

Pengenalan Aksara Batak Dengan Metode *Perceptron*

Frinto Tambunan

Teknik Informatika, Universitas Potensi Utama
Jl. K.L Yos Sudarso Km. 6.5, 20146
frintoaja@gmail.com

Abstract

Water is a good solvent, so that the water in nature is never pure. If the content of various substances and microorganisms contained in water exceeds the allowable threshold, the water quality will be compromised. Impaired water quality is said to be contaminated water. With so keep in mind how the level of contamination found in the water. With the construction of an artificial neural network using the Perceptron network is expected to be built capable of providing the level of water pollution prediction results.

Keywords: *Neural Networks, Perceptron, Aksara Batak, Training Data and Testing Data.*

Abstrak

Jaringan Saraf Tiruan merupakan salah satu aplikasi kecerdasan buatan yang mampu menyelesaikan masalah yang tidak terstruktur dan rumit serta mampu mengolah data-data input tanpa harus memiliki target. Salah satu modelnya adalah jaringan Perceptron. Model jaringan ini yang kemudian digunakan untuk menentukan pola Aksara Batak dengan menggunakan 3 variabel data seperti, a, ha, dan ma. Selanjutnya dibentuk beberapa arsitektur jaringannya dan dilanjutkan dengan melakukan pelatihan menggunakan data training dan pengujian menggunakan data testing pada setiap arsitektur. Dari hasil proses pelatihan dan pengujian maka diperoleh arsitektur jaringan yang terbaik dengan gradien error terendah. Hasil akhir menunjukkan bahwa Jaringan Saraf Tiruan model Perceptron dapat mengenali pola gambar yang diberikan dengan menggunakan arsitektur yang dibangun.

Kata kunci: *Jaringan Saraf Tiruan, Perceptron, Aksara Batak, Data Training dan Data Testing.*

1. Pendahuluan

Perkembangan ilmu pengetahuan dan teknologi telah membawa perubahan di hampir semua aspek kehidupan manusia dimana pembagian permasalahan dapat dipecahkan dengan upaya penguasaan dan peningkatan ilmu pengetahuan dan teknologi. Manusia dapat mengetahui ruang dari segi ciri-cirinya, yang diistilahkan dengan aksioma [1]. Tetapi hal ini dapat dipengaruhi oleh pengalaman atau mungkin dengan intuitifnya atau strategi pemetaan dalam berpikir, sedangkan untuk komputer, untuk mengenali dan membedakan suatu citra diperlukan data – data fisik yaitu bentuk dari ciri tersebut dan data – data lain seperti penambahan tekstur warna sehingga komputer akan lebih mudah untuk mengenalinya, namun dengan menggunakan metode yang ada dalam Jaringan Saraf Tiruan diharapkan komputer dapat mengenali sebuah citra objek sederhana [2].

Untuk penyelesaian permasalahan tersebut sangat dibutuhkan sebuah metode yang cocok dalam penyelesaian permasalahan tersebut dengan menggunakan metode *Perceptron*. Metode *Perceptron* adalah metode yang cukup handal dalam Jaringan Saraf Tiruan yaitu metode mengenali pola dengan baik, bisa dikatakan handal karena metode *Perceptron* ini memiliki prosedur belajar yang dapat menghasilkan bobot konvergen sehingga memungkinkan *output* yang didapat sesuai dengan target tiap input pola [3].

2. METODOLOGI PENELITIAN

Metode yang digunakan dalam perancangan jaringan saraf tiruan ini menggunakan metode *Perceptron* dikarenakan metode ini sangat baik dalam menangani masalah pengenalan pola-pola yang kompleks dan non-linear.

2.1. Defenisi Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (*Artificial Neural Network*) atau disingkat JST adalah sistem komputasi dengan arsitektur dan operasinya diilhami dari pengetahuan tentang sel saraf biologi di dalam otak. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi *aproksimasi nonlinear*, klasifikasi data, *cluster* dan *regresi non parametric* atau sebagai sebuah simulasi dari koleksi model saraf biologi[4].

JST merupakan generalisasi model matematis dengan beberapa asumsi berikut :

1. Pemrosesan informasi terjadi pada *neuron*.
2. Sinyal dikirimkan di antara *neuron-neuron* melalui penghubung *dendrite* dan *akson*.
3. Penghubungan antar elemen memiliki bobot yang akan menambah atau mengurangi sinyal.
4. Untuk menentukan *output*, setiap *neuron* memiliki fungsi aktivasi yang dikenakan pada jumlah semua inputnya. Besar *output* akan dibandingkan dengan nilai *threshold* tertentu.

Berdasarkan model matematis tersebut, baik tidaknya suatu model JST ditentukan oleh hal-hal berikut :

1. Arsitektur jaringan, yaitu sebuah arsitektur yang menentukan pola antar *neuron*.
2. Metode pembelajaran (*learning method*), yaitu metode yang digunakan untuk menentukan dan mengubah bobot.

2.2. Arsitektur Jaringan

Baik tidaknya suatu model JST salah satunya ditentukan oleh hubungan antarneuron atau yang biasa disebut sebagai arsitektur jaringan. Neuron-neuron tersebut terkumpul dalam lapisan-lapisan yang disebut *neuron layer*. Lapisan-lapisan penyusun JST dibagi menjadi tiga, yaitu :

1. Lapisan Input (*Input Layer*), unit yang bertugas menerima inputan dari luar yang menggambarkan suatu permasalahan.
2. Lapisan Output (*Output Layer*), unit yang merupakan solusi JST terhadap suatu permasalahan[5].

2.3. Analisa Pengumpulan dan Pembagian Data

Data-data yang digunakan dalam penelitian ini berasal dari proses pengambilan data sampel aksara sebanyak 1 buah dari jumlah keseluruhan aksara. Dengan pengambilan aksara tersebut akan diubah ke dalam piksel gambar dengan ukuran 7 x 9 piksel yang akan direduksi menjadi bilangan bilangan biner sebelum dilakukan pengujian.

Pada proses ini, bertujuan agar jaringan yang dibangun bisa mendapatkan pelatihan yang secukupnya dan pengujian untuk menguji data dari proses pelatihan yang telah dilakukan.

2.4. Analisa Perancangan Struktur Jaringan yang Optimum

Untuk menentukan vektor masukan tiap titik dalam pola yang diambil sebagai komponen vektor, segionggga setiap vektor masukan memiliki $9 \times 7 = 63$ masukan vektor titik dalam pola yang bertanda hitam dinyatakan dengan 1 dan yang berwarna putih dinyatakan dengan 0 dan pembacaan pola dilakukan dari atas ke bawah. Dari pola – pola didapatkan membentuk sebuah target yang diharapkan dapat melakukan perhitungan sehingga mendapatkan pengenalan terhadap pola yang telah ditentukan dan mendapatkan epoch. Adapun nilai target yang akan disesuaikan pada *output* dengan melalui perhitungan seperti yang akan di bahas pada poin berikutnya. Dengan demikian pada tabel 4.1 adalah target yang akan disesuaikan dengan nilai hasil.

Tabel 1. Tabel target

Pola masukan	Target
Pola 1	1
Pola 2	-1
Pola 3	-1

Perceptron akan melakukan perhitungan dengan mela sebuah sampel yang dapat mewakili perhitungan ke 63 elemen vektor yaitu :

Misalkan s sebagai vektor masukan, t adalah target keluaran, α adalah laju pemahaman, θ adalah nilai threshold. Perhatikan algoritma untuk pelatihan *Perceptron* di bawah ini :

Langkah pertama

Inisialisasi semua bobot dan bias (umumnya $w_i = b = 0$). Set laju pembelajaran α ($0 < \alpha \leq 1$) (untuk penyederhanaan set $\alpha = 1$).

Langkah kedua

Selama kondisi berhenti bernilai FALSE atau selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target ($y \neq t$), lakukan langkah-langkah 2 – 6. Atau selama *output* belum sesuai dengan input maka akan dilakukan perhitungan kembali seperti di atas.

Langkah ketiga

Untuk setiap pasangan (s, t), kerjakan langkah 3 – 5. Pada langkah ini epoch = epoch + 1. Epoch atau iterasi akan berhenti jika $y = t$ atau tercapainya epoch maksimum.

Langkah keempat

Set aktivasi unit masukan $x_i = s_i$ ($i = 1, \dots, n$)

Langkah kelima

Hitung respon untuk unit *output* :

$$y_{in} = \sum_i^n (x_i * w_i) + \dots + (w_i * w_n) + b \quad (1)$$

$$y_{(net)} = \begin{cases} 1 & \text{jika } net > \Theta \\ 0 & \text{jika } -\Theta \leq net \leq \Theta \\ -1 & \text{jika } net < -\Theta \end{cases} \quad (2)$$

Langkah keenam

Untuk menghitung nilai error yaitu jika nilai yang diharapkan atau target tidak sesuai dengan nilai hasil yang didapatkan melalui perhitungan dilakukan dengan menggunakan rumus :

$$e_p = y_{\Delta_p} - y_p \quad (3)$$

Langkah ketujuh

Perbaiki bobot dan bias pola jika terjadi kesalahan, $y \neq t$. Jika pada setiap epoch diketahui bahwa keluaran jaringan tidak sama dengan target yang diinginkan, maka bobot harus di ubah menggunakan rumus :

$$w_i = \alpha * t * x_i \quad (4)$$

Dan untuk mencari nilai wight baru adalah dengan melakukan rumus seperti dibawah ini :

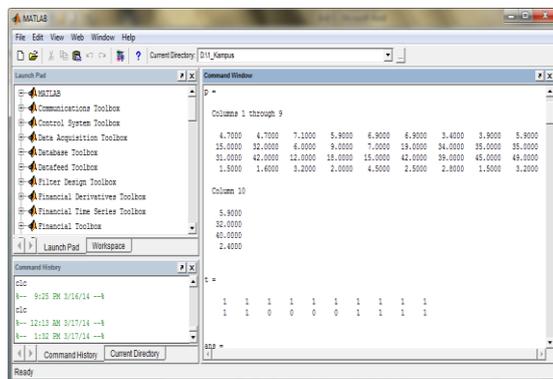
$$\Delta w_i = bobolama + w_i \quad (5)$$

Langkah ketujuh

Test kondisi berhenti, jika tidak terjadi perubahan bobot pada epoch tersebut maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE.

2.5. Penggunaan Aplikasi

MATrix Laboratory (Matlab) adalah suatu bahasa pemrograman untuk komputasi numerik yang berorientasi pada matriks. Pada gambar 1 menunjukkan tampilan *matriks* data *training* yang diinputkan ke dalam program MATLAB [5].



Gambar 1. Tampilan *Matriks* Data *Training* dalam Matlab

2.6. Pengujian Metode *Perceptron*

2.6.1. Tahap Pelatihan dan Tahap Pengujian

Algoritma pelatihan *Perceptron* adalah sebagai berikut :

Inisialisasi semua bobot dan bias (umumnya $w_i = b = 0$) tentukan laju pemahaman ($= \alpha$). Untuk penyederhanaan biasanya diberikan nilai $\alpha = 1$. Selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target, lakukan :

Set aktivas untuk masukan $x_i = s_i (i = 1, \dots, n)$

Hitung respon unit keluaran :

$$y_{in} = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{jika } y_{in} > +\theta \\ 0 & \text{jika } -\theta \leq y_{in} \leq +\theta \\ -1 & \text{jika } y_{in} < -\theta \end{cases}$$

Perbaiki bobot pola yang mengandung kesalahan ($y \neq t$) menurut persamaan :

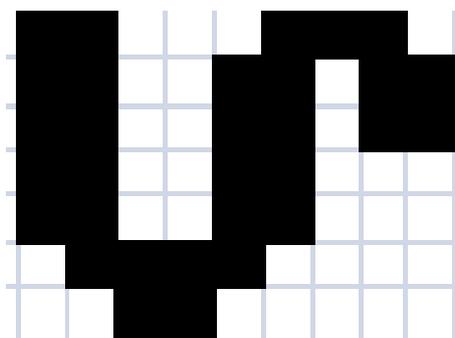
$$w_{i,baru} = w_{i,lama} + \Delta w (i = 1, \dots, n)$$

dengan

$$\Delta w = \alpha x_i b(\text{baru})$$

Pada permasalahan ini arsitektur Jaringan Saraf Tiruan yang digunakan adalah Jaringan Saraf Tiruan dengan algoritma *Perceptron*. Untuk menguji kinerja neuron ini diperlukan masukan yang sudah dimodifikasi sedemikian rupa berbeda dari data masukan awal, namun polanya masih kelihatan. Berikut adalah data awal dan data yang telah dimodifikasi (Tabel 1) untuk masukan masing aksara pertama, kedua dan ketiga. Dengan menggunakan variabel target yang sama dengan saat pembelajaran, dapat disimulasikan apakah arsitektur jaringan yang dibuat dapat mengidentifikasi data masukan yang telah

dimodifikasi. Berhasil tidaknya proses identifikasi ditentukan dengan kesamaan keluaran jaringan neuron dengan target yang telah ditetapkan. Selanjutnya, dilakukan langkah-langkah yang sama seperti pada saat pembelajaran. Perbedaannya adalah pada saat pembelajaran ini tidak diperlukan lagi instruksi train. Berikut data masukan untuk masing - masing huruf yang sudah dimodifikasi :

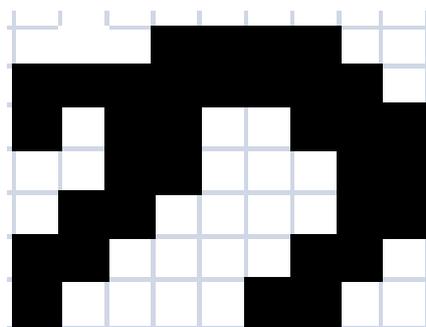


Gambar 2. Aksara Pertama

Gambar 2 Aksara pertama ini aksara yang dipecah dari gambar 2 sehingga aksara inilah yang akan diubah menjadi nilai biner seperti yang tertera di bawah ini. Dengan gambar di atas didapatkan pola sebagai berikut untuk aksara pertama.

1111100 - 1111110 - 0000011 - 0000011 - 0111110 - 1111100 - 1000000 -
1110000 - 0110000.

Untuk aksara berikutnya ada pada gambar 3.

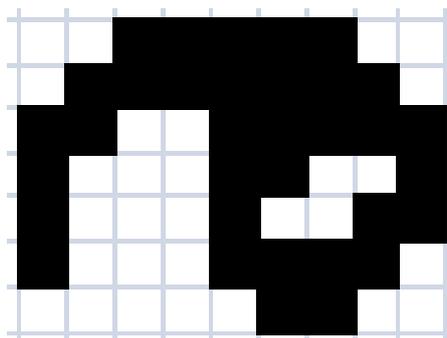


Gambar 3. Aksara Kedua

Gambar 3 Aksara kedua ini aksara yang dipecah dari gambar 4.2 sehingga aksara inilah yang akan diubah menjadi nilai biner seperti yang tertera di bawah ini. Dengan gambar di atas didapatkan pola sebagai berikut untuk aksara kedua :

0110011 - 0100110 - 0111100 - 1111000 - 1100000 - 1100001 - 1110011 -
0111110 - 0011100

Untuk aksara berikutnya ada pada gambar 4.5.

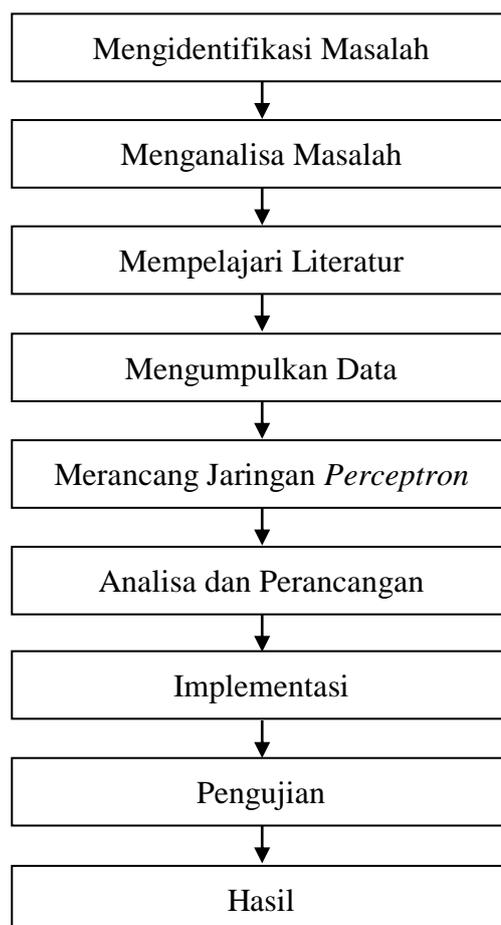


Gambar 4. Aksara Ketiga

Gambar 4 Aksara ketiga ini aksara yang dipecah dari gambar 4.2 sehingga aksara inilah yang akan diubah menjadi nilai biner seperti yang tertera di dibawah ini. Dengan gambar di atas didapatkan pola sebagai berikut untuk aksara ketiga.

0011110 - 0110000 - 1100000 - 1100000 - 1111110 - 1111011 - 1110011 -
0110110 - 1100011

Perancangan Manual Jaringan Saraf Tiruan :



Gambar 4. Kerangka Kerja (*Frame Work*) Penelitian

Menuliskan dan memberikan nilai bobot awal untuk untuk $w_1 = 0$, $w_2 = 0$, beserta dengan $\alpha = 0,8$, threshold = 0,5 dan target [1 -1 -1 -1]. Sehingga didapatkan nilai sebagai berikut : Bobot awal : $w = [0,0 \ 0,0]$ Bobot bias awal : $b = [0,0]$

Epoch ke-1

Data ke-1

$$y_{in} = 0,0 + 0,0 + 0,0 = 0,0$$

$$\text{Hasil aktivasi} = 0 \ (-0,5 < y_{in} < 0,5)$$

Target = 1

Bobot baru:

$$w_1 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$$

$$w_2 = 0,0 + 0,8 * 1,0 * 1,0 = 0,8$$

$$\text{Bobot bias baru} : b = 0,0 + 0,8 * 1,0 = 0,8$$

Data ke-2

$$y_{in} = 0,8 + 0,8 + 0,0 = 1,6$$

$$\text{Hasil aktivasi} = 1 \ (y_{in} > 0,5)$$

Target = -1

Bobot baru:

$$w_1 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$$

$$w_2 = 0,8 + 0,8 * -1,0 * 0,0 = 0,8$$

$$\text{Bobot bias baru} : b = 0,8 + 0,8 * -1,0 = 0,0$$

Data ke-3

$$y_{in} = 0,0 + 0,0 + 0,8 = 0,8$$

$$\text{Hasil aktivasi} = 1 \ (y_{in} > 0,5)$$

Target = -1

Bobot baru:

$$w_1 = 0,0 + 0,8 * -1,0 * 0,0 = 0,0$$

$$w_2 = 0,8 + 0,8 * -1,0 * 1,0 = 0,0$$

$$\text{Bobot bias baru} : b = 0,0 + 0,8 * -1,0 = -0,8$$

Data ke-4

$$y_{in} = -0,8 + 0,0 + 0,0 = -0,8$$

$$\text{Hasil aktivasi} = -1 \ (y_{in} < -0,5)$$

Target = -1

untuk mempersingkat perhitungan disini saya langsung memasukkan perhitungan ke epoch yang ke 9

Epoch ke-9

Data ke-1

$$y_{in} = -3,2 + 1,6 + 1,6 = 0,0$$

$$\text{Hasil aktivasi} = 0 \ (-0,5 < y_{in} < 0,5)$$

Target = 1

Bobot baru:

$$w_1 = 1,6 + 0,8 * 1,0 * 1,0 = 2,4$$

$$w_2 = 1,6 + 0,8 * 1,0 * 1,0 = 2,4$$

$$\text{Bobot bias baru} : b = -3,2 + 0,8 * 1,0 = -2,4$$

Data ke-2

$$y_{in} = -2,4 + 2,4 + 0,0 = 0,0$$

Hasil aktivasi = 0 ($-0,5 < y_{in} < 0,5$)
Target = -1
Bobot baru:
 $w_1 = 2,4 + 0,8 * -1,0 * 1,0 = 1,6$
 $w_2 = 2,4 + 0,8 * -1,0 * 0,0 = 2,4$
Bobot bias baru : $b = -2,4 + 0,8 * -1,0 = -3,2$
Data ke-3
 $y_{in} = -3,2 + 0,0 + 2,4 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1
Data ke-4
 $y_{in} = -3,2 + 0,0 + 0,0 = -3,2$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1
Epoh ke-10
Data ke-1
 $y_{in} = -3,2 + 1,6 + 2,4 = 0,8$
Hasil aktivasi = 1 ($y_{in} > 0,5$)
Target = 1
Data ke-2
 $y_{in} = -3,2 + 1,6 + 0,0 = -1,6$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1
Data ke-3
 $y_{in} = -3,2 + 0,0 + 2,4 = -0,8$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1
Data ke-4
 $y_{in} = -3,2 + 0,0 + 0,0 = -3,2$
Hasil aktivasi = -1 ($y_{in} < -0,5$)
Target = -1

Pada epoh ke-10 ini sudah tidak terjadi perubahan bobot, sehingga proses pembelajaran dihentikan. Hasil akhir diperoleh: Nilai bobot, $w_1 = 1,6$; dan $w_2 = 2,4$. Bobot bias, $b = -3,2$. Dengan demikian garis yang membatasi daerah positif dengan daerah nol memenuhi pertidaksamaan: $1,6 x_1 + 2,4 x_2 - 3,2 > 0,5$. Sedangkan garis yang membatasi daerah negatif dengan daerah nol memenuhi pertidaksamaan: $1,6 x_1 + 2,4 x_2 - 3,2 < -0,5$.

Pada pembahasan ini kembali melakukan korelasi antara nilai *output* dan nilai target yang telah ditentukan sebelumnya dan nilai *output* yang didapatkan sampai epoh yang ke-10, dengan mengidentifikasi proses epoh ke-1 sampai dengan epoh ke-10 didapatkan nilai yang sama antara nilai *output* [1 -1 -1 -1] dan nilai target sama dengan seperti dideklarasikan sebelumnya.

Dengan mendapat nilai yang sama antara target dan hasil *output* dari hasil yang dikeluarkan oleh *Perceptron* maka didapatkan kesimpulan bahwa pengenalan pola yang dilakukan oleh *Perceptron* dapat bekerja dengan baik dan mendapatkan

epoch yang sampai nol yang ini mengartikan bahwa pengenalan pola yang dilakukan oleh *Perceptron* adalah baik.

Sehingga dalam pengenalan ini *Perceptron* dapat mengenali sebuah pola dengan baik dan benar, baik itu untuk aksara pertama, kedua dan ketiga dan sesuai dengan perhitungan yang dilakukan.

