

# SIMULASI PENGENALAN TULISAN MENGGUNAKAN LVQ (LEARNING VECTOR QUANTIZATION)

Fachrul Kurniawan, Hani Nurhayati  
Jurusan Teknik Informatika, Sains dan Teknologi Universitas Islam Negeri (UIN)  
Maulana Malik Ibrahim Malang

**Abstrak-** *Proses simulasi yang dilakukan untuk menghasilkan sebuah sistem pengenalan meliputi beberapa tahap, yaitu tahap pengolahan citra dan tahap pelatihan dan pengenalan. Tahap pengolahan citra dimulai dari Gray Scale, Thresholding, segmentasi, dan normalisasi. Pengolahan citra diperlukan untuk memudahkan pengolahan data gambar sebelum masuk dalam tahap pelatihan. Kemudian tahap kedua adalah pelatihan dan pengenalan. Pada tahap ini, metode LVQ mulai digunakan dan menentukan bobot, target error, maxepoch, dan laju pelatihan (Learning rate). Data yang dijadikan sebagai input adalah citra huruf yang dinormalisasi sehingga berukuran 20x20 dan berekstensi bitmap (.bmp). Simulasi pengenalan tulisan ini dilakukan dalam beberapa tahapan sehingga bisa dengan membuat pengenalan menjadi lebih tajam. Tolak ukur keberhasilan sistem pengenalan tulisan tangan ini adalah dengan menghitung nilai Termination Error Rate dan tingkat keakuratan dalam pengenalan tanda tangan. Dari simulasi ini diperoleh struktur JST dengan jumlah nilai learning rate 0,003 nilai target error 0,00001 dan jumlah epoch sebesar 10.000 karena dalam rentang epoch 1000 sampai 10000 perubahan epoch tidak mempengaruhi kinerja sistem. Sistem yang terbentuk mampu mengenali citra yang berisi huruf yang digunakan sebagai bobot dengan nilai keakuratan rata – rata sebesar 61,07% dan rata – rata keakuratan hasil pengenalan terhadap citra yang belum dilakukan pembelajaran sebesar 48,17%.*

**Kata Kunci :** *Pengolahan Citra, Learning Vector Quantization*

## 1. Pendahuluan

Penelitian mengenai pengenalan huruf tulisan khususnya tangan terus dikembangkan. Metode yang digunakan dalam penelitian pengenalan huruf tulisan tangan pada umumnya menggunakan metode pencocokan citra dan pendekatan statistik. Penggunaan dari metode ini akan berhasil baik jika digunakan untuk mengenali huruf cetak dengan tipe dan ukuran tertentu dan tidak akan berhasil baik jika digunakan untuk mengenali huruf tulisan tangan dengan tipe dan ukuran yang bisa berbeda. Karena keterbatasan dari metode pencocokan citra dan pendekatan statistik maka untuk mengenali huruf tulisan tangan diperlukan metode lain yang memungkinkan memberikan hasil yang lebih baik (Kusumoputro, dkk. 1999).

Pada simulasi ini disajikan metode pengenalan tulisan khususnya tangan

menggunakan jaringan syaraf tiruan. Jaringan syaraf tiruan telah banyak diaplikasikan pada *pattern recognition*, khususnya pada bidang pengenalan karakter.

Jaringan saraf LVQ (*Learning Vector Quantization*) adalah suatu metode klasifikasi pola yang masing-masing unit keluaran mewakili kategori atau kelas tertentu. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika 2 vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama.

