjadi grayLevelnya. Gambar B mewakili matriks yang merubah gambar A dengan nilai '0' untuk pixel yang grayLevelnya lebih dari 127, dan nilai '1' untuk pixel yang grayLevelnya kurang dari atau sama dengan 127.

255	255	255	150	60	0
255	255	150	60	0	60
255	150	60	0	60	150
150	60	0	60	150	255
60	0	60	150	255	255
0	60	150	255	255	255

Gambar 4. Gambar A

0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	1	1	0
0	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	0

Gambar 5. Gambar B

Kemudian ubah matriks B menjadi sebuah array dengan nilai yang bersesuaian secara horisontal:

0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1 1 0 0 0

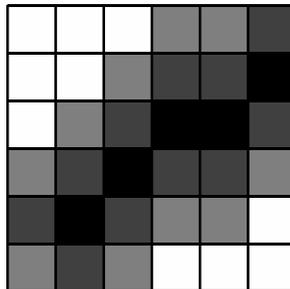
3. Inisialisasi arsitektur jaringan, *learning rate* yang digunakan saat ini adalah 0,5, serta menetapkan nilai-nilai bobot sinaptik melalui pembangkitan nilai acak dengan interval nilai sembarang. Interval yang digunakan saat ini adalah [-1,+1].
4. Lakukan perhitungan maju : Kalikan setiap nilai *input* dengan bobotnya, kemudian dijumlahkan. Hitung nilai aktivasinya.

Urutan array ke-i	X = Nilai Array/Input	W = Nilai Bobot	$X_i * W_i$
1	0	-1	0
2	0	0	0
3	0	0	0
4	0	0	0
5	1	1	1
6	1	1	1
7	0	0	0
8	0	0	0
9	0	1	0
10	1	1	1
11	1	0	0
12	1	0	0
13	0	0	0
14	0	-1	0
15	1	1	1
16	1	1	1
17	1	-1	-1
18	0	-1	0
19	0	1	0
20	1	-1	-1
21	1	0	0
22	1	0	0
23	0	1	0
24	0	0	0
25	1	0	0

Urutan array ke-i	X = Nilai Array/Input	W = Nilai Bobot	X <sub>i</sub> * W <sub>i</sub>
26	1	1	1
27	1	1	1
28	0	-1	0
29	0	-1	0
30	0	1	0
31	1	0	0
32	1	0	0
33	0	-1	0
34	0	1	0
35	0	-1	0
36	0	1	0
Total	5		

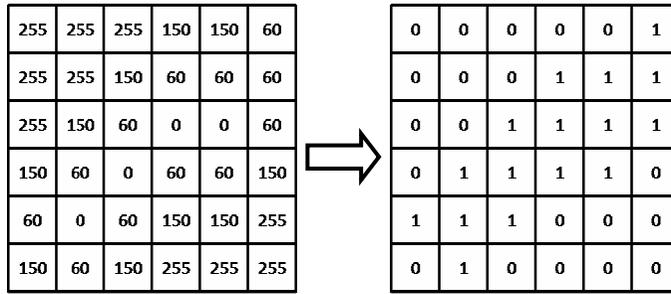
Aktivasi :  $\frac{1}{1+e^{-5}}$  = 0,997527377. Maka nilai *outputnya* ialah 0,997527377. Perhitungan terus dilakukan sebanyak jumlah citra referensinya.

5. Lakukan perhitungan mundur : Nilai galat yang didapat digunakan sebagai parameter dalam pelatihan. Misalkan galat yang didapat pada contoh ini adalah 0,74653251. Pelatihan akan selesai jika galat yang diperoleh sudah dapat diterima. Galat yang didapat dikembalikan lagi ke lapis-lapis yang berada di depannya. Selanjutnya, neuron pada lapis tersebut akan memperbaiki nilai-nilai bobotnya. Bobot baru = 0,5 x 0,74653251 x 1 = 0,373266255.
6. Langkah-langkah diatas adalah untuk satu kali proses pelatihan. Langkah-langkah tersebut terus diulang hingga tercapai batas *error* tertentu. Batas *error* yang digunakan saat ini adalah 0,1. Hasil akhirnya merupakan bobot-bobot baru.
7. Setelah melakukan proses pelatihan terhadap seluruh citra referensi (acuan), maka dapat melakukan proses pengenalan citra / pengenalan huruf yang diinput pengguna.
8. Tentukan citra yang akan digunakan sebagai bahan pengujian.



