



Available online at : <http://bit.ly/InfoTekJar>

# InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



## Sistem Pengenalan Huruf Latin Dengan Metode *Perceptron* Berbasis *Neural Network*

Rizalul Akram<sup>1</sup>, Novianda<sup>1</sup>, Khairul Muttaqin<sup>1</sup>, Rozzi Kesuma Dinata<sup>2</sup>

<sup>1</sup>Program Studi Teknik Informatika, Universitas Samudra, Kota Langsa 24416, Indonesia

<sup>2</sup>Program Studi Teknik Informatika, Universitas Malikussaleh, Aceh Utara 24351, Indonesia

### KEYWORDS

Manusia, huruf latin, mata, metedo perceptron

### CORRESPONDENCE

Phone: +6285260969695

E-mail: rizalulakram@unsam.ac.id

### A B S T R A C T

Huruf adalah sesuatu yang setiap hari dilihat oleh manusia. Manusia dapat mengenal berbagai macam karakter huruf dalam tulisan dengan mata dan otaknya. Setiap huruf memiliki pola dan karakter yang berbeda-beda. Jenis huruf pada huruf latin terdiri dari 29 huruf. Saat ini huruf latin terdiri dari ratusan jenis huruf (font). Dengan banyaknya jenis huruf ini dapat dikenali dengan mudah oleh manusia. Dalam penelitian ini penulis mencoba menciptakan sebuah aplikasi yang dapat mengenali berbagai jenis huruf dengan jenis font yang berbeda-beda. Program pengenalan huruf ini dibangun dengan aplikasi Matlab 2017. Pola image yang menjadi inputan adalah citra hitam putih yang memiliki warna latar belakang putih dengan format file bitmap (.bmp) 16 bit, A-Z, a-z, 0-9, beserta tulisan tangan. Untuk dapat mengenali jenis huruf penulis merancang dan membangun aplikasi berbasis Neural Network dengan metode Perceptron. Metode perceptron diyakini cukup baik dalam proses pelatihan dan pengujian pengenalan huruf.

### INTRODUCTION

Huruf adalah sesuatu yang setiap hari dilihat oleh manusia. Manusia dapat mengenal berbagai macam karakter huruf dalam tulisan dengan mata dan otaknya. Setiap huruf memiliki pola dan karakter yang berbeda-beda. Jenis huruf pada huruf latin terdiri dari 29 huruf. Saat ini huruf latin terdiri dari ratusan jenis huruf (font).

Perkembangan teknologi saat ini telah menghasilkan banyak aplikasi untuk berbagai bidang kehidupan, tidak terkecuali juga bidang citra atau yang disebut juga image processing. Penggunaan image processing saat ini telah banyak dikembangkan dalam bidang pengenalan berbagai macam objek, seperti objek wajah manusia, sidik jari manusia, dan berbagai macam pengenalan objek lainnya dalam kehidupan sehari-hari.

Dalam hal pengenalan objek huruf latin karakter A-Z, a-z, saat ini penulis mengambil contoh sistem yang bangun oleh yohannes pangaribuan dan masdiana sagala dengan judul "Menerapkan Jaringan Syaraf Tiruan untuk Mengenali Pola Huruf Menggunakan Metode Perceptron". Dalam sistemnya mendapatkan hasil pengenalan citra masukan untuk huruf kecil

sebesar 84,62 % (dengan percobaan uji sebanyak lebih dari 5 kali pengujian per karakter) dan pengenalan citra masukan untuk huruf besar sebesar 92,3 % (dengan percobaan uji sebanyak lebih dari 5 kali pengujian per karakter).

Pada penelitian ini penulis membangun sistem pengenalan huruf latin A-Z, a-z, 0-9 dengan pengenalan 20 (dua puluh) jenis font dan tulisan tangan yang menjadi perbedaan dari penelitian-penelitian sebelumnya.