Pada simulasi ini akan dilakukan pengujian terhadap citra berisi tulisan tangan yang tidak hanya berisi satu huruf

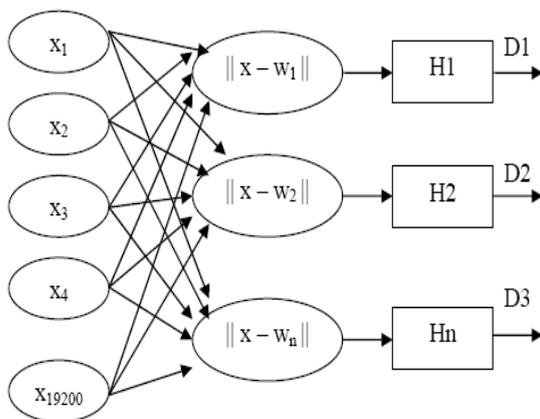
saja, melainkan beberapa huruf yang membentuk kata maupun kalimat. Oleh karena itu, sebelum proses jaringan syaraf tiruan dilakukan. Banyak langkah yang harus dilakukan sebelum tulisan tangan dapat dikenali oleh komputer yang meliputi pengumpulan data, analisis, *scanning*, binerisasi, segmentasi, dan klasifikasi.

## 2. Metode Learning Vector Quantization (LVQ)

Jaringan saraf tiruan merupakan representasi buatan yang mencoba mensimulasikan proses pembelajaran pada otak manusia. Jaringan saraf tiruan diimplementasikan dengan menggunakan program komputer sehingga mampu menyelesaikan proses perhitungan ketika proses pembelajaran berlangsung (Kusumadewi, 2003).

*Learning Vector Quantization (LVQ)* adalah metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas yang dihasilkan berdasarkan jarak vektor tersebut. Jika ada dua vektor memiliki jarak yang cukup dekat atau mendekati sama maka kedua vektor tersebut dikelompokkan ke dalam kelas yang sama.

Arsitektur LVQ dapat dilihat seperti gambar 2.10 berikut:



Gambar 1. Arsitektur LVQ

dengan:

- a)  $x_1$  sampai dengan  $x_{19200}$  = nilai input
- b)  $\|x - w_1\|$  sampai dengan  $\|x - w_n\|$  = jarak bobot
- c)  $H_1$  sampai dengan  $H_n$  = lapisan output
- d)  $D_1$  sampai dengan  $D_n$  = nilai output
- e)  $n$  = jumlah data karakter (jumlah kelas)

Gambar 2.10 merupakan arsitektur rancangan jaringan saraf tiruan dengan menggunakan *Learning Vector Quantization* dimana  $x_1$  sampai dengan  $x_{19200}$  merupakan elemen matriks dalam setiap pola karakter yang akan dijadikan sebagai nilai input. Kemudian  $\|x - w_1\|$  sampai dengan  $\|x - w_n\|$  merupakan perhitungan jarak bobot terkecil dengan  $w_1$  sampai dengan  $w_n$  adalah nilai data inisialisasi. Sedangkan  $H_1$  sampai dengan  $H_n$  adalah lapisan output dan  $D_1$  sampai dengan  $D_n$  adalah bobot akhir yang nantinya akan dipakai dalam proses pengujian dengan data karakter baru yang dimasukkan.

Pelatihan jaringan syaraf tiruan dikatakan berhasil jika pelatihan konvergen, dan gagal jika pelatihan divergen. Suatu pelatihan dikatakan konvergen jika kesalahan pada setiap iterasi pelatihan selalu mengecil, sampai pada titik dimana nilai bobot pada setiap neuron telah mencapai nilai yang paling baik untuk data pelatihan yang diberikan. Sebaliknya, pelatihan dikatakan divergen jika kesalahan pada pelatihan tidak cenderung mengecil menuju sebuah titik tertentu (Ali Akbar, 2007).

**Algoritma LVQ adalah sebagai berikut:**

### 1. Tetapkan :

bobot( $w$ ), maksimum epoch(maxEpoch), error minimum yang diharapkan(eps), learning rate( $\alpha$ ).

