

# Pengenalan Simbol Jarimatika Menggunakan Orientasi Histogram dan Multi-layer Perceptron

Andi Sunyoto\*<sup>1</sup>, Agus Harjoko<sup>2</sup>

<sup>1</sup>Magister Teknik Informatika STMIK AMIKOM Yogyakarta

<sup>2</sup>Dosen Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM Yogyakarta

E-mail: \*<sup>1</sup>[andi@amikom.ac.id](mailto:andi@amikom.ac.id), <sup>2</sup>[aharjoko@ugm.ac.id](mailto:aharjoko@ugm.ac.id)

## Abstrak

Makalah ini membahas tentang pengenalan simbol-simbol Jarimatika menggunakan Jaringan Syaraf Tiruan (JST). Hasil penelitian ini dapat digunakan untuk pengembangan aplikasi perhitungan Jarimatika dan interaksi antara manusia dan komputer yang lebih natural. Segmentasi yang digunakan adalah orientasi histogram, algoritma JST yang digunakan adalah back propagation multi-layer perceptron. Layer-layer JST tersebut adalah satu layer input, satu hidden layer dan satu output layer. Penelitian ini bertujuan untuk implementasi pengenalan pola simbol Jarimatika menggunakan JST multi-layer perceptron, implementasi harus mampu menghasilkan klasifikasi dengan benar, sistem harus mampu melakukan klasifikasi dari gambar statis, sehingga dapat menganalisa pengenalan gestur tangan dari simbol-simbol Jarimatika. Penelitian ini menggunakan 18 simbol dasar Jarimatika. Total citra yang digunakan adalah 360 yang terbagi atas 270 citra untuk training dan 90 citra untuk testing. Hasil penelitian ini menunjukkan bahwa JST multi-perceptron dapat digunakan untuk pengenalan simbol Jarimatika dengan akurasi 93.33%. Jumlah neuron yang optimal pada hidden layer adalah 725. Implementasi penelitian ini menggunakan Matlab versi 7 (R2010a).

**Kata Kunci** —jarimatika, jaringan syaraf tiruan, back propagation, perceptron

## Abstract

This paper focuses on the recognition of Jarimatika symbols using Artificial Neural Network (ANN). The results of this research can be used to develop applications for the Jarimatika and to make interaction between humans and computers more natural. The Segmentation used is orientation histograms, the ANN algorithm used is back propagation multi-layer perceptron. The layers of the ANN are one input layer with 19 data, one hidden layer and one output layer. This research aims to implement Jarimatika symbols with pattern recognition and multi-layer perceptron algorithm, the implementation must be able to produce the correct classification, the system must be able to perform the classification of static images, so can analyze the recognition of hand gestures from Jarimatika symbols. This research uses 18 basic Jarimatika symbols. Total image used were 360, consisting of 270 images for training and 90 images for testing. The results of this study indicate that the multi-layer perceptron ANN can be used for recognition of Jarimatika symbols with accuracy 93.33%. The optimal number of neurons in the hidden layer is 725. Implementation of this research using Matlab version 7 (R2010a).

**Keywords**—jarimatika, neural network, back propagation, perceptron

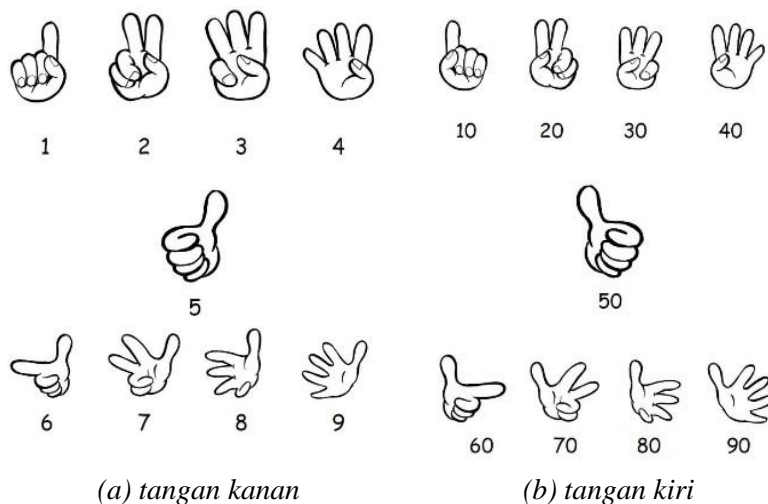
---

## 1. PENDAHULUAN

Manusia biasanya menggunakan gerakan tangan untuk mengekspresikan dan memberitahukan perasaan mereka, sehingga interpretasi visual gestur tangan dapat membantu mencapai kemudahan, termasuk penyelesaian perhitungan teknik menghitung Jarimatika. Nilai-nilai yang akan dihitung dalam Jarimatika disimbolkan menggunakan gestur-gestur tangan. Hasil penelitian ini merupakan proses awal yang dapat digunakan untuk membangun aplikasi perhitungan metode Jarimatika berbasis komputer visi. Gestur didefinisikan sebagai gerakan fisik dari tangan, lengan, atau badan yang memberikan pesan ekspresif [1], dan sistem pengenalan gestur digunakan untuk menterjemahkan dan menjelaskan gerakan sebagai perintah yang berarti [1] [2].

Teknik yang berbeda telah diterapkan untuk menangani pengenalan gestur. Teknik-teknik tersebut seperti hidden Markov Model (HMM): [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] dan Finite State Machine (FSM): [29] [30] dengan pendekatan berdasarkan metode perangkat lunak komputasi seperti Fuzzy Clustering: [31], Genetik Algorithm System (GAS): [32] dan Artificial Neural Network (ANN): [33] [34] [7] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63]. Penelitian tentang gestur tangan dan pengenalannya masih terbuka [64], karena tangan manusia adalah obyek yang kompleks dengan banyak sendi dan hubungan antar komponen pembentuk tangan.