### METODE PENELITIAN

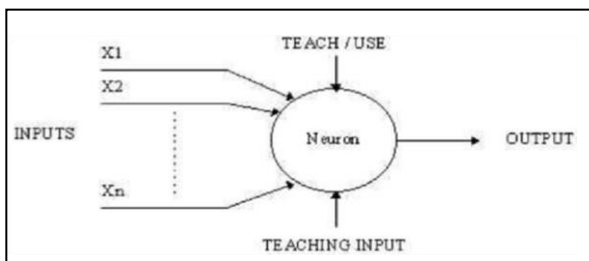
#### A. Jaringan Saraf Tiruan (Neural Network)

Menurut T. Sutojo, dkk (2011), Jaringan Saraf Tiruan (JST) adalah paradigma pemrosesan informasi yang terinspirasi oleh sistem saraf secara biologis, sama seperti otak yang memproses suatu informasi. JST bisa dibayangkan seperti otak buatan didalam cerita-cerita fiksi ilmiah. Otak buatan ini dapat berpikir seperti manusia dan juga sepandai manusia dalam menyimpulkan sesuatu dari potongan-potongan informasi yang diterima. Elemen dasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi.

Arief Hermawan (2006), mendefinisikan JST adalah sistem komputasi dimana arsitektur dan operasinya diilhami dari pengetahuan tentang sel saraf biologis dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. JST tercipta sebagai suatu generalisasi model matematis dan komputasi untuk fungsi aproksimasi non-linier, klasifikasi data *cluster* dan regresi non-parametrik atau sebuah simulasi koleksi model saraf biologi dari pemahaman manusia (*human cognition*) yang didasarkan pada asumsi berikut :

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut *neuron*.
2. Isyarat mengalir diantara sel saraf melalui suatu sambungan penghubung setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan/ mengalikan isyarat yang dikirim melaluinya. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.

Diyah Puspitaningrum (2006), menjabarkan salah satu contoh pengambilan ide dari jaringan saraf biologis adalah adanya elemen-elemen pemrosesan pada JST yang saling terhubung dan beroperasi secara paralel.



Gambar 1. Struktur Jaringan Saraf sederhana  
Sumber : Andi Prasajo Jurusan Teknik Elektro  
Fakultas Teknik Universitas Diponegoro

#### Arsitektur Jaringan Saraf Tiruan

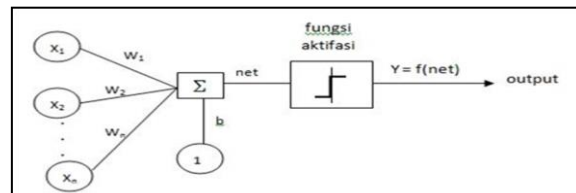
Sri Kusumadewi dan Sri Hartati (2010), menjabarkan hubungan antar neuron dalam jaringan saraf mengikuti pola tertentu tergantung pada arsitektur jaringan sarafnya dan dibagi kedalam 3 macam arsitektur jaringan saraf, yaitu :

1. Jaringan saraf dengan lapisan tunggal (*singel layer net*)  
Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima masukan kemudian secara langsung akan mengolahnya menjadi keluaran tanpa harus melalui lapisan tersembunyi (*hidden layer*).
2. Jaringan saraf dengan banyak lapisan (*multilayer net*)  
Jaringan saraf dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan masukan dan lapisan keluaran (memiliki satu atau lebih lapisan tersembunyi).
3. Jaringan saraf dengan lapisan kompetitif (*competitive layer net*)

Arsitektur ini memiliki bentuk yang berbeda, dimana antarneuron dapat saling dihubungkan. JST digambarkan dengan mengadopsi nilai dasar jaringan saraf biologi sebagai

berikut : menerima input atau masukan (baik dari data yang dimasukkan atau dari output sel saraf pada jaringan saraf). Setiap input datang melalui suatu koneksi atau hubungan yang mempunyai suatu bobot (*weight*). Setiap sel saraf mempunyai sebuah nilai batas ambang (*threshold*). Jumlah bobot dari input dan dikurangi dengan nilai ambang kemudian akan mendapatkan suatu aktivasi dari sel saraf (*post synaptic potential*, PSP dari sel saraf). Signal aktivasi kemudian menjadi fungsi aktivasi / fungsi transfer untuk menghasilkan keluaran dari sel saraf.

Jika tahapan fungsi aktivasi digunakan (keluaran sel saraf = 0 jika masukan < 0 dan 1 jika masukan  $\geq 0$ ) maka tindakan sel saraf tiruan sama dengan sel saraf biologi yang dijelaskan diatas (pengurangan nilai batas ambang dari jumlah bobot dan membandingkannya dengan 0 adalah sama dengan membandingkan jumlah bobot dengan nilai batas ambang). Biasanya tahapan fungsi jarang digunakan dalam Jaringan Saraf Tiruan. Fungsi aktivasi ( $f(n)$ ) dapat dilihat pada gambar 2.2.