### 2. Masukan :

Input :  $x(m,n)$

Target :  $T(1,n)$

### 3. Tetapkan kondisi awal :

Created with

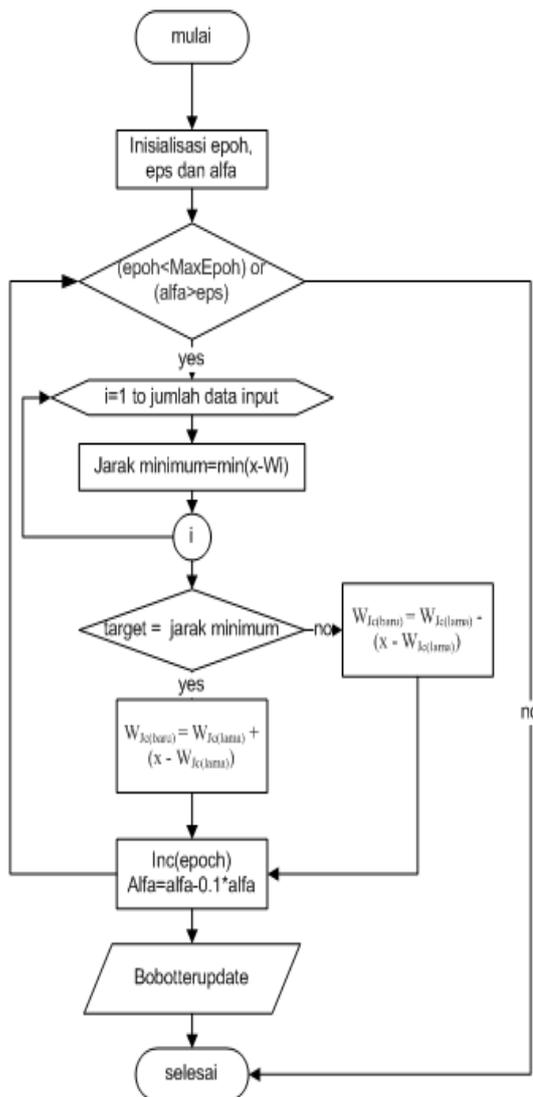
epoch = 0;

**4. Kerjakan jika :**

(epoch < maxEpoch) atau ( $\alpha > \text{eps}$ )

a. epoch = epoch + 1;

- i. Kerjakan untuk  $i = 1$  sampai  $n$   
Tentukan  $j$  sedemikian hingga  $\|x - w_j\|$  minimum (sebut sebagai  $C_j$ )
- ii. Perbaiki  $w_j$  dengan ketentuan:
  - jika  $T = C_j$  maka  $w_j(\text{baru}) = w_j(\text{lama}) + \alpha(x - w_j(\text{lama}))$
  - jika  $T$  tidak sama dengan  $C_j$  maka  $w_j(\text{baru}) = w_j(\text{lama}) - \alpha(x - w_j(\text{lama}))$
- b. Kurangi nilai  $\alpha$  Algoritma diatas bisa digambarkan dalam bentuk flow chart sebagai berikut ;