Gambar 6. Citra pengujian

9. Lakukan langkah nomor 2 hingga nomor 4 terhadap citra tersebut.



0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 0 0 0 0

Urutan array ke-i	X = Nilai Array/Input	W = Nilai Bobot	$X_i * W_i$
1	0	1	0
2	0	-1	0
3	0	-1	0
4	0	0	0
5	0	-1	0
6	1	1	1
7	0	-1	0
8	0	1	0
9	0	0	0
10	1	1	1
11	1	-1	-1
12	1	0	0
13	0	1	0
14	0	-1	0
15	1	1	1
16	1	0	0
17	1	1	1
18	1	0	0
19	0	-1	0
20	1	0	0
21	1	-1	-1
22	1	0	0
23	1	1	1
24	0	-1	0
25	1	1	1
26	1	0	0
27	1	1	1
28	0	1	0
29	0	1	0
30	0	1	0
31	0	1	0
32	1	-1	-1
33	0	0	0
34	0	0	0
35	0	1	0
36	0	-1	0
Total	4		

Aktivasi :  $\frac{1}{1+e^{-4}} = 0.982013790037908$ . Maka nilai *outputnya* ialah 0.982013790037908.

10. Kemudian lakukan pencarian nilai yang terbesar berdasarkan nilai *outputnya*.

## 5. Hasil Eksperimen

Berikut ini adalah skenario eksperimen yang dilakukan:

1. Masukan data-data acuan.
2. Mulai proses pembelajaran, jaringan saraf tiruan akan mempelajari data-data acuan.
3. Hentikan proses pembelajaran (tingkat error: 22,3760)
4. Lakukan pengujian terhadap huruf yang di-input. Aplikasi akan menunjukkan hasil pengenalan huruf, 2 huruf yang paling mirip dengan *inputan* dari pengguna (lihat kolom 1<sup>st</sup> dan 2<sup>nd</sup>)
5. Ulangi langkah no.2-4, dan hentikan pada tingkat error yg lebih rendah
6. Dari sekian banyak pengujian yg dilakukan, hanya 4 pengujian saja yg dimunculkan disini (dimana saat pengujian berikutnya dilakukan tingkat error berhenti pada nilai: 8,221; 3,0669; 0,0282)
7. Hasil pengenalan huruf yang terbaik diperoleh ketika nilai error paling kecil.

Tabel 3 adalah pengujian yang dilakukan berdasarkan tingkat error yang berbeda untuk mendapatkan hasil yang terbaik. Tabel ini ditentukan langsung saat melakukan pengujian terhadap aplikasi. Nilai error dicatat ketika aplikasi diperintahkan berhenti belajar. Pengujian dilakukan berulang kali, namun yang ditampilkan pada Tabel 3 hanya 4 pengujian dengan 4 nilai error yg berbeda. Persentase yang tertera pada Tabel 3 adalah persentase kemiripan input dengan hasil pengenalan. Hasil tertinggi (persentase terbesar) akan diperoleh ketika tingkat error paling rendah.

