Jurnal SPIRIT Vol. 12 No. 1 Mei 2020, hal 9-15 P- ISSN : 2085 – 3092
E- ISSN : 2721 – 057X

PEMANFAATAN ARTIFICIAL NEURAL NETWORK DENGAN METODE HEBB RULE UNTUK PENGENALAN BAHASA ISYARAT INDONESIA STATIS

Dwi Setiawan[1)], Achmad Zakki Falani[2)]
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Narotama
Email : dwisetiawann8@gmail.com, achmad.zakki@narotama.ac.id

Abstrak: *Communication limitations are a social problem faced by persons with disabilities who are deaf and speech impaired. This problem is not only experienced by people who are deaf and speech impaired. Because it is less able to translate sign language this also becomes a problem for normal people in communicating with people who are deaf and speech impaired. Based on the problems outlined above, an analysis will be carried out for the introduction of static Indonesian sign language patterns. Where the data used is visual data in the form of pictures taken based on the visual form of the hand that refers to the Indonesian sign language type SIBI (Indonesian Sign System) using the artificial neural network method of hebb rule. Visual data in the form of images are obtained from capture results which are then collected and inputted through a system with the Python programming language as much as 72 training data which are then processed with several stages including preprocessing which has 3 stages namely grayscaling, edge detection, and thresholding. Furthermore, the data is processed at the segmentation stage and classification testing is performed on the test data using the hebb rule method which has a percentage of pattern recognition accuracy of 100% in the training data testing and an accuracy percentage of 80.37% in the test data obtained from the capture of 72 test data.*

Keywords : *Hebb Rule, Artificial Neural Network, Grayscale, Edge Detection, Threshold, Ekstraksi, Bahasa Isyarat Indonesia, SIBI.*

1. PENDAHULUAN

1.1. Latar Belakang

Keterbatasan dalam berkomunikasi menjadi masalah sosial yang dihadapi oleh penyandang disabilitas tuna rungu dan tuna wicara. Permasalahan ini tidak hanya dialami bagi penderita tuna rungu dan tuna wicara saja. Karena kurang bisa menerjemahkan bahasa isyarat hal ini pun menjadi masalah bagi orang normal dalam berkomunikasi dengan penderita tuna rungu dan tuna wicara.

Berdasarkan permasalahan yang telah diuraikan tersebut maka dalam penelitian ini akan dilakukan analisis untuk pengenalan pola bahasa isyarat Indonesia statis. Dimana data yang digunakan adalah data visual berupa gambar yang diambil berdasarkan bentuk visual tangan yang mengacu pada bahasa isyarat Indonesia jenis SIBI (Sistem Isyarat Bahasa Indonesia) dengan memanfaatkan *artificial neural network* metode *hebb rule.*

1.2. Tujuan

Adapun tujuan dari penelitian ini adalah penerapan *artificial neural network* metode *hebb-rule* untuk menerjemahkan bahasa isyarat Indoensia jenis SIBI kedalam bentuk teks terjemahan dengan menganalisis data visual yang berupa gambar dari bahasa isyarat Indonesia jenis SIBI. Serta mengetahui berapa besar hasil presentase akurasi pengenalan pola bahasa isyarat Indonesia jenis SIBI.

1.3. Manfaat

Adapun manfaat dari penelitian ini adalah pemanfaatan hasil analisis yang dapat membantu menerjemahkan bahasa isyarat Indoensia jenis SIBI kedalam bentuk teks terjemahan dan hasil presentase akurasi pengujian dapat dijadikan sebagai perbandingan dengan metode lain untuk mengembangkan penelitian ini pada penelitian selanjutnya.

2. TINJAUAN PUSTAKA

2.1. Bahasa Isyarat

Bahasa isyarat memiliki jenis yang sangat beragam. Walaupun dengan bahasa tertulis yang sama tetapi hampir di setiap negara memiliki jenis bahasa isyarat yang berbeda, begitupun sebaliknya tak sedikit juga di setiap negara memiliki bahasa isyarat yang sama walaupun dengan bahasa tertulis yang berbeda.

Contoh jenis bahasa isyarat yakni, *American Sign Language* (ASL), *British Sign Language* (BSL), Bahasa Isyarat Indonesia (BISINDO), Sistem Isyarat Bahasa Indonesia (SIBI), dan lain sebagainya.