Jarimatika adalah cara berhitung matematika dengan menggunakan kombinasi jari-jari tangan. Operasi berhitung matematika tersebut yaitu penjumlahan, pengurangan, perkalian dan pembagian. Metode ini dikembangkan oleh Septi Peni Wulandani sekitar tahun 2004 [65]. Logika penghitungan Jarimatika mirip dengan logika penggunaan sempoa [65]. Pada penelitian [66], Jarimatika dapat meningkatkan prestasi belajar matematika khususnya berhitung. Dalam jarimatika, jari tangan kanan digunakan untuk melambangkan satuan, sedangkan jari tangan kiri digunakan untuk melambangkan puluhan [65]. Gambar 1 menunjukkan 18 simbol dasar Jarimatika.



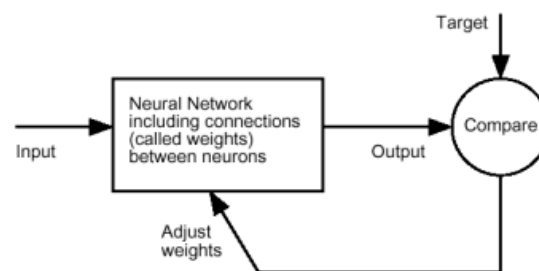
Gambar 1. Simbol Jarimatika

Penelitian ini bertujuan: (1) implementasi pengenalan pola simbol Jarimatika menggunakan JST *multi-layer perceptron*, (2) implementasi harus mampu menghasilkan klasifikasi dengan benar, (3) sistem harus mampu melakukan klasifikasi dari gambar statis, sehingga dapat menganalisa pengenalan gestur tangan dari simbol-simbol Jarimatika. Metode pengenalan yang digunakan adalah JST dengan algoritma *back propagation multi-layer perceptron* yang terdiri dari satu layer input dengan 19 inputan, satu hidden layer, dan satu layer

output, sedangkan ekstraksi ciri menggunakan orientasi histogram. Implementasi penelitian ini menggunakan Matlab versi 7 (R2010a).

### 1.1. Jaringan Syaraf Tiruan

JST terdiri dari elemen-elemen sederhana yang beroperasi secara paralel [53]. Elemen-elemen tersebut terinspirasi dari sistem syaraf biologi. Secara alami, fungsi jaringan ditentukan oleh koneksi antar elemen. JST dapat dilatih untuk mengerjakan fungsi-fungsi tertentu dengan menyesuaikan nilai koneksi (bobot) antar elemen [53]. Pada umumnya JST dilatih hingga input tertentu menghasilkan output tertentu. Jaringan disesuaikan berdasarkan perbandingan antara output dan target, sampai output jaringan sesuai dengan target [53] dengan melakukan proses belajar melalui perubahan bobot sinapsisnya seperti ilustrasi pada Gambar 2. Pada penelitian [67] JST banyak digunakan untuk aplikasi pengenalan gestur tangan. JST sudah terbukti efisien untuk proses klasifikasi, telah banyak digunakan dan hasilnya memuaskan dengan tingkat pengenalan bisa mencapai 95% [56].



Gambar 2. Diagram jaringan syaraf [56]

Secara umum proses JST terbagi menjadi 2 bagian yaitu training dan testing, yaitu training yang merupakan proses pembelajaran dari sistem jaringan syaraf yang mengatur nilai input serta bagaimana pemetaannya pada output sampai diperoleh model yang sesuai dan testing yang merupakan proses pengujian ketelitian dari model yang sudah diperoleh dari proses training [68]. Ada beberapa keuntungan penggunaan JST pada penelitiannya [56].

1. *Pattern recognition*: adalah teknik yang *powerfull* untuk memanfaatkan informasi dalam data dan generalisasi. JST untuk mempelajari dan mengenali pola-pola yang ada dalam kumpulan data.
2. Sistem dikembangkan dengan belajar, tidak harus dituliskan baris program. Dengan penulisan programming akan memakan waktu dan membutuhkan analisa yang tepat dari model. JST belajar sendiri dengan data yang ada untuk menentukan pola dan tidak diperlukan analisis.
3. JST fleksibel dalam perubahan lingkungan. Aturan dalam sistem atau aturan yang ditulis dalam program terbatas pada kondisi saat mereka dirancang. Ketika kondisi berubah mereka tidak lagi valid. Meskipun JST dapat mempelajari perubahan yang drastis, mereka sangat baik beradaptasi pada perubahan yang berkesinambungan.
4. JST dapat membangun model informasi ketika pendekatan yang konvensional gagal. Karena JST dapat menangani interaksi yang sangat kompleks yang mereka dapat dengan yang mudah memodelkan data yang sulit dengan pendekatan tradisional seperti statistik inferensial atau logika pemrograman.
5. Kinerja JST setidaknya sama baiknya dengan pemodelan statistik klasik, dan lebih baik pada beberapa permasalahan. JST membangun model yang lebih mencerminkan struktur data dalam waktu yang singkat.

Meskipun JST memiliki bermacam keunggulan, tetapi setiap sistem juga mempunyai keterbatasan. Berikut adalah beberapa batasannya [56]:

1. JST tidak mampu menjelaskan model atau jaringan yang telah dibangun. JST selalu mendapatkan hasil yang lebih baik tetapi sulit dan memerlukan waktu lama bagaimana menerangkan cara mendapatkannya. Penjelasan ini penting terutama untuk analisis yang ingin tahu perilaku model tersebut.
2. JST tidak akan menghasilkan hasil yang baik jika data masukan tidak mewakili masalah. Situasi ini diklasifikasikan sebagai “input sampah” menghasilkan “produk sampah”. Sehingga analisis menghabiskan waktu untuk memahami masalah atau hasil yang diharapkan. Dan analisis harus memilih data yang sesuai digunakan untuk melatih sistem dan diukur dengan cara yang mencerminkan perilaku faktor.
3. JST membutuhkan waktu untuk melatih model ketika set data yang kompleks. Teknik ini akan memperlambat pada komputer yang memiliki spesifikasi rendah atau komputer tanpa fasilitas perhitungan matematik khusus. Tapi secara keseluruhan, itu masih lebih cepat dari pada pendekatan analisis data yang lainnya. Lagi pula sekarang bukanlah masalah besar, karena saat ini prosesor kebanyakan komputer cukup cepat untuk training JST.