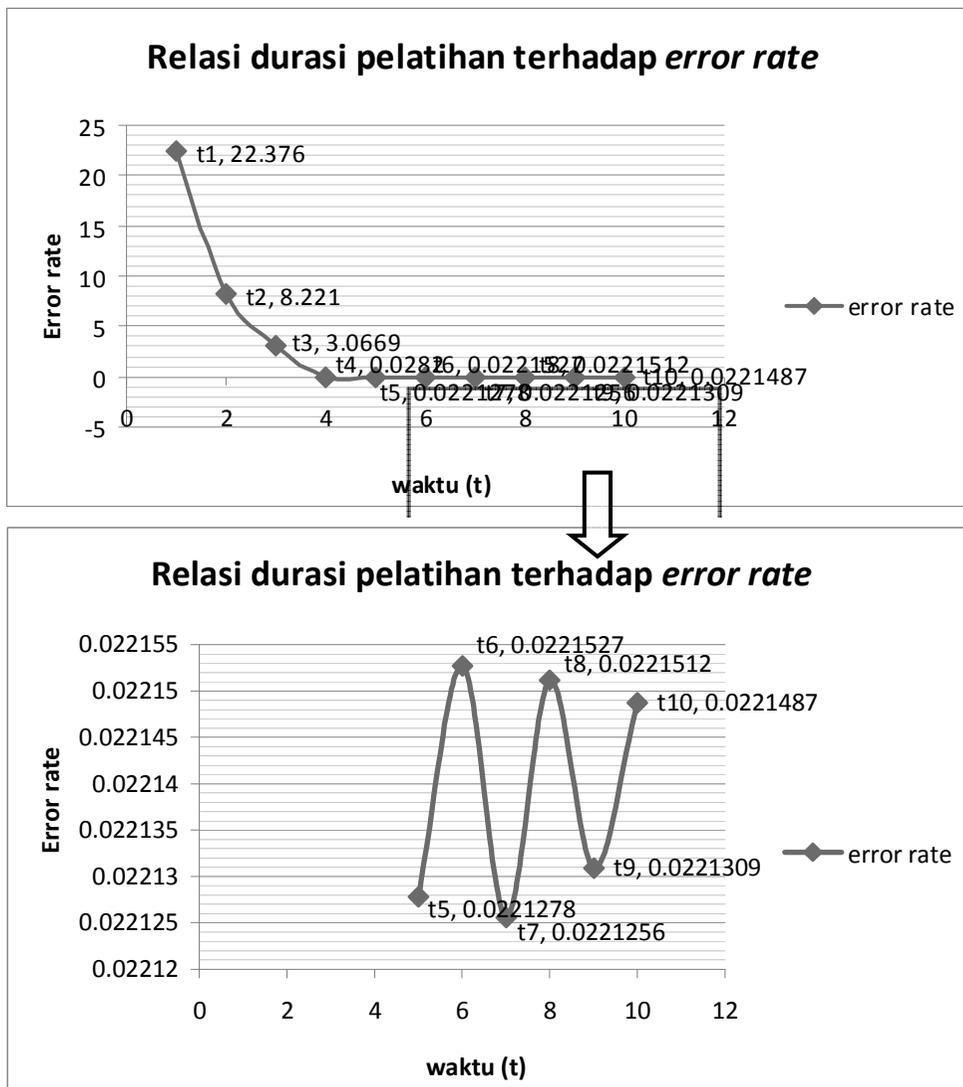
Tabel 3. Tabel pengujian tingkat *error* terhadap hasil pengenalan

No.	Huruf Input	Hasil pengenalan dengan tingkat error yang digunakan sbb. :							
		22,3760		8,221		3,0669		0,0282	
		1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>
1.									
		SE 3,3%	NU 2,7%	HE 85,5%	FU 0,5%	HE 56,2%	FU 1,2%	HE 96,9%	HA 0,1%

No.	Huruf Input	Hasil pengenalan dengan tingkat error yang digunakan sbb. :							
		22,3760		8,221		3,0669		0,0282	
		1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>
2.	く	め ME 3,3%	は HA 2,1%	を WO 13,1%	ん N 6,5%	く KU 76,7%	あ A 1,2%	く KU 90%	ひ HI 0,6%
3.	え	た TA 5,2%	あ A 2,6%	え <sup>E</sup> E 56,4%	あ A 0,0%	え E 90,7%	あ A 0,0%	え E 90,2%	あ A 0,0%
4.	ち	め ME 3,9%	か KA 1,7%	ち CHI 70,2%	あ A 11,6%	ち CHI 80,3%	あ A 6,5%	ち CHI 93,1%	あ A 1,3%
5.	こ	せ SE 5,4%	あ A 3,8%	こ KO 93,2%	ひ HI 17,5%	こ KO 98,4%	ひ HI 1,2%	こ KO 95,6%	ひ HI 1,8%
6.	し	る RU 6,6%	ら RA 6,1%	へ HE 1,8%	ふ FU 0,5%	し SHI 96,6%	き KI 2,2%	し SHI 93%	い I 15,5%
7.	み	ぬ NU 5,2%	の NO 3,3%	み MI 39,8%	あ A 2,7%	み MI 46,8%	ふ FU 5,2%	み MI 98,4%	え E 4,2%
8.	ゆ	わ WA 2,7%	く KU 2,5%	へ HE 3,4%	え E 1,7%	へ HE 6%	あ A 4,2%	ゆ YU 94,2%	か KA 1,3%
9.	さ	た TA 3,5%	れ RE 2,9%	さ SA 42,5%	ん N 2,6%	さ SA 69%	き KI 2,9%	さ SA 85,8%	れ RE 3,4%
10.	そ	を WO 3,6%	た TA 3,5%	を WO 27,3%	そ SO 9%	く KU 10,7%	い I 4%	そ SO 95,6%	さ SA 2,3%
11.	を	た TA 5,1%	ぬ NU 4,8%	を WO 10,8%	あ A 2,1%	を WO 45,8%	ら RA 2,8%	を WO 96,9%	み MI 10,9%
12.	ほ	う U 2,5%	あ A 1,8%	ほ HO 86,9%	ひ HI 0,2%	ほ HO 91,1%	は HA 0,3%	ほ HO 97,7%	は HA 2,8%
13.	ふ	ぬ NU 5,2%	ま MA 2,1%	を WO 47,6%	ふ FU 37,3%	ま MA 18,3%	ふ FU 27,1%	ふ FU 37,8%	ち CHI 1%
14.	ね	た TA 2,9%	れ RE 2,7%	ね NE 77,7%	く KU 2,5%	ね NE 87,3%	み MI 1,8%	ね NE 96,6%	む MU 1,4%