2.2. Siatem Isyarat Bahasa Indonesia (SIBI)

Sistem Isyarat Bahasa Indonesia atau SIBI merupakan bahasa isyarat resmi yang telah di bakukan oleh pemerintah karena penggalan kata dan kalimatnya hampir sama menyerupai ASL dan memiliki struktur yang sama dengan tata bahasa lisan indonesia dimana SIBI digunakan untuk mempresentasikan tata bahasa lisan indonesia ke dalam sebuah bahasa isyarat. Berikut adalah gambar 2.1 isyarat huruf dari SIBI.

2.3. Mean Square Error (MSE)

Mean square error adalah sebuah metode yang digunakan untuk mengevaluasi nilai *error* dengan kuadrat rata-rata antara nilai asli pada data *training* dengan nilai hasil perhitungan *hebb rule.* Untuk rumus perhitungan *mean square error* adalah sebagai berikut :

MSE = $\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2$.....(1)

Dengan :

f_i : Nilai asli data training
y_i : Nilai hasil perhitungan hebb
n : banyaknya range nilai

2.4. Artificial Neural Network (ANN)

Artificial neural network atau jaringan syaraf tiruan (JST) merupakan sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. kemampuan yang dimiliki JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau input yang dimasukkan dan membuat predikasi tentang kemungkinan output yang akan muncul atau menyimpan karakteristik dari input yang disimpan kepadanya. (Hermawan, 2006)

2.5. Hebb Rule

Hebb rule merupakan suatu model jaringan syaraf tiruan tertua pada *supervised learning* yang dikenalkan pertama kali oleh *O.B. Hebb* pada tahun 1949. Dimana pada model *hebb-rule* ini memiliki beberapa unit masukan yang dihubungkan langsung dengan sebuah unit keluaran yang digunakan untuk menghitung bobot dan bias secara *continue* dengan memanfaatkan pembelajaran dengan supervisi yang menyebabkan bobot dan bias dapat dihitung secara otomatis tanpa harus melakukan uji coba. Cara yang untuk perbaikan bobot adalah :

w_i (baru) = w_i (lama) + x_i * y.....(2)

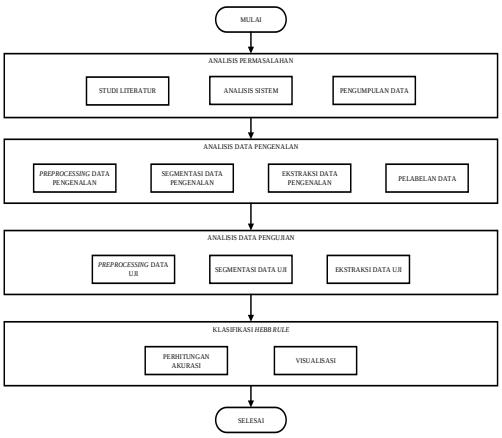
b (baru) = b (lama) + y.....(3)

Dengan :

w_i : bobot data input ke-i
x_i : input data ke-i
y : output data
b : nilai bias

3. METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai metodologi penelitian atau langkah-langkah yang digunakan dalam menyelesaikan penelitian ini. Dimana metodologi penelitian yang dilakukan yakni dimulai dari analisis permasalaha, analisis data pengenalan, analisis data pengujian, serta klasifikasi *hebb rule*.



Gambar 1. Metodologi Penelitian

4. HASIL DAN PEMBAHASAN

4.1. Analisis Data Pengenalan

Tahapan analisis data pengenalan yang dilakukan pada penelitian ini adalah dengan pengumpulan data sebagai data pengenalan, *preprocessing*, segmentasi, dan ekstraksi. Setelah melalui tahapan tersebut, data akan ditetapkan sebagai data label atau data *training.*

4.1.1. Pengumpulan Data

Pengumpulan data dilakukan dengan cara *input* data menggunakan sistem yang telah dibuat dengan menggunakan bahasa pemrograman *python*. Data yang digunakan pada penelitian ini merupakan data visual berupa gambar dari bahasa isyarat indonesia.