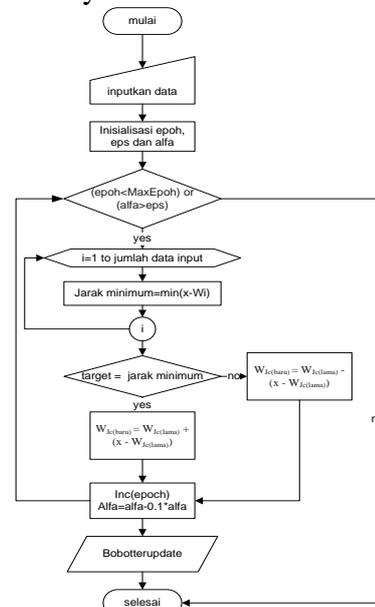


**Gambar 2.** Proses Algoritma LVQ

**3. Metode Perancangan Proses Simulasi Penulisan dan Pengenalan Tulisan**

Dalam proses pelatihan dan pengenalan dalam simulasi program, data yang dijadikan sebagai *input* adalah image berekstensi *Bitmap (.bmp)*. Kemudian menentukan *bobot* ( $W$ ), *Maksimum Epoch (MaxEpoch)*, *error minimum* yang diharapkan ( $Err$ ), dan *nilai learning rate* ( $\alpha$ ). Setelah menentukan parameter-parameter tersebut, image akan dipecah menjadi huruf perhuruf dalam bentuk matrix dan kemudian akan dikenali dengan cara membandingkan matrix tersebut dengan *knowledge* (bobot akhir) yang didapatkan dari proses training. Setiap huruf tersebut akan dibandingkan dengan seluruh bobot dan dihitung jaraknya sampai didapatkan jarak yang terkecil. Demikian satu persatu huruf dibandingkan sehingga semua huruf selesai dibandingkan. Bobot yang akan digunakan adalah bobot akhir yang telah diproses pada langkah pelatihan, yang terdiri dari empat jenis tulisan tangan.

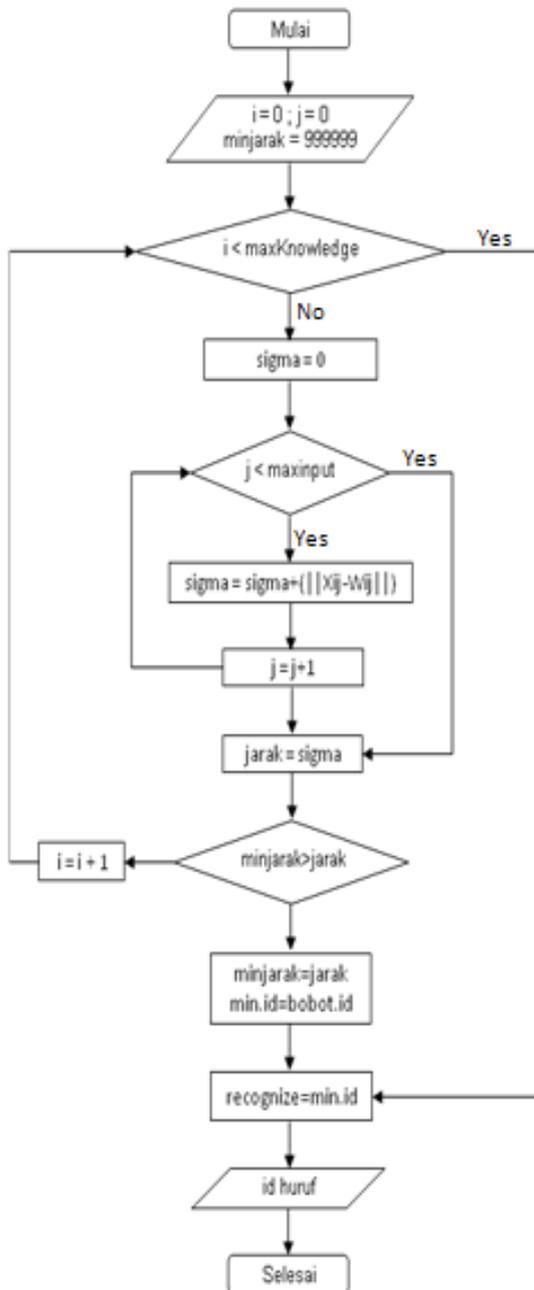
Berikut adalah alur dari proses pelatihan dimana masing-masing fungsi akan bisa kita lihat dalam logika matematikanya.



**Gambar 3.** Flow Chart Simulasi

### Proses Pelatihan Tulisan Tangan

Selanjutnya adalah proses yang terjadi pada pengenalan tulisan, sehingga langkah-langkah per logika akan bisa ketahu.



Gambar 4. Flow Chart Simulasi Proses Pengenalan Tulisan Tangan

## 4. Hasil Simulasi

### 4.1 Implementasi Jaringan Syaraf Tiruan

Setelah selesai dilakukan proses pengolahan citra, maka proses selanjutnya adalah proses jaringan syaraf

tiruan. Pada proses Jaringan Saraf Tiruan ini terdapat dua proes yaitu pembelajaran (*training*) dan pengenalan (*Recognition*).