#### 1.1.1. Teknologi Pengenalan Gestur Tangan

Pengenalan gestur adalah tidak hanya berisi gerakan manusia, tapi juga interpretasi gerakan perintah semantik yang bermakna [2]. Dalam membangun sistem apapun, langkah pertama adalah mengumpulkan data yang diperlukan untuk menyelesaikan tugas tertentu. Untuk gestur tangan dan sistem pengenalannya, ada berbeda teknologi yang digunakan untuk memperoleh input data. Teknologi yang digunakan untuk mengenali gestur, [69] membagi menjadi tiga pendekatan yang digunakan untuk menterjemahkan gestur, yaitu: berbasis visi, sarung tangan *instrumented* (data), dan penanda warna. Penelitian ini menggunakan pendekatan berbasis visi.

#### 1.1.2. Teknik Pengenalan Gestur

Pengenalan gestur didalamnya terdiri dari beberapa konsep seperti pengenalan pola (*pattern recognition*)[70], deteksi pergerakan (*motion detection*) dan analisis [70], dan *machine learning*[70]. Alat dan teknik yang berbeda digunakan dalam sistem pengenalan gestur, seperti komputer visi (*computer vision*)[71][72], pengolahan citra [1], pengenalan pola [1], model statistik [1].

## 2. METODE PENELITIAN

### 2.1. Database Feature Jarimatika

#### 2.1.1. Setting Data Citra Training dan Testing

Langkah pertama yang dilakukan pada penelitian ini adalah membuat database *featur* simbol Jarimatika dari data berupa citra. Citra tersebut akan digunakan untuk tahap training dan testing JST. Citra yang digunakan pada penelitian ini diambil dari webcam yang terpasang di laptop dalam RGB format file .png.

Jumlah citra yang digunakan adalah mewakili seluruh simbol Jarimatika (18 simbol). Setiap simbol terdiri dari 20 citra (15 citra training, 5 citra testing), sehingga citra training adalah  $15 \times 18 = 270$ , dan jumlah citra testing adalah  $5 \times 18 = 90$ . Sehingga total citra yang digunakan adalah 360. Contoh citra untuk training dan testing dapat dilihat pada Gambar 3.

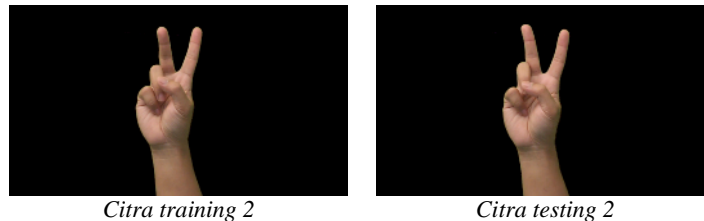
### 2.1.2. Ekstraksi feature

Sebuah gesture harus sama tanpa memperhatikan dari mana terbentuk. Untuk mencapai hal tersebut kita akan mengabaikan posisi sama sekali, dan mentabulasi sebuah histogram seberapa sering orientasi elemen dalam masing-masing citra. Dalam prakteknya, seseorang dapat memilih sebuah set data training dan data testing dengan subtansi orientasi histogram yang berbeda untuk masing-masing.

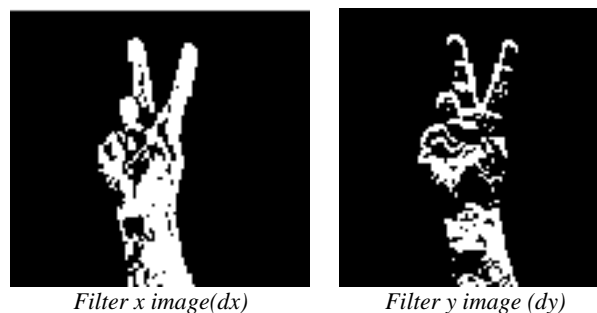
Gradient vektor dengan orientasi tepi adalah sebagai feature yang akan disajikan penulis ke JST. Alasan untuk ini bahwa deteksi tepi sudah cukup baik dan memungkinkan untuk menguji jaringan dengan gambar dari database yang berbeda. Citra telah diproses akan menghasilkan keluaran sebuah vektor berisi sebuah elemen sama dengan jumlah *bins* dari orientasi histogram. Berikut langkah-langkah ekstraksi citra menjadi *feature vector*:

1. Membaca database citra RGB, kemudian dikonversi ke bentuk grayscale.
2. Merubah ukuran citra menjadi 96x170 piksel. Ukuran ini sudah cukup untuk mendapatkan detail citra dan mempercepat proses pengolahan citra.
3. Langkah selanjutnya deteksi tepi menggunakan 2 filter masing-masing untuk arah 'x' (horizontal)  $x=[0 \ -1 \ 1]$  dan untuk arah 'y' (vertikal) Untuk arah y,  $y=\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$ . Hasil langkah ini seperti dilihat pada Gambar 4.
4. Hasil elemen matrik dy dibagi dengan elemen matrik dx. Setelah dibagi, ambil *arctangent* untuk memperoleh orientasi gradien. Hasil orientasi gradien dapat dilihat pada Gambar 5.
5. Kemudian mengkonversi nilai radian kolom matrik menjadi nilai derajat untuk menscan vektor untuk nilai 0° sampai 90°. Set *bin* histogram menjadi 19, dengan masing-masing bin diset 10°. Nilai tersebut yang akan dijadikan *feature vector* tiap citra.
6. Nilai *feature vector* pada langkah dinormalisasi dengan *formula (1)*. Kemudian simpan ke dalam file teks.