Gambar 2. Data pengenalan huruf A

4.1.2. *Preprocessing* Data Pengenalan

Preprocessing data pengenalan dilakukan pada data yang telah di *input* kan oleh sistem. Tahap pertama yakni dengan merubah data dengan citra RGB menjadi data dengan citra *grayscale.*

Gambar 3. Data *grayscale* huruf A

Setelah dilakukan proses *grayscaling* tahap selanjutnya adalah *edge detection*. berikut merupakan *source code* bahasa pemrograman *python* yang digunakan untuk proses *edge detection.*

Gambar 4. Data citra *threshold* huruf A

4.1.3. Segmentasi Data Pengenalan

Segmentasi data pada penelitian ini dilakukan pada data hasil *thresholding* yang digunakan untuk merubah data visual berupa gambar menjadi data angka yang memiliki *range* nilai piksel 0 dan 255 yang disajikan kedalam *file* dengan format *csv.* Dimana data tersebut memiliki nilai *range* kolom sebanyak 640 kolom dan nilai *range* baris sebanyak 480 baris. Banyaknya range kolom dan *range* baris tersebut didapatkan dari data gambar yang memiliki ukuran 640 piksel kali 480 piksel. Berikut adalah *source code* untuk proses segmentasi citra menjadi data angka.

Segmen Program 5 Proses Segmentasi

```
01 :csv=numpy.savetxt(('data_u
   ji/data_raw/data_uji.csv'),
   threshold, delimiter=',')
```

Dari *source code* tersebut akan dijelaskan pada baris ke-01 berisi mengenai *code library Numpy* dari *python* yang berfungsi untuk merubah data citra hasil dari proses *thresholding* menjadi data angka

4.1.4. Ekstraksi Data Pengenalan

Tahap selanjutnya setelah mendapatkan hasil dari tahap segmentasi yang berupa data csv adalah tahap ekstraksi data yang dilakukan untuk menyederhanakan banyaknya nilai data segmentasi. Adapun cara yang digunakan adalah dengan mencari mean dari data segmentasi tersebut. Ekstraksi data dilakukan dengan menjalankan *code* pemrograman *python*. Hasil dari ekstraksi data adalah nilai mean dari data segmentasi yang hanya memiliki jumlah range kolom sebanyak 1 kolom dan range baris sebanyak 640 yang disajikan ke dalam file dengan format csv. Tahap selanjutnya adalah merubah nilai data ekstraksi menjadi nilai fungsi bipolar yang hanya memiliki nilai -1 dan 1

4.1.5. Pelabelan Data

Sebelum melakukan tahap pelabelan yang harus dilakukan adalah dengan merubah format data ekstraksi yakni csv menjadi data format .xlsx agar tahap pelabelan dapat mudah diproses menggunakan bahasa pemrograman *python.* selanjutnya tahap pelabelan data pada penelitian ini dilakukan dengan menyimpan hasil dari ekstraksi data sebanyak 72 data menjadi 1 file .xlsx. tahapan yang dilakukan adalah dengan menjalankan bahasa pemrograman *python*.

4.2. Analisis Data Pengujian

Analisis proses pengujian data pada penelitian ini merupakan suatu proses yang dilakukan untuk mengetahui berapa besar

presentase akurasi mengenali pola data uji terhadap data pengenalan yang telah dijadikan sebagai acuan pengujian. Proses pengujian data dilakukan melalui beberapa tahapan, yakni melakukan pengambilan data uji dengan cara memanggil data uji dari sistem, setelah itu dilanjutkan pada tahap pengolahan data uji. Data uji yang telah diambil akan diolah dengan beberapa tahap, yakni *pro-processing*, segmentasi, dan ekstraksi.

4.2.1. *Input* Data

Input data pengujian merupakan tahap awal untuk pengujian data yakni dengan memanggil data yang akan diuji melalui sistem yang dijalankan dengan menggunakan bahasa pemrograman *python.* berikut merupakan *source code* pemrograman bahasa *python* pada proses *input* data uji dan *capture* data uji.

Segmen Program 7 Proses *Input* Data

```
01 : root = Tk()
02 : root.filename =
   filedialog.askopenfilename(tit
   le = 'Select file', filetypes
   = (('jpg files','*.jpg'),('all
   files','*.*')))
03 : imginput = root.filename
04 : print(imginput)
```

Dari *source code* tersebut akan dijelaskan pada baris ke-01 berisi mengenai *code library Tkinter* dari *python* yang berfungsi untuk menampilkan *directory.* Pada baris ke-02 digunakan untuk menentukan *type* data yang akan dipanggil, pada *source code* tersebut *type* data yang digunakan adalah *file* dalam format *.jpg*. padabaris ke-03 *variabel* "*imginput*" memiliki fungsi untuk memanggil data yang telah dipilih. Dan baris ke-04 yakni mencetak data yang telah dipanggil.