#### a. Proses Pembelajaran

Pada proses pembelajaran ini pertama kali akan ditentukan maxEpoch, learning rate ( $\alpha$ ), *error* yang diharapkan ( $\epsilon$ ), dan pengurangan nilai  $\alpha$ . Setiap data latih akan dibandingkan dengan bobot dan akan disimpan setiap nilai pixelnya.

Pada proses pelatihan akan dilakukan perbaikan bobot yang dilakukan dengan cara mencari jarak *input* dan bobot yang sudah ditetapkan sebelumnya dengan rumus yang telah ditentukan. Lalu dilakukan perbaikan bobot dengan membandingkan apakah bobotupdate.iduser sama dengan pattern.iduser. Jika sama maka bobot diperbaiki dengan menjumlahkan bobot lama dengan selisih *data training* dan bobot lama dikalikan dengan learning rate. Jika berbeda, maka bobot diperbaiki dengan mengurangi bobot lama dengan selisih *data training* dan bobot lama dikalikan dengan *learning rate*. Proses iterasi di ulang dengan pengurangan nilai *learning rate* – (*learning rate* \* 0,1) dan dilakukan sampai *epoch* < MaxEpoch atau *learning rate* > *target error*.

#### b. Proses Pengenalan

Pada proses pengenalan mula-mula akan dipilih *knowledge* (bobot) yang akan digunakan. Selanjutnya setelah citra melalui pemrosesan citra, masing-masing citra akan dibandingkan dengan bobot yang sudah dipilih sebelumnya, lalu akan dicari jarak terkecilnya.

Pada proses pengenalan ini dilakukan perbandingan jarak *input* dengan bobotupdate yang didapatkan pada proses pelatihan. Jika jarak < dari minJarak maka id *input* sama dengan id *knowledge*.

### 4.2 Hasil Uji Coba Sistem

Untuk memperoleh hasil jaringan syaraf tiruan yang terbaik maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan cara melatih

jaringan syaraf tiruan dengan parameter-parameter yang berbeda. Dalam pengujian sistem syaraf tiruan ini sampel yang akan dilatih terbagi atas 5 orang yang berbeda, yaitu tulisan tangan Husna, Jamil, Arfi, Khosib, dan Fitri. Masing-masing sampel berjumlah 62 karakter sehingga jumlah total semua sampel adalah 310 karakter.

Berikut disajikan data-data hasil percobaan yang nantinya akan diambil satu struktur jaringan syaraf yang terbaik untuk pengenalan huruf.

a. Uji coba untuk mendapatkan nilai *Termination Error Rate* terbaik dengan melakukan uji coba nilai *learning rate*, *target error (epsilon)*, dan *maxepoch*.

**Tabel 1.** Hasil Percobaan Terhadap Nilai *Learning Rate*

Nilai <i>Learning Rate</i>	<i>MaxEpoch</i>	<i>Target error (epsilon)</i>	<i>Termination Error Rate</i>
0,001	10000	0,0001	9.69773729787523E-6
0,002	10000	0,0001	9.2767953731762E-6
0,003	10000	0,0001	9.12975816651136E-6
0,004	10000	0,0001	9.86013881983226E-6
0,005	10000	0,0001	9.98339055508017E-6
0,006	10000	0,0001	9.70385561953792E-6
0,007	10000	0,0001	9.17014356046334E-6
0,008	10000	0,0001	9.43214766219086E-6
0,009	10000	0,0001	9.55004950796825E-6

Hasil percobaan untuk mengetahui hasil keakuratan pelatihan jaringan terhadap nilai *learning rate*. Percobaan ini dilakukan menggunakan *maxepoch* sebanyak 10000 dan *target error* sebesar 0,0001. Dari percobaan ini diperoleh terjadi pada *learning rate* 0,003 sebesar 9.12975816651136E-6. *Termination Error Rate* terkecil.