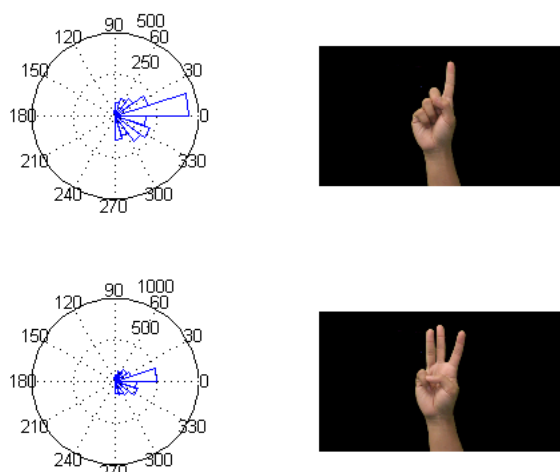
$$P(i, j) = \frac{((0.9-0.1)(D(i, j) - \min(D(i)))}{\max(D(i)) - \min(D(i))} + 0.1 \quad (1)$$



Gambar 3. Contoh citra untuk training dan testing



Gambar 4. Hasil filter x-y



Gambar 5. Orientasi histogram simbol ‘1’ dan ‘3’

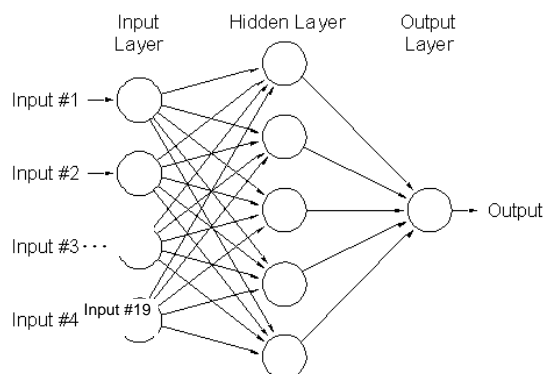
## 2.2. Proses Training

### 2.2.1. Algoritma Learning Jaringan

Prosedur training dalam JST adalah melatih jaringan untuk melakukan beberapa tugas tertentu dengan mengganti nilai bobot dan bias sebuah jaringan. Algoritma yang digunakan adalah *supervised learning*. Dalam *supervised learning* aturan disediakan dengan satu set contoh (*training set*) dari perilaku jaringan yang tepat: dimana sebuah input akan dimasukkan ke dalam jaringan, dan akan dibandingkan apakah output sesuai dengan target [61].

Algoritma training JST yang digunakan adalah *back propagation* dengan *multi-layer perceptron* dimana layer terdiri dari satu *input layer*, satu *hidden layer*, dan *output layer* seperti pada Gambar 6. *Back propagation* mengubah nilai-nilai bobot dalam jaringan dengan arah mundur. JST *Back propagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan dalam mengenali pola yang digunakan selama training serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola input yang serupa (tetapi tidak sama) dengan pola yang dipakai selama training [68]. Menurut [73] langkah-langkah dalam algoritma *back propagation* adalah:

1. Pilih data training dan masukkan vektor input ke dalam jaringan
2. Hitung output jaringan
3. Bandingkan output dengan target yang diinginkan dan cari kesalahannya
4. Modifikasi bobot sehingga dapat mengurangi error.

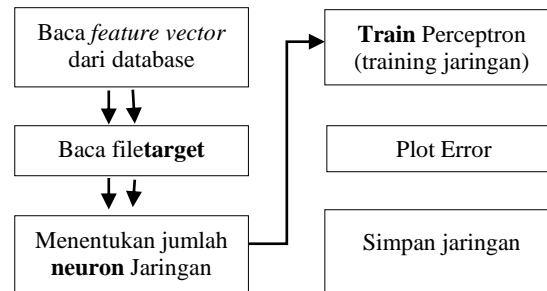


Gambar 6. Jaringan *Back propagation*

Kesalahan pada keluaran dari jaringan merupakan selisih antara keluaran aktual (*current output*) dan keluaran target (*desired output*). Kesalahan makalah ini dihitung dengan SSE (*Sum Squared Error*) yang merupakan hasil penjumlahan nilai kuadrat error neuron-1 dan neuron-2 pada lapisan output tiap data [74].

### 2.2.2. Training Perceptron

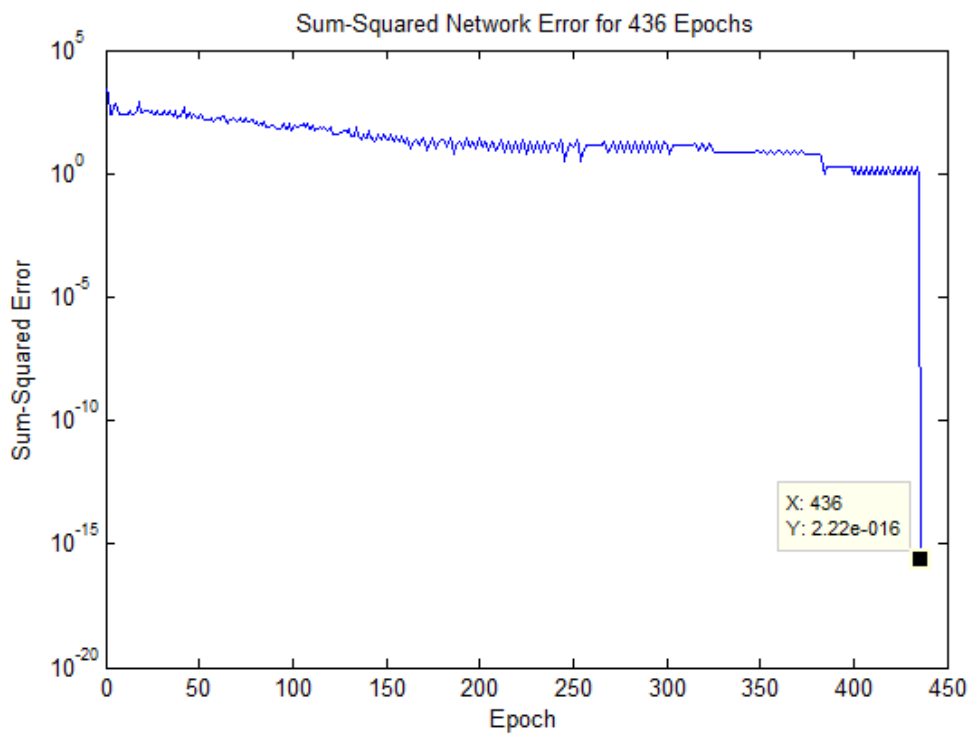
Perceptron adalah salah satu konsep '*learn*' (pembelajaran) pada JST. Pada langkah selanjutnya berdasarkan aturan pembelajaran perceptron untuk melatih jaringan untuk pengenalan pola (*pattern recognition*). Pada Gambar 7 ditunjukkan flowchart untuk program perceptron.