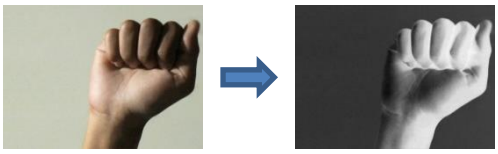
Segmen Program 8 Proses *Capture* Data Uji

```
01 : camera = cv2.VideoCapture(0)
02 : top, right, bottom, left =
   140, 177, 510, 670
03 : num_frames = 0 #inisialisasi
   num dari frame
04 : while(True):
05 :    (grabbed, frame) =
   camera.read()
06 :    frame =
   imutils.resize(frame,
   width=700)
07 :    frame = cv2.flip(frame, 1)
08 :    clone = frame.copy()
09 :    (height, width) =
   frame.shape[:2]
10 :    roi = frame[top:bottom,
   right:left]
```

Dari *source code* tersebut akan dijelaskan pada baris ke-01 berfungsi untuk mengaktifkan *webcam* pada laptop. Pada baris ke-02 adalah *source code* untuk menentukan koordinat *region of interest* dari kamera. Pada baris ke-03 adalah inisialisasi num dari frame gambar pada kamera. Pada baris ke-04 adalah fungsi *looping*. Pada baris ke-05 adalah fungsi untuk mengaktifkan frame kamera. Pada baris ke-06 adalah dungsi untuk mengatur ukuran frame yakni 700px. Pada baris ke-07 adalah fungsi untuk mengubah posisi kamera agar tidak seperti cermin. Pada baris ke-08 adalah fungsi untuk menggandakan frame. Pada baris ke-09 adalah fungsu untuk mengatur tinggi dan lebar pada frame. Pada baris ke-10 adalah fungsi untuk mendapatkan koordinat *region of interest.*

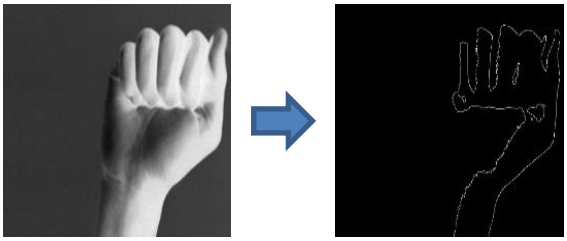
4.2.2. *Input* Data

Tahap *preprocessing* pada data uji memiliki langkah atau tahapan yang sama dengan *preprocesing* data pengenanalan. Yakni dengan merubah data dengan citra RGB menjadi data dengan citra *grayscale.* Perubahan data dilakukan dengan menjalankan *code* bahasa pemrograman *python* dan menggunakan *library cv2* yang digunakan untuk mengolah data gambar.



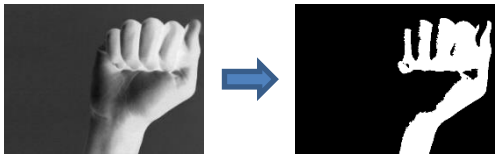
Gambar 5. Data Uji *Grayscaling*

Tahap selanjutnya adalah *edge detection* yaitu mencari tepian dari data hasil dari grayscaling. Adapun pada tahap *edge detection* ini dilakukan dengan cara menjalankan *code* bahasa pemrograman *python* dan menggunakan *library cv2* yang digunakan untuk mengolah data gambar.



Gambar 6. Data Uji *Edge Detection*

Setelah dilakukan proses *edge detction* tahap selanjutnya adalah merubah data *grayscale* menjadi data *threshold* yang tidak memiliki warna keabuan dan hanya memiliki 2 warna yakni warna hitam dan warna putih. Thresholding dilakukan dengan menjalankan code bahasa pemrograman *python* dan menggunakan *library cv2* yang digunakan untuk mengolah data gambar.



Gambar 7. Data Citra *Threshold*

4.2.3. Segmentasi Data Uji

Segmentasi data uji pada penelitian ini sama dengan proses segmentasi yang dilakukan pada data pengenalan. Berikut adalah *source code* untuk proses segmentasi citra menjadi data angka.