Gambar 2. Fungsi Aktifasi

Sumber : Donna Ayu Silviana Universitas Airlangga, 2012

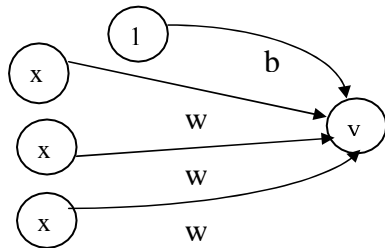
Diyah Puspitaningrum (2006), menuliskan pembagian JST berdasarkan struktur pola koneksinya kedalam dua kategori, yaitu: JST umpan maju (*feedforward networks*) dan JST berulang / umpan balik (*reccurent / feedback networks*).

#### B. Metode Perceptron

##### Arsitektur Perceptron

Output dari unit assosiator adalah biner vektor. Vektor tersebut dikatakan sebagai sinyal input terhadap sinyal output atau unit response. Oleh karena bobot-bobot dari assosiator ke unit output dapat diubah-ubah, maka yang akan diperhatikan hanya pada bagian layer tersebut.

Tujuan dari jaringan ini adalah mengklasifikasikan setiap pola *input* ke dalam kelas tertentu. Apabila *outputnya* +1, maka *input* yang diberikan termasuk kelas tertentu, sebaliknya jika *outputnya* -1, maka *input* yang diberikan tidak masuk dalam kelas tertentu. Arsitektur *perceptron* digambarkan seperti yang tampak pada gambar 2.3 berikut :



Gambar 3. Arsitektur *perceptron* sederhana

Keterangan :

- $x_1, \dots, x_i, \dots, x_n$  = neuron input
- $y$  = neuron output
- $b$  = bias
- $w_1, w_i, w_n$  = bobot

**Pelatihan Perceptron**

Metode jaringan *perceptron* merupakan salah satu model yang paling baik pada saat itu. Model ini ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969).

Bobot koneksi dari unit assosiator ke unit response (atau *output*) ditentukan melalui pelatihan (*learning rate*) *perceptron*. Untuk setiap *input* training jaringan akan menghitung response dari unit *output*, kemudian jaringan akan menentukan apakah suatu error terjadi pada pola tersebut dengan cara membandingkan *output* hasil perhitungan dengan nilai targetnya. Jaringan tersebut akan membedakan error antara *output* hasil perhitungan 0 dengan target -1 atau *output*nya +1 dengan target -1.

Dalam kedua kasus tersebut tanda dari error menunjukkan bahwa bobot koneksi harus diubah dalam arah yang dinyatakan oleh nilai target. Namun demikian hanya bobot-bobot pada koneksi dari unit pengiriman sinyal selain 0 ke unit *output* yang akan disesuaikan nilainya, karena hanya sinyal tersebut yang menambah error. Jika error tidak terjadi maka bobot-bobot tersebut tidak akan diubah tetapi sebaliknya jika suatu error terjadi untuk pola input pelatihan tertentu, bobot-bobot akan diubah menurut rumus:

$$w_i (\text{new}) = w_i (\text{old}) + a t x_i$$

Keterangan :

- $x_i$  = input ke -i
- $t$  = target yang nilainya +1 atau -1
- $a$  = kecepatan belajar (*learning rate*) yaitu  $0 < a \leq 1$
- $w$  = bobot

jika error tidak terjadi, maka bobot-bobot tidak akan berubah.