4. SIMPULAN

Penelitian yang dilakukan oleh penulis tentang pengenalan pola aksara batak dengan menggunakan *Perceptron* menghasilkan beberapa kesimpulan:

- a. Jaringan Saraf Tiruan mampu mengenali aksara batak dan mampu menganalisa dengan hasil yang tepat dengan membandingkan nilai hasil *output* dan nilai target yang telah dimasukkan terlebih dahulu.
- b. Pengenalan pola yang dilakukan oleh Jaringan Saraf Tiruan bekerja dengan melakukan bivolar untuk mengekstrak nilai sebuah aksara ke dalam sebuah bilangan biner dan nilai biner ini akan diteruskan pada pemberian nilai bobot dan bias, dan setelah mendapatkan nilai tersebut maka dilakukan pencarian epoch.
- c. Jaringan Saraf Tiruan melakukan pembelajaran dengan melalui beberapa epoch mulai dari epoch pertama sampai Jaringan Saraf Tiruan mendapatkan nilai *output* yang sama dengan nilai target yang telah ditentukan, setelah nilai sama maka pola tersebut dikenali dengan baik dan proses perulangan berhenti.

5. Saran

Hasil dan analisa, pengujian, dan kesimpulan dalam penelitian yang pengenalan pola dengan menggunakan metode *Perceptron*, maka penulis memberikan beberapa saran sebagai berikut :

1. Dengan adanya penelitian ini diharapkan adanya penelitian selanjutnya dengan memaksimalkan penelitian ini dengan menggabungkan tanda baca yang digunakan pada aksara batak sehingga aksara batak lebih sempurna dalam penulisannya sehingga dapat diimplementasikan ke dalam font Ms.Office.
2. Diharapkan dengan metode yang lain juga membahas aksara batak, dengan membahas masalah yang sama yaitu aksara batak sehingga didapatkan nilai yang membandingkan metode mana yang lebih maksimal dalam menyelesaikan masalah sehingga pada penelitian selanjutnya dapat lebih maksimal.
3. Dengan adanya penelitian ini yang dirancang dengan menggunakan software Matlab, diharapkan pada penelitian selanjutnya dapat menggunakan bahasa – bahasa yang lain sehingga menambah penelitian sebagai referensi terhadap permasalahan yaitu pengenalan pola, atau dalam kasus apapun yang terkait dengan Jaringan Saraf Tiruan.

DAFTAR PUSTAKA:

- [1] Anindita Septiarini (2012). "*Sistem Pengenalan Karakter Dengan Jaringan Saraf Tiruan Algoritma Perceptron*". Volume 3. Informatika Mulawarman.
- [2] Keur, et al, (2013). "Analysis of Multi layer *Perceptron* Network". Volume 7. International Journal Of Computers & Technology.
- [3] Omri, et al, (2010). "Channel Estimation For Lte Uplink System By *Perceptron* Neural Network". Volume 2. International Journal of Wireless & Mobile Networks
- [4] Bhattacharjee, et al, (2011). "A Parallel Framework for Multilayer *Perceptron* for Human Face Recognition". Volume 3. International Journal of Computer Science and Security.
- [5] I Made Gede Sunarya (2013). "Sistem Biometrika Identifikasi Tanda Tangan Menggunakan Metode Jaringan Saraf Tiruan Model *Perceptron*". Volume 7. Informatika.
- [6] Liza Afriyanti (2010). "Rancang Bangun Tool untuk Jaringan Saraf Tiruan (JST) Model *Perceptron*". SNATI 2010.
- [7] Regina Simbolon (2013). "Perangkat Lunak Untuk Identifikasi Dan Pengenalan Huruf Braille Dengan Algoritma *Perceptron*". Volume IV. Pelita Informatika Budi Darma.
- [8] Rimico Adyaksyah at.all (2012.) "Perancangan Sistem Kriptanalisis RSA Menggunakan Jaringan Saraf Tiruan *Perceptron*". Volume I, Jurnal Teknik Pomits.
- [9] Mike Susmikanti (2010). "Pengenalan Pola Berbasis Jaringan Saraf Tiruan Dalam Analisa Ct Scantumor Otak Beligna". SNATI. Jogjakarta.
- [10] Ardi Pujianta (). "Pengenalan Citra Objek Sederhana Dengan Jaringan Saraf Tiruan Metode *Perceptron*". Volume 3. Jurnal Informatika.