Hasil percobaan untuk mengetahui hasil keakuratan pelatihan jaringan terhadap nilai *epsilon*. Percobaan ini dilakukan menggunakan *maxepoch* sebanyak 10.000 dan *learning rate* sebesar 0,003. Dari percobaan ini diperoleh *Termination Error Rate* terkecil yang terjadi pada *epsilon* 0,00001 sebesar 9.12975816651136E-6.

**Tabel 2.** Hasil Percobaan Terhadap Nilai *Epsilon*

Nilai <i>Learning rate</i>	<i>Max Epoch</i>	<i>Target error (epsilon)</i>	<i>Termination Error Rate</i>
0,003	10000	0,00001	9.12975816651136E-6
0,003	10000	0,00002	1.90880563234078E-5
0,003	10000	0,00003	2.90932118936257E-5
0,003	10000	0,00004	3.99083839418734E-5
0,003	10000	0,00005	4.9269609804782E-5
0,003	10000	0,00006	5.47440108942022E-5
0,003	10000	0,00007	6.75851986348175E-5
0,003	10000	0,00008	7.50946651497972E-5
0,003	10000	0,00009	8.3438516833108E-5

**Tabel 4.3** Hasil Percobaan Terhadap Nilai *MaxEpoch*

Nilai <i>Learning rate</i>	<i>Max Epoch</i>	<i>Target error (epsilon)</i>	<i>Termination Error Rate</i>
0,003	1000	0,00001	9.12975816651136E-6
0,003	2000	0,00001	9.12975816651136E-6
0,003	3000	0,00001	9.12975816651136E-6
0,003	4000	0,00001	9.12975816651136E-6
0,003	5000	0,00001	9.12975816651136E-6
0,003	6000	0,00001	9.12975816651136E-6
0,003	7000	0,00001	9.12975816651136E-6
0,003	8000	0,00001	9.12975816651136E-6
0,003	9000	0,00001	9.12975816651136E-6
0,003	10000	0,00001	9.12975816651136E-6

Dari percobaan ini diperoleh bahwa perubahan iterasi tidak memberikan dampak pada sistem. Hasil dari percobaan pertama digunakan sebagai parameter dari hasil pembelajaran dengan jumlah nilai *learning rate* sebesar 0.003, nilai *target error* sebesar 0.00001 dan *max epoch* sebesar 10000.

b. Uji coba untuk mendapatkan *input* citra yang terbaik dengan menggunakan *input* citra normal, *input* citra hasil penipisan (*thinning*), dan *input* citra hasil deteksi tepi (*edge detection*) dari data tulisan tangan yang telah diambil data sampelnya.

Pengujian menggunakan data latih dan knowledge masing-masing tulisan tangan.

**Tabel 4.4** Hasil Percobaan Data Latih Citra Normal

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	60	2	96.77%
Jamil	61	1	98.39%
Arfi	61	1	98.39%
Khosib	62	0	100%
Fitri	61	1	98.39%

**Tabel 4.5** Hasil Percobaan Data Latih Citra Thinning

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	61	1	98.39%
Jamil	61	1	98.39%
Arfi	61	1	98.39%
Khosib	61	1	98.39%
Fitri	61	1	98.39%

**Tabel 4.6** Hasil Percobaan Data Latih Citra Deteksi Tepi

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	54	6	87.10%
Jamil	61	1	93.55%
Arfi	61	1	98.39%
Khosib	61	1	98.39%
Fitri	58	4	48.39%

**Tabel 4.7** Hasil Percobaan Perbandingan Data Latih

DATA UJI	NORMAL	THINNING	DETEKSI TEPI
Husna	60/2	61/1	54/6
Jamil	61/1	61/1	61/1
Arfi	61/1	61/1	61/1
Khosib	62/0	61/1	61/1
Fitri	61/1	61/1	58/4
<b>TOTAL</b>	305/310 98.39%	305/310 98.39%	295/310 95.16%

**Tabel 4.8** Hasil Percobaan Data Uji Citra Normal

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	57	36	61.29%
Jamil	46	47	49.46%
Arfi	48	45	51.61%
Khosib	66	27	70.96%
Fitri	61	32	65.59%

Dari percobaan ini didapatkan bahwa citra hasil normal dan *thinning* paling bagus digunakan sebagai pengujian pada data latih itu sendiri dengan tingkat keakuratan 98.39%. Pengujian

menggunakan data uji dan knowledge masing-masing tulisan tangan.