No.	Huruf Input	Hasil pengenalan dengan tingkat error yang digunakan sbb. :							
		22,3760		8,221		3,0669		0,0282	
		1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	1 <sup>st</sup>	2 <sup>nd</sup>
15.	や	の NO 3,2%	か KA 2,3%	ね NE 4,6%	け KE 3%	や YA 74,9%	え E 4,9%	や YA 91%	え E 2,7%

Pengujian diatas dilanjutkan beberapa kali, dan diperoleh nilai error yang sedikit naik turun, yaitu : 0,0221278; 0,0221527 ; 0,0221256 ; 0,0221512 ; 0,0221309; 0,0221487, namun demikian, nilai *errort* tetap akan semakin kecil yang berarti aplikasi ini tetap dapat mengenali input dengan baik. Bila digambarkan dengan grafik, maka dapat terlihat pada Gambar 7 berikut ini :



Gambar 7. Grafik relasi durasi pelatihan terhadap *error rate*

## **6. Kesimpulan**

Pengujian jaringan saraf tiruan yang dilakukan dengan menggunakan gambar pelatihan, bertujuan menguji ingatan jaringan, sebab kasus-kasus yang dimasukkan telah dipelajari sebelumnya. Ketika pengujian jaringan saraf tiruan dilakukan dengan menggunakan gambar baru yang belum pernah dipelajari oleh jaringan, yang diuji adalah kemampuan jaringan saraf tiruan dalam men-generalisasi-kasus yang dihadapi dan kemudian menarik kesimpulan yang cenderung ke output tertentu. Durasi pelatihan jaringan yang semakin lama akan membuat nilai error semakin menipis, sehingga tingkat keakurasian akan meningkat. Dengan demikian, komputer akan dapat membantu mengenali huruf yang diberikan (input) serta memberitahukan arti dan cara pembacaannya kepada pengguna. Hal ini dapat membantu sang pengguna dalam mempelajari bahasa Jepang.

## **7. Saran**

Saran – saran yang diberikan untuk pengembangan aplikasi selanjutnya adalah:

1. Memperbanyak varian pattern masing-masing huruf, sehingga memungkinkan penulisan huruf yang lebih fleksibel.
2. Memperluas ruang lingkup pengenalan huruf Jepang hingga ke huruf Kanji, bahkan huruf-huruf bahasa lain, misal : bahasa Mandarin, Korea.
3. Menambahkan fitur audio untuk membaca pelafalan huruf-hurufnya.
4. Menyediakan tutorial cara penulisan huruf Hiragana dan Katakana.

## **Daftar Pustaka**

- [Dar07] Darmawan Handoyo, Erico; Diktat Mata Kuliah Pengolahan Citra Digital; Universitas Kristen Maranatha, 2007, Bandung.
- [Int07] Internet World Stats. Top 10 Internet Languages. Retrieved October, 2007, from <http://www.internetworldstats.com/stats7.htm>
- [Jap07] Japan Zone. Alphabets Hiragana, Katakana. Retrieved October, 2007, from <http://www.japan-zone.com/new/alphabet.shtml>
- [Kus04] Kusumadewi, Sri; Membangun Jaringan Syaraf Tiruan menggunakan MATLAB dan Excel link; Penerbit Graha Ilmu, 2004, Yogyakarta.
- [Mof07] Moffat, Charles Alexander; The Top Ten Languages of the World. Retrieved October, 2007, from <http://www.lilithgallery.com/articles/2005/languagesoftheworld.html>
- [Mun04] Munir, Rinaldi; Pengolahan Citra Digital dengan Pendekatan Algoritmik; Penerbit Informatika, 2004, Bandung.
- [Pus06] Puspitaningrum, Diyah; Pengantar Jaringan Saraf Tiruan; Penerbit Andi, 2006, Yogyakarta.

[Sai07] Saiga.Japanese Language. Retrieved October, 2007, from [http://www.saiga-jp.com/japanese\\_language.html](http://www.saiga-jp.com/japanese_language.html)

[Suy07] Suyanto; Artificial Intelligence; Penerbit Informatika, 2007, Bandung