**Algoritma Perceptron**

Algoritma *perceptron* yang digunakan mempunyai susunan sebagai berikut :

Langkah 1:

- a. Inisialisasi bobot dan bias (agar sederhana bobot dan bias mula-mula dibuat = 0)
- b. Tentukan *learning rate* dengan nilai  $0 < a \leq 1$ . Untuk

penyederhanaan, a diberi nilai = 1

Langkah 2 : Selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target, lakukan :

- a. Set aktivasi unit masukan  $X_i = S_i$  ( $i = 1, \dots, n$ )
- b. Hitung respon unit keluaran :  $\text{net} = \sum x_i w_i + b$

$$y = f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases}$$

Langkah 3 : Perbaiki bobot pola yang mengandung kesalahan ( $y \neq t$ ) menurut persamaan :

- $w_i$  (baru) =  $w_i$  (lama) +  $\Delta w$  ( $i = 1, \dots, n$ )
- dengan  $\Delta w = a t x_i$   $b$  (baru) =  $b$  (lama) +  $\Delta b$  dengan  $\Delta b = a t$

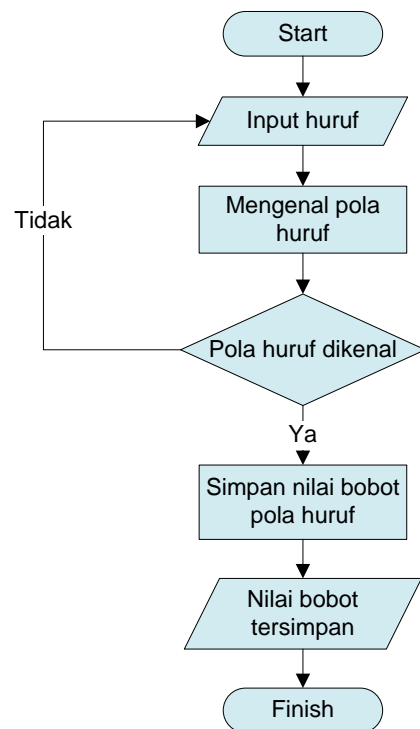
**HASIL DAN PEMBAHASAN**

**A. Rancangan Sistem**

Sistem pengenalan huruf latin dengan metode Perceptron ini untuk mengenali huruf latin A-Z, a-z, 0-9, huruf besar dan kecil. Selain itu juga untuk mengenali puluhan jenis font dan tulisan tangan. Sistem dirancang dengan inputan gambar berekstensi .bmp 16 bit.

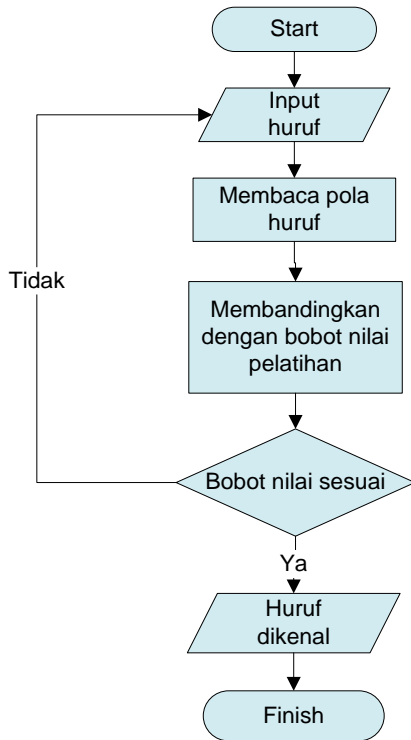
Tahapan pengenalan ini dibagi menjadi dua, yaitu tahapan pelatihan dan tahapan pengujian. Kedua tahapan tersebut bekerja seperti pada gambar flowchart 3.1 dan 3.2.

**1. Tahapan pelatihan**



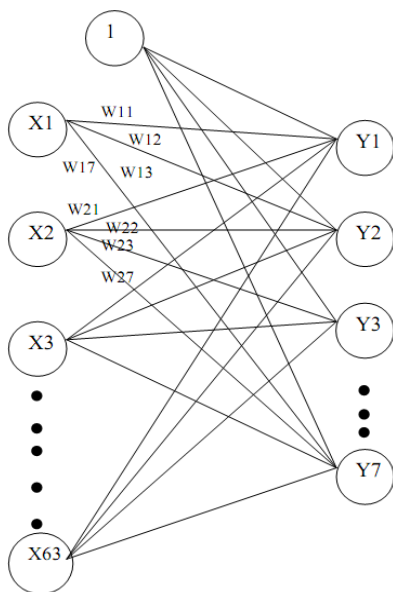
Gambar 4. Flowchart sistem pelatihan pengenalan huruf

2. Tahapan pengujian



Gambar 5. Flowchart sistem pengujian pengenalan huruf dan angka

Arsitektur jaringan saraf tiruan yang digunakan dalam sistem ini adalah metode perceptron dengan 63 neuron input dan 7 neuron output dengan arsitektur lapis tunggal (neuron hidden =0). Seperti terlihat pada gambar 3.3.



Gambar 6. Arsitektur jaringan saraf tiruan multilayer

Dari arsitektur jaringan yang telah dirancang didapat algoritma untuk melakukan pelatihan data yaitu :

1. Tentukan nilai  $\alpha$ .
2. Untuk  $i$  dari 1 sampai 7 lakukan langkah 3-7.
3. Untuk  $j = 1$  sampai 64 lakukan  $W_{ij}=0$ .
4. Selama kondisi =false lakukan langkah 5-7.

5. Untuk setiap pola pelatihan lakukan set aktivasi untuk input  $X_i = S_i$
6. Hitung respon unit output
7. Perubahan bobot

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot x_i \cdot y_i$$

Keterangan :

$w$  = Bobot

$x$  = biner dari huruf

$y$  = output

B. Hasil

Pelatihan Pengenalan Pola Huruf

Pelatihan yang digunakan untuk sistem ini adalah semua huruf latin A-Z (huruf besar), a-z (huruf kecil), tulisan tangan, dan angka 0-9 (angka besar dan kecil). Huruf dan angka yang diinputkan di tulis langsung di dalam sistem dan dapat diinputkan gambar dengan ekstensi file gambar bitmap (.bmp). Dalam hal ini *user* melatih huruf dan angka yang diinputkan untuk kemudian disimpan nilai bobotnya. Sampel pelatihan yang digunakan adalah 20 jenis font huruf beserta angka dengan total pelatihan 300 kali.

Pengujian Pengenalan Huruf

Setelah dilakukan pelatihan selanjutnya dilakukan proses pengenalan huruf dan angka, Tahapan ini berfungsi sebagai proses keluaran dari seluruh proses. Tahapan pelatihan adalah untuk mendapat bobot nilai pada file huruf dan angka dari pelatihan yang telah diinputkan selanjutnya dilakukan proses yang dilakukan adalah pengenalan huruf dan angka. Pengujian dilakukan dengan menginputkan beberapa sampel huruf dan angka yang hanya terdiri dari satu huruf ataupun satu angka saja.

Hasil pengukuran unjuk kerja dari sistem yang telah dibangun seperti pada tabel 3.1 dibawah. Pelatihan dilakukan terhadap 20 jenis font pada huruf latin, termasuk tulisan tangan. Pengujian dilakukan terhadap lebih dari 20 jenis font dan tulisan tangan.

Tabel 1. Hasil unjuk kerja sistem

Jumlah Pelatihan	Jumlah Pengujian	Hasil pengujian	
		Benar	Tidak dikenal
120	60	24	36
220	60	30	30
300	60	35	25

Hasil yang dipaparkan pada tabel unjuk kerja memperlihatkan semakin tinggi jumlah pelatihan yang dilakukan semakin tinggi pula tingkat kebenaran dalam hal pengenalan huruf/angka dan sebaliknya semakin banyak jumlah pelatihan yang dilakukan maka tingkat kesalahan dalam hal pengenalan huruf semakin kecil. Untuk memperbaiki tingkat kesalahan dalam sistem ini maka diperlukan perbaikan metode pelatihan, dimana setiap sampel huruf dan angka yang dilatih akan diuji. Selain itu juga harus ada perbaikan perhitungan pada sistem sehingga semakin

banyak pelatihan semakin akurat sistem dalam mengenal huruf dan angka dari berbagai jenis font.

Hasil pengujian dapat digunakan untuk memperbaiki nilai bobot pelatihan supaya hasil dari pengujian selanjutnya tingkat kebenaran dalam pengenalan huruf ataupun angka semakin meningkat.

#### Fungsi Pelatihan

Proses pengujian berfungsi untuk memperbaiki nilai bobot pelatihan supaya pengujian selanjutnya dalam pengenalan huruf maupun angka semakin meningkat kebenarannya atau semakin mudah dikenali.

#### Perhitungan Perceptron

Proses metode perceptron adalah tahapan terakhir setelah proses pembelajaran dengan jaringan multilayer. Berikut adalah contoh perhitungan perceptron untuk pembelajaran dengan inputan huruf A, vektor masukan untuk huruf A dan bobot awal (0,3 0,5 0,4 0,3 0,5 0,4 0,3) adalah sebagai berikut:

0 0 1 1 1 0 0

0 1 0 0 0 1 0

1 0 0 0 0 0 1

1 0 0 0 0 0 0

1 0 0 0 0 0 0

1 0 0 0 0 0 0

1 0 0 0 0 0 1

0 1 0 0 0 1 0

0 0 1 1 1 0 0

$$w1 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w1 = 0.3*0 + 0.5*0 + 0.4*1 + 0.3*1 + 0.5*1 + 0.4*0 + 0.3*0 = 1.2$$

$$w2 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w2 = 0.3*0 + 0.5*1 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*1 + 0.3*0 = 0.9$$

$$w3 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w3 = 0.3*1 + 0.5*0 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*0 + 0.3*1 = 0.6$$

$$w4 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w4 = 0.3*1 + 0.5*0 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*0 + 0.3*0 = 0.3$$

$$w5 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w5 = 0.3*1 + 0.5*0 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*0 + 0.3*0 = 0.3$$

$$w6 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w6 = 0.3*1 + 0.5*0 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*0 + 0.3*0 = 0.3$$

$$w7 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w7 = 0.3*1 + 0.5*0 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*0 + 0.3*1 = 0.6$$

$$w8 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w8 = 0.3*0 + 0.5*1 + 0.4*0 + 0.3*0 + 0.5*0 + 0.4*1 + 0.3*0 = 0.9$$

$$w9 = w1*i1 + w2*i2 + w3*i3 + w4*i4 + w5*i5 + w6*i6 + w7*i7$$

$$w9 = 0.3*0 + 0.5*0 + 0.4*1 + 0.3*1 + 0.5*1 + 0.4*0 + 0.3*0 = 1.2$$

Setelah perhitungan didapatkan bobot awal (w) untuk huruf A adalah (1,2 1,9 0,6 0,3 0,3 0,3 0,6 0,9 1,2). Bobot ini nantinya digunakan untuk pelatihan huruf A selanjutnya. Contoh hasil pada sistem ini adalah seperti pada gambar 3.4.



Gambar 7. Hasil percobaan

#### KESIMPULAN

Dari banyaknya sampel jenis font yang dilatih dan di uji pada sistem ini maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Inputan huruf dan angka ke dalam sistem ini masih berupa gambar yang diinput dengan ekstensi bitmap (.bmp) atau di tulis langsung didalam aplikasi.
2. Tingkat persentasi kebenaran semakin tinggi jika pelatihan dilakukan semakin banyak.
3. Jenis font yang mudah dikenal adalah jenis font yang sudah pernah dilatih
4. Hasil pengujian dapat digunakan untuk perbaikan nilai bobot pelatihan supaya hasil dari pengujian selanjutnya tingkat persentase kebenarannya dalam pengenalan huruf ataupun angka semakin meningkat.

#### REFERENSI

- [1] Pangaribuan Yohanes dan Sagala Masdiana, "Menerapkan Jaringan Saraf Tiruan untuk Mengenal Pola Huruf Menggunakan Metode Perceptron", Jurnal Teknik Informatika Unika St. Thomas (JTIUST), Volume 02 Nomor 02, Desember 2017, ISSN: 2548-1916.
- [2] Sutojo, T, dkk. 2011, "Kecerdasan Buatan", Yogyakarta: Penerbit Andi.
- [3] Arif Hermawan. (2006), "Jaringan Syaraf Tiruan (Teori dan Aplikasi)", Yogyakarta: Andi.
- [4] Diyah Puspitaningrum. 2006, "Pengantar Jaringan Syaraf Tiruan", Yogyakarta: Andi.
- [5] Sri Kusumadewi & Sri Hartati (2010), "Neuro Fuzzy:

*Integrasi Sistem Fuzzy & Jaringan Syaraf Edisi 2*,  
Yogyakarta: Graha Ilmu.

- [5] David. 2011, “*Perancangan Perangkat Lunak Pengenalan Pola Karakter Menggunakan Jaringan Syaraf Tiruan Perceptron*”, Pontianak: Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak.
- [6] Jong Jek Siang, “*Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan MATLAB*”, Andi Yoyakarta, 2009.