Gambar 7. Flowchart learn Perceptron

Proses Perceptron berdasarkan flowchart di atas adalah:

1. Baca file dari database *feature vector*. File teks (train.txt) dihasilkan dari proses ekstraksi *feature vector*. Setiap citra terdiri dari 19 kolom, total citra untuk training 270. Jadi dimensi matrik input untuk training adalah 19x270.
2. Membaca file target (targetTrain.txt). Berisi nilai target vektor yang dibangun user sesuai dengan jumlah output yang dihasilkan.
3. Menentukan jumlah neuron pada jaringan. Proses ini sifatnya coba-coba sampai mendapatkan jumlah neuron yang optimal.
4. Train Perceptron. Pada tahap ini, data yang sebelumnya diproses dimasukkan dalam layer Perceptron. Layer ini berisi sejumlah neuron yang ditentukan pada langkah ke 3. Nilai *feature vector* akan dimasukkan ke dalam jaringan kemudian dikalikan dengan neuron (bobot). Output akan dibandingkan dengan vektor target. Setelah itu, jika ada error, Perceptron jaringan akan menyesuaikan nilai bobot (*weight*) sampai tidak ada error atau meminimalkan error dan setelah itu akan berhenti. Setiap melewati vektor input disebut epochs.
5. Gambar 8 menunjukkan grafik plotting kesalahan perceptron proses training dan jumlah epochs. Hasil proses training menghasilkan SSE (*Sum-Squared Error*)=0.
6. Simpan jaringan yang paling optimal dalam file 'bobot.mat'.

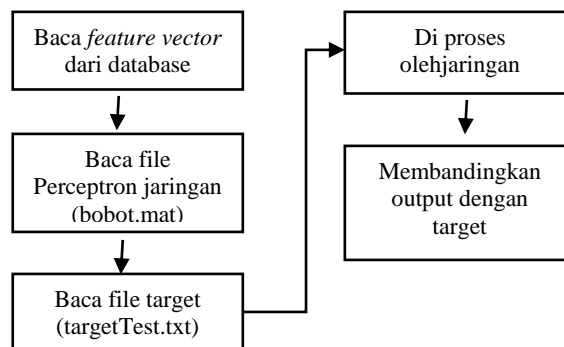


Gambar 8. Grafik Plotting Error Perceptron

### 2.3. Proses Testing

Setelah melakukan proses training dan ditemukan jaringan yang optimal, langkah berikutnya adalah proses testing. Proses ini digunakan untuk mentesting apakah jaringan yang dibangun dengan data lain. Setiap simbol diwakili oleh 5 citra, sehingga jumlah seluruh citra testing adalah  $9 \times 18 = 90$ . Proses testing Perceptron dapat dilihat pada flowchart Gambar 9.

1. Baca *feature vector* dari database (test.txt). Kemudian memasukkan vektor tersebut ke dalam jaringan yang telah ditraining.
2. Baca file perceptron jaringan (bobot.mat). File ini menyimpan jaringan yang paling optimal pada saat proses training.
3. Baca file target (targetTest.txt). File ini berisi matrik target citra yang akan ditesting.
4. Proses dengan Perceptron Network. Sekarang *feature vector* dari citra testing dimasukkan ke dalam jaringan. Nilai *feature vector* akan melewati semua bobot (*neurons*) dalam Perceptron jaringan dan akan menghasilkan output.
5. Output yang dihasilkan akan di bandingkan dengan target yang telah ditentukan. Proses ini akan mengeluarkan citra testing mana yang terklasifikasi dan mana yang tidak.



Gambar 9. Flowchart Testing Jaringan Perceptron



### 3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan 360 citra yang dibagi menjadi 270 training dan 90 citra testing. Semua proses training yang dilakukan menghasilkan nilai SSE=0. Dengan melakukan uji coba nilai untuk menentukan nilai neuron yang optimal, dihasilkan bahwa jumlah neuron pada hidden layer yang paling optimal adalah 725 dengan prosentase akurasi 93.33%.

Tabel 1. Jumlah neuron dan prosentase akurasi

Neuron	Iterasi	Max Epochs	Sum Clasified	% Clasified
722	617	10000	75	83.33%
723	1739	10000	74	82.22%
724	577	10000	72	80%
725	527	10000	84	93.33%
726	845	10000	70	77.78%
727	310	10000	75	83.33%
728	886	10000	71	78.89%
729	675	10000	82	91.11%
730	389	10000	73	81.11%
731	470	10000	72	80%

Tabel 2. Perbandingan klasifikasi tiap simbol

Simbol	Clasified/ not Clasified	Simbol	Clasified/ Not Clasified
Simbol 1	5/5	Simbol 10	5/5
Simbol 2	3/5	Simbol 20	3/5
Simbol 3	5/5	Simbol 30	5/5
Simbol 4	5/5	Simbol 40	5/5
Simbol 5	5/5	Simbol 50	5/5
Simbol 6	5/5	Simbol 60	4/5
Simbol 7	5/5	Simbol 70	4/5
Simbol 8	5/5	Simbol 80	5/5
Simbol 9	5/5	Simbol 90	5/5

Seluruh simbol dilakukan proses testing, dan hasil perbandingan jumlah data testing dan jumlah yang terklasifikasi dapat dilihat pada Tabel 2. Format nilai pada Tabel 2 adalah “citra testing/citra terklasifikasi”. Contoh jika tertulis 5/5, artinya jumlah citra terklasifikasi 5 dari 5 citra testing, 3/5 berarti citra terklasifikasi 3 dari 5 citra testing. Dari Tabel 2 dapat dilihat bahwa 14 simbol Jarimatika dapat terklasifikasi sempurna.

