

RANCANG BANGUN *TOOL* UNTUK JARINGAN SYARAF TIRUAN (JST) MODEL PERCEPTRON

Liza Afriyanti

Laboratorium Komputasi dan Sistem Cerdas
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia
Jl. Kaliurang Km. 14 Yogyakarta 55501
E-mail: zhazha_moet@yahoo.co.id

ABSTRAKS

Jaringan Syaraf Tiruan (JST) merupakan topik yang hangat dibicarakan dan mengundang banyak kekaguman dalam dasa warsa terakhir. Hal ini disebabkan karena kemampuan JST untuk meniru sifat sistem yang dimasukkan. Pada dasarnya JST mencoba meniru cara kerja otak makhluk hidup, yaitu bentuk neuron-nya (sel syaraf). Faktor kecerdasan dari syaraf tidak ditentukan di dalam sel tetapi terletak pada bentuk dan topologi jaringannya. Salah satu model JST yang sering digunakan untuk pembelajaran adalah perceptron. Metode perceptron merupakan metode pembelajaran dengan pengawasan dalam sistem jaringan syaraf. Dalam merancang jaringan neuron yang perlu diperhatikan adalah banyaknya spesifikasi yang akan diidentifikasi. Jaringan neuron terdiri dari sejumlah neuron dan sejumlah masukan. Penelitian dilakukan dengan membangun sebuah tool yang digunakan