Segmen Program 11 Proses Segmentasi

```
01 :csv=numpy.savetxt(('data_u
   ji/data_raw/data_uji.csv'),
   threshold, delimiter=',')
```

Dari *source code* tersebut akan dijelaskan pada baris ke-01 berisi mengenai *code library Numpy* dari *python* yang berfungsi untuk merubah data citra hasil dari proses *thresholding* menjadi data angka

4.2.4. Eksraksi Data Uji

Tahap ekstraksi data uji pada penelitian ini sama dengan proses ekstraksi yang dilakukan pada data pengenalan. Tahap ekstraksi dilakukan setelah mendapatkan hasil dari tahap segmentasi yang berupa data csv adalah tahap ekstraksi data yang dilakukan untuk menyederhanakan banyaknya nilai data segmentasi. Adapun cara yang digunakan adalah dengan mencari mean dari data segmentasi tersebut. Ekstraksi data dilakukan dengan menjalankan *code* pemrograman *python*. Hasil dari ekstraksi data adalah nilai mean dari data segmentasi yang hanya memiliki jumlah range kolom sebanyak 1 kolom dan range baris sebanyak 640 yang disajikan ke dalam file dengan format csv. Tahap selanjutnya adalah merubah nilai data ekstraksi menjadi nilai fungsi bipolar yang hanya memiliki nilai -1 dan 1.

4.3. Klasifikasi Dengan *Hebb Rule*

Setelah dilakukan proses *thresholding* pada data uji selanjutnya adalah proses klasifikasi. Proses klasifikasi dilakukan menggunakan metode perhitungan *hebb rule.* Selanjutnya adalah proses perhitungan untuk mencari nilai bobot baru pada data pengenalan dan nilai bobot baru pada data uji. Dimana nilai bobot baru awal diketahui memiliki nilai 0. Perhitungan bobot baru dilakukan dengan cara menjumlahkan nilai bobot baru dengan nilai bobot lama. Nilai bobot baru yang telah didapat selanjutnya akan disimpan dengan label bobot baru. Selanjutnya adalah proses mencari nilai bias. Dimana nilai bias tersebut didapat dari perkalian data bipolar hasil *threshold* data pengenalan dikalikan dengan nilai bobot baru data pengenalan dan dijumlahkan dengan nilai bobot baru data uji. Setelah diketahui hasil dari nilai bias selanjutnya adalah merubah nilai bias tersebut menjadi fungsi bipolar yang selanjutnya akan disimpan dengan label nilai bias.

4.3.1. Perhitungan Akurasi Pengujian Dan Nilai MSE

Perhitungan akurasi pada penelitian ini dilakukan dengan mencari persamaan dari hasil perhitungan nilai bias dengan nilai bipolar hasil *thresholding* data uji. Selain berdasarkan data training pengujian juga dilakukan dengan pengambilan data uji dengan cara *capture.* Selanjutnya adalah perhitungan nilai *error* dari hasil pengujian dengan menggunakan metode MSE. Berikut merupakan rata-rata hasil pengujian yang disajikan pada tabel 1 rata-rata hasil pengujian.

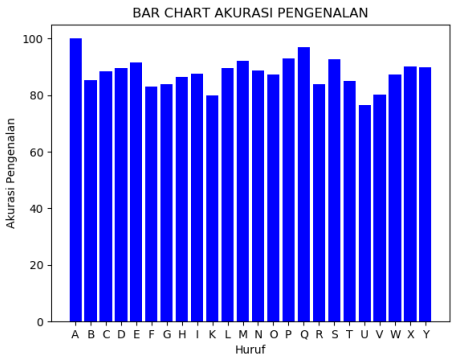
Tabel 1 Rata-rata Hasil Pengujian

	Pengujian ke-1	**Pengujian ke-2**	**Pengujian ke-3**	**Total rata-rata**
Nilai Akurasi	81.26 %	79.82 %	80.03 %	80.37 %
Nilai MSE	0.75 %	0.81 %	0.80 %	0.79 %

Pada tabel 1 rata-rata hasil pengujian didapat kan nilai akurasi pada pengujian ke-1 adalah 81.26%, pengujian ke-2 adalah 79.82%, dan pengujian ke-3 adalah 80.03%. Dan dari semua pengujian didapatkan rata-rata hasil akurasi pengujian adalah sebesar 80.37 %. Sedangkan untuk nilai MSE pada pengujian ke-1 adalah 0.75%, pengujian ke-2 adalah 0.81%, dan pengujian ke-3 adalah 0.80%. Dan dari semua pengujian didapatkan rata-rata hasil nilai MSE adalah sebesar 0.79%.

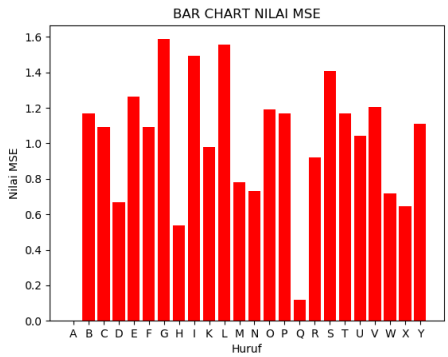
4.3.2. Visualisasi

Selanjutnya adalah proses visualisasi dari hasil perhitungan akurasi pengujian dan hasil nilai MSE yang disajikan dalam bentuk grafik. Berikut adalah bentuk visual grafik hasil perhitungan akurasi.