#### 4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan pengenalan pola menggunakan JST *multi-layer perceptron* dengan algoritma *back propagation* dengan gambar static sebagai inputan. Akurasi pada penelitian ini dapat dioptimalkan dengan melakukan uji coba nilai jumlah neuron sampai ditemukan perceptron jaringan yang optimal. Jumlah neuron yang optimal pada penelitian ini adalah 725 dan berhenti pada iterasi 527.

JST dapat digunakan untuk pengenalan simbol jarimatika dengan akurasi 93.33%, dan 14 dari 18 simbol Jarimatika yang ada dapat dikenali sempurna. Penelitian berikutnya berfokus pada bagaimana membangun aplikasi pengenalan simbol jarimatika menggunakan JST *multi-layer perceptron*.

#### 5. SARAN

Penambahan akurasi pada pengenalan dapat dicoba oleh peneliti-peneliti berikutnya dengan melakukan proses training dan testing untuk citra yang mengalami transformasi dan menggunakan metode pengenalan yang lain seperti *Hidden Markov Model (HMM)*, *Finite State Machine (FSM)* atau *Genetic Algoritm*.

#### DAFTAR PUSTAKA

- [1] Mitra, S., Acharya, T., 2007, Gesture Recognition: A Survey, *IEEE Transaction on System, Man, and Cybernetics Part C: Applications and Reviews*, Vol 37, No 3, hal 311-324.
  - [2] Murthy, G. R. S., Jadon, R.S., 2009, A Review of Vision Based Hand Gestures Recognition, *International Journal of Information Technology and Knowledge Management*, Vol 2, No 2, hal 405-410.
  - [3] Rabiner, L., 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*.
  - [4] Yamato, J., Ohya, J., Ishii, K., 1992, Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR '92*, Champaign, 15-18 Juni 1992.
  - [5] Samaria, F., Young, S., 1994, HMM-Based Architecture for Face Identification. *Image and Computer Vision*, Vol 12, No 8, hal 537-543.
  - [6] Starner, T., Pentland, A., 1997, Real-Time American Sign Language Recognition from Video Using Hidden Markov Models, *Motion-Based Recognition*, Vol 9, hal 227-243.
  - [7] Rigoll, G., Kosmala, A., Schuster, M., 1997, High Performance Gesture Recognition Using Probabilistic Neural Networks and Hidden Markov Models, *Time-Varying Image Processing and Moving Object Recognition 4*, hal 233-287.
  - [8] Marcel, S., Bernier, O., Viallet, J. E., Collobert, D., 2000, Hand Gesture Recognition Using Input-Output Hidden Markov Models, *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, 28-30 Maret 2000.
  - [9] Fahn, C. S., Chu, K. Y., 2011, *Hidden-Markov-Model-Based Hand Gesture Recognition Techniques Used for a Human-Robot Interaction System*, *Human-Computer Interaction: Interaction Techniques and Environments*, Springer-Verlag, Berlin Heidelberg.
  - [10] Binh, N. D., Ejima, T., 2006, Real-Time Hand Gesture Recognition Using Pseudo 3-D Hidden Markov Model, *Fifth IEEE International Conference on Cognitive Informatics (ICCI 2006)*, Beijing, 17-19 Juli 2006.
-

- 
- [11] Chang, C. C., Pengwu, C.M., 2004, Gesture recognition approach for sign language using curvature scale space and hidden Markov model, *IEEE International Conference on Multimedia and Expo (ICME '04)*, Taipei, 30 Juli 2004.
- [12] Dennemont, Y., Bouyer, G., Otmame, S., Malle, M., 2012, A Discrete Hidden Markov Models Recognition Module for Temporal Series: Application to Real-Time 3D Hand Gestures, *3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Istanbul, 15-18 Oktober 2012.
- [13] Elmezain, M., Al-Hamadi, A., Krell, G., El-Etriby, S., Michaelis, B., 2007, Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models, *IEEE International Symposium on Signal Processing and Information Technology*, Giza, 15-18 Desember 2007.
- [14] Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B., 2008, A Hidden Markov Model-based Continuous Gesture Recognition System for Hand Motion Trajectory, *19th International Conference on Pattern Recognition (ICPR 2008)*, Tampa, 8-11 Desember 2008.
- [15] Falinie, Y., Gaus, A., Wong, F., 2012, Hidden Markov Model-Based Gesture Recognition with Overlapping Hand-Head/Hand-Hand Estimated Using Kalman Filter, *Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Kota Kinabalu, 8-10 Februari 2012.
- [16] Irteza, K. M., Ahsan, S. M. M., Deb, R. C., 2012, Recognition of Hand Gesture Using Hidden Markov Model, *15th International Conference on Computer and Information Technology (ICCIT)*, Chittagong, 22-24 Desember 2012.
- [17] Morguet, P., Lang, M., 1998, Spotting Dynamic Hand Gestures in Video Image Sequences Using Hidden Markov Models, *International Conference on Image Processing (ICIP '98)*, Chicago, 4-7 Oktober 1998.
- [18] Rao, J., Gao, T., Gong, Z., Jiang, Z., 2009, Low Cost Hand Gesture Learning and Recognition System Based on Hidden Markov Model, *Second International Symposium on Information Science and Engineering (ISISE)*, Shanghai, 26-28 Desember 2009.
- [19] Shrivastava, R., 2013, A Hidden Markov Model Based Dynamic Hand Gesture Recognition System Using OpenCV, *IEEE 3rd International Advance Computing Conference (IACC)*, Ghaziabad, 22-23 Februari 2013.
- [20] Ulas, A., Yıldız, O. T., 2009, An Incremental Model Selection Algorithm Based on Cross-Validation for Finding the Architecture of a Hidden Markov Model on Hand Gesture Data Sets, *Eight International Conference on Machine Learning and Applications (ICMLA '09)*, Miami Beach, 13-15 Desember 2009.
- [21] Hieu, D. V., Nitsuwat, S., 2008, Image Preprocessing and Trajectory Feature Extraction based on Hidden Markov Models for Sign Language Recognition, *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '08)*, Phuket, 6-8 Agustus 2008.
- [22] Wan, J., Ruan, Q., An, G., Li, W., 2012, Gesture recognition based on Hidden Markov Model from sparse representative observations, *IEEE 11th International Conference on Signal Processing (ICSP)*, Vol 2, Beijing, 22-25 Oktober 2012.
- [23] Yang, R., Sarkar, S., 2006, Gesture Recognition using Hidden Markov Models from Fragmented Observations, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol 1, New York, 17-22 Juni 2006.
-

- 
- [24] Yang, Z., Li, Y., Chen, W., Zheng, Y., 2012, Dynamic Hand Gesture Recognition Using Hidden Markov Models, *7th International Conference on Computer Science Education (ICCSE)*, Melbourne, 14-17 Juli 2012.
- [25] Bilal, S., Akmeliawati, R., Shafie, A. A., Salami, M. J., 2013, Hidden Markov Model for Human to Computer Interaction: A Study on Human Hand Gesture Recognition, *Artificial Intelligence Review*, Vol 40, No 4, hal 1-22.
- [26] Elmezain, M., Al-Hamadi, A., Appenrodt, J., and Michaelis, B., 2009, A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition, *International Journal of Electrical, Computer, and Systems Engineering (IJECE)*, Vol 3, No 3, hal 156-163.
- [27] Kim, I. C., IlChien, S., 2001, Analysis of 3D Hand Trajectory Gestures Using Stroke-Based Composite Hidden Markov Models, *Applied Intelligence*, Vol 15, No 2, hal 131–143.
- [28] Wilson, A. D., Bobick, A. F., 1999, Parametric Hidden Markov Models for Gesture Recognition, *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol 21, No 9, hal 131-143.
- [29] Hong, P., Turk, M., Huang, T. S., 2000, Constructing Finite State Machines for Fast Gesture Recognition, *15th International Conference on Pattern Recognition*, Vol 3, Barcelona, 3-7 September 2000.
- [30] Verma, R., Dev, A., 2009, Vision Based Hand Gesture Recognition Using Finite State Machines and Fuzzy Logic, *International Conference on Ultra Modern Telecommunications and Workshops*, St. Petersburg, 12-14 Oktober 2009.
- [31] Li, X., 2003, *Gesture Recognition Based on Fuzzy C-Means Clustering Algorithm*, Department of Computer Science, The University of Tennessee, Knoxville.
- [32] Glodberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Boston.
- [33] Krose, B., Smagt, P. V. D., 1993, *An Introduction to Neural Networks*, University of Amsterdam, Amsterdam.
- [34] Littmann, E., Drees, A., Ritter, H., 1996, Visual Gesture Recognition by a Modular Neural System, *International Conference on Artificial Neural Networks (ICANN 96)*, Venice, 21-23 Agustus 1996.
- [35] Stergiopoulou, E., Papamarkos, N., Atsalakis, A., 2005, Hand Gesture Recognition Via a New Self-organized Neural Network, *Progress in Pattern Recognition, Image Analysis and Applications*, Springer Berlin Heidelberg.
- [36] Angelopoulou, A., Rodriguez, J. G., Psarrou, A., Gupta, G., 2010, Hand Gesture Modelling And Tracking Using A Self-Organising Network, *The 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, 18-23 Juli 2010.
- [37] Araga, Y., Shirabayashi, M., Kaida, M. K., Hikawa, H., 2012, Real Time Gesture Recognition System Using Posture Classifier And Jordan Recurrent Neural Network, *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, 10-15 Juni 2012.
- [38] Cracknell, J., Cairns, A. Y., Gregor, P., C., Ramsay, Ricketts, I. W., 1994, Gesture Recognition: An Assessment of the Performance of Recurrent Neural Networks Versus Competing Techniques, *IEE Colloquium on Applications of Neural Networks to Signal Processing*, London, 15 Desember 1994.
-

- [39] Ghosh, D. K., Ari, S., 2011, A Static Hand Gesture Recognition Algorithm Using K-Mean Based Radial Basis Function Neural Network, *8th International Conference on Information, Communications and Signal Processing (ICICS)*, Singapura, 13-16 Desember 2011.
- [40] Hagg, J., Curuklu, B., Akan, B., Asplund, L., 2008, Gesture Recognition Using Evolution Strategy Neural Network, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, Hamburg, 15-18 September 2008.
- [41] King, L. M., Nguyen, H. T., Taylor, P. B., 2005, Hands-free Head-movement Gesture Recognition using Artificial Neural Networks and the Magnified Gradient Function, *27<sup>th</sup> Annual International Conference of the Engineering in Medicine and Biology Society*, Shanghai, 17-18 Januari 2006.
- [42] Lin, D. T., 1998, Spatio-Temporal Hand Gesture Recognition Using Neural Networks, *The 1998 IEEE International Joint Conference on Neural Networks Proceedings, IEEE World Congress on Computational Intelligence*, Vol 3, Anchorage, 4-9 Mei 1998.
- [43] Maraqa, M., Zaiter, R. A., 2008, Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks, *First International Conference on the Applications of Digital Information and Web Technologies*, Ostrava, 4-6 Agustus 2008.
- [44] Murakami, K., Taguchi, H., 1991, *Gesture Recognition Using Recurrent Neural Networks*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 27 April 1991.
- [45] Murthy, G. R. S., Jadon, R. S., 2010, Hand Gesture Recognition Using Neural Networks, *2<sup>nd</sup> International Advance Computing Conference (IACC 2010)*, Patiala, 19-20 Februari 2010.
- [46] Paulraj, M. P., Yaacob, S., Desa, H., Majid, W. M. R. W. A., 2009, Gesture Recognition System for Kod Tangan Bahasa Melayu (KTBM) Using Neural Network, *5<sup>th</sup> International Colloquium on Signal Processing Its Applications (CSPA 2009)*, Kuala Lumpur, 6-8 Maret 2009.
- [47] Setiawan, H., Setyawan, I., Nugroho, S., 2013, Hand Gesture Recognition Using Optimized Neural Network Shape Fitting on Arm11, *International Conference on Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, 7-8 Oktober 2013.
- [48] Su, M. C., Jean, W. F., Chang, H. T., 1996, A Static Hand Gesture Recognition System Using A Composite Neural Network, *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, New Orleans, 8-11 September 1996.
- [49] Tumor, B., Varkonyi-Koczy, A. R., 2010, Circular Fuzzy Neural Network Based Hand Gesture and Posture Modeling, *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, Austin, 3-6 Mei 2010.
- [50] Weissmann, J., Salomon, R., 1999, Gesture Recognition for Virtual Reality Applications Using Data Gloves and Neural Networks, *International Joint Conference on Neural Networks (IJCNN '99)*, Washington, 10-16 Juli 1999.
- [51] Wysoski, S. G., Lamar, M. V., Kuroyanagi, S., Iwata, A., 2002, A Rotation Invariant Approach on Static-Gesture Recognition Using Boundary Histograms and Neural Networks, *Proceedings of the 9th International Conference on Neural Information Processing*, Vol 4, Singapura, 18-22 November 2002.
-

- [52] Xu, D., 2006, A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG, *18<sup>th</sup> International Conference on Pattern Recognition (ICPR 2006)*, Vol. 3, Hongkong, 20-24 Agustus 2006.
- [53] Yewale, S. K., Bharne, P. K., 2011, Hand Gesture Recognition Using Different Algorithms Based On Artificial Neural Network, *International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Udaipur, 22-24 April 2011.
- [54] Zhu, C., Sheng, W., 2009, Online Hand Gesture Recognition Using Neural Network Based Segmentation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, St. Louis, 10-15 Oktober 2009.
- [55] Hasan, H., Kareem, S. A., 2014, Static Hand Gesture Recognition Using Neural Networks, *Artificial Intelligence Review*, Vol 41, No 2, hal 147–181.
- [56] Maung, T. H. H., 2009, Real-Time Hand Tracking And Gesture Recognition System Using Neural Networks, *International Science Indexworld*, Vol 3, No 2, hal 393-397.
- [57] Nölker, C., Ritter, H., 2002, Visual Recognition of Continuous Hand Postures, *IEEE Transactions on Neural Networks*, Vol 13, No 4, hal 983-994.
- [58] Oz, C., Leu, M. C., 2011, American Sign Language Word Recognition with a Sensory Glove Using Artificial Neural Networks, *Engineering Applications of Artificial Intelligence*, Vol 24, No 7, hal 1204-1213.
- [59] Stergiopoulou, E., Papamarkos, N., 2009, Hand Gesture Recognition Using a Neural Network Shape Fitting Technique, *Engineering Applications of Artificial Intelligence*, Vol 22, No 8, hal 1141-1158.
- [60] Supriyati, E., Iqbal, M., 2013, Recognition System of Indonesia Sign Language based on Sensor and Artificial Neural Network, *MAKARA of Technology Series*, Vol 17 No 1, hal 25-31.
- [61] Symeonidis, K., 1996, Hand Gesture Recognition Using Neural Networks, *Neural Network*, Vol 13.
- [62] Yoshiike, N., Takefuji, Y., 2003, Object Segmentation Using Maximum Neural Networks for the Gesture Recognition System, *Neurocomputing*, Vol 51, hal 213-224.
- [63] Yewale, S. K., Bharne, P. K., 2011, Artificial Neural Network Approach for Hand Gesture Recognition, *International Journal of Engineering Science and Technology*, Vol 3 No 4.
- [64] Bilal, S., Akmeliawati, R., El Salami, M. J., Shafie, A.A., 2011, Vision-Based Hand Posture Detection and Recognition for Sign Language - A Study, *4th International Conference On Mechatronics (ICOM)*, Kuala Lumpur, 17-19 Mei 2011.
- [65] Wulandani, S. P., 2007, *Jarimatika*, Kawan Pustaka, Jakarta Selatan.
- [66] Soleh, D. H. P., Abidin, Z., Ariati, J., 2011, Pengaruh Metode Jarimatika Terhadap Prestasi Belajar Matematika Siswa Tunanetra Sekolah Dasar SLB Negeri 1 Pematang, *Jurnal Fakultas Psikologi Universitas Diponegoro*, Vol 10, No 2, hal 115-125.
- [67] Sunyoto, A., Hardjoko, A., 2014, Review Teknik, Teknologi, Metodologi dan Implementasi Pengenalan Gesture Tangan Berbasis Visi, *Seminar Nasional Aplikasi Teknologi Informasi 2014 (SNATi 2014)*, Yogyakarta, 21 Juni 2014.
- [68] Antara, I. P. R., Sumarminingsih, E., Handoyo, S., 2013, Model Jaringan Syaraf Tiruan Backpropagation dengan Input Berdasarkan Model Regresi Terbaik, *Jurnal Mahasiswa Statistik*, Vol 1, No 1, hal 9-12.
-

- 
- [69] Nayakwadi, V., Pokale, N. B., 2014, Natural Hand Gestures Recognition System for Intelligent HCI: A Survey, *International Journal of Computer Applications Technology and Research (IJCATR)*, Vol 3, No 1, hal 10-19.
- [70] Wu, Y., Huang, T. S., 1999, Vision-Based Gesture Recognition: A Review, *Urbana*, Vol 51, hal 103-115.
- [71] Wachs, J. P., Kölsch, M., Stern, M., Edan, M., 2011, Vision-Based Hand-Gesture Applications, *Communications of the ACM*, Vol 54, No 2, hal 60-71.
- [72] Samantaray, A., Nayak, S. K., Mishra, A. K., 2013, Hand Gesture Recognition using Computer Vision, *Technomanthan 2013*, Vol 4, No 6,
- [73] Lippmann, R. P., 1987, An Introduction to Computing with Neural Nets, *ASSP Magazine*, Vol 4, No 2, hal 4-22.
- [74] Kusumadewi, S., 2002, *Buku Ajar Kecerdasan Buatan*, Teknik Informatika Universitas Islam Indonesia, Yogyakarta.
-