**Tabel 4.9** Hasil Percobaan Data Uji Citra *Thinning*

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	26	67	27.85%
Jamil	22	71	23.66%
Arfi	36	57	38.71%
Khosib	24	69	25.81%
Fitri	28	65	30.11%

**Tabel 4.10** Hasil Percobaan Data Uji Citra Deteksi Tepi

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	43	50	46.24%
Jamil	27	66	29.03%
Arfi	34	59	36.56%
Khosib	43	50	46.24%
Fitri	45	48	48.39%

**Tabel 4.11** Hasil Percobaan Perbandingan Data Uji

DATA UJI	NORMAL	THINNING	DETEKSI TEPI
Husna	57/36	26/67	43/50
Jamil	46/47	22/71	27/66
Arfi	48/45	36/57	34/59
Khosib	66/27	24/69	43/50
Fitri	61/32	28/65	45/48
<b>TOTAL</b>	278/465 59.78%	136/465 29.25%	192/465 41.29%

Dari percobaan ini didapatkan bahwa citra normal paling bagus digunakan sebagai pengujian pada data uji dengan keakuratan 59.78%, sehingga untuk selanjutnya citra normal digunakan sebagai data uji. Uji coba yang dilakukan terhadap 5 data tulisan tangan yang telah diambil semua data sampel tulisan tangannya.

**Tabel 4.12** Hasil Percobaan Menggunakan Data Latih

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	61	1	98.39%
Jamil	61	1	98.39%
Arfi	60	1	98.39%
Khosib	62	0	100%
Fitri	61	1	98.39%
Rata-rata			98.71%

Dari percobaan ini didapatkan bahwa tingkat keakuratan paling rendah sebesar 98.39% dan paling tinggi sebesar 100%.

**Tabel 4.13** Hasil Percobaan  
Menggunakan Data Sampel Yang Telah  
Dilatih

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Husna	57	36	61.29%
Jamil	48	45	51.61%
Arfi	58	35	62.37%
Khosib	60	33	64.52%
Fitri	61	32	65.59%
Rata-rata			61.07%

Dari percobaan ini didapatkan bahwa tingkat keakuratan paling rendah sebesar 51.61% dan paling tinggi sebesar 65.59%. Uji coba yang dilakukan terhadap 5 data tulisan tangan yang belum diambil data sampel tulisan tangannya.

**Tabel 4.13** Hasil Percobaan  
Menggunakan Data Sampel Yang Belum  
Dilatih

DATA UJI	JUMLAH BENAR	JUMLAH SALAH	TINGKAT KEAKURATAN
Yaya	49	44	52.69%
Joni	35	58	37.63%
Ropa	46	47	49.46%
Ulin	45	48	48.39%
Ristri	49	44	52.69%
Rata-rata			48.17%

Dari percobaan ini didapatkan tingkat keakuratan paling rendah sebesar 37.63% dan paling tinggi sebesar 52.69%.

## DAFTAR PUSTAKA

Kusumadewi, Sri. (2003). *Artificial Intelligence ( Teknik dan Aplikasinya)*. Penerbit Graha Ilmu. Yogyakarta.

Jaeger, S., Liu, C.L., Nakagawa, M. (2003). *The State of The Art in Japanese Online Handwriting Recognition Compared to Techniques in Western Handwriting Recognition*. IJDAR (2003) vol. 6 pp.75-78.

Jong Jek Siang, M.Sc, Drs. (2004). *Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab*. Penerbit Andi. Yogyakarta.