Gambar 8. Grafik Akurasi Pengenalan

Pada gambar 7 grafik akurasi pengenalan adalah grafik nilai akurasi dari sample data uji huruf A yang memiliki nilai *barchart* tertinggi. Adapun berikut merupakan visualisasi dari nilai MSE.



Gambar 9. Grafik Nilai *MSE*

Pada gambar 8 grafik nilai MSE adalah grafik nilai MSE dari dalah satu data uji huruf A dengan nilai terendah yang ditunjukkan pada *barchart*.

5. PENUTUP

5.1. Kesimpulan

Berdasarkan dari hasil penelitian yang telah dilakukan, maka dapat diperoleh kesimpulan bahwa hasil pengenalan pola menggunakan metode *hebb rule* pada data pengujian yang dilakukan sebanyak 3 kali di setiap hurufnya dengan data hasil *capture* memiliki rata-rata presentase akurasi pengenalan sebesar 80,37%.

5.2. Saran

Penelitian ini tentu tidak lepas dari kekurangan baik dari segi analisis maupun data yang digunakan pada penelitian ini. Maka dilangkah kedepan disarankan agar dilakukannya perbaikan untuk penelitian selanjutnya. Untuk penelitian selanjutnya disarankan memperbanyak data *training* dengan berbagai motif gambar hasil *capture* yang digunakan agar akurasi kecocokan data uji pada data pengenalan bisa memiliki presentase akurasi yang lebih maksimal.

DAFTAR PUSTAKA

Andono, P. N., Sutojo, T., & Muljono. (2017). *PENGOLAHAN CITRA DIGITAL*. Yogyakarta: ANDI (Anggota IKAPI).

Budi, A., Suma'inna, & Maulana, H. (2016). Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA). *JURNAL TEKNIK INFORMATIKA VOL 9 NO. 2*, 1-9.

Delsavonita, & Candra, F. (2018). SISTEM PENGENALAN POLA KARAKTER HURUF KOREA MENGGUNAKAN METODE PRINCIPAL COMPONENT ANALYSIS DAN JARINGAN SYARAF TIRUAN - BACK PROPAGATION. *Jom FTEKNIK Volume 5 Edisi 2 Juli s/d Desember 2018*, 1-8.

Fadhilla, M., Saf, M. R., & Sahid, D. S. (2017). Pengenalan Kepribadian Seseorang Berdasarkan Pola Tulisa Tangan Menggunakan Jaringan Syaraf Tiruan. *JNTETI, Vol 6, No.3*, 1-9.

Faridh, H. M. (2013). Pengenalan Karakter Huruf Tulisan Tangan Menggunakan Metode Principle Component Analysis. 6.

Fathia, S. (2013). Dampak Reduksi Sample Menggunakan Principle Component Analysis (PCA) Pada Pelatihan Jaringan Syaraf Tiruan Terawasi (Studi Kasus : Pengenalan Angka Tulisan Tangan). 9.

Fatta, H. A. (2009). *Rekayasa Sistem Pengenalan Wajah.* Yogyakarta: C.V Andi Offset.

Hermawan, A. (2006). Jaringan Syaraf Tiruan.

Kong, X., Hu, C., & Duan, Z. (2017). *Principal Component Analysis Network And Algorithms.* Beijing: Science Press.

Muliono, R., & Lubis, J. H. (2018). Jaringan Syaraf Tiruan Pengenalan pola Huruf Dengan Jaringan Hebb. *Jurnal Teknik Informatika Kaputama (JTIK) Vol 2, No. 1*, 1-5.

Muliono, R., & Lubis, J. H. (2018). Jaringan Syaraf Tiruan Pengenalan Pola Huruf Dengan Jaringan Hebb. *Jurnal Teknik Informatika Kaputama (JTIK) Vol. 2, No. 1*, 1-5.

Mulyana, T. M. (2015). SEGMENTASI CITRA MENGGUNAKAN HEBB-RULE. *JURNAL TEKNOLOGI INFORMASI, VOLUME 11, NOMOR 1*, 1-9.

Algoritma Principal Component Analysis. *11*, 11.

Tjolleng, A. (2017). *Pengantar Pemrograman Matlab.* Jakarta: PT. Elex Media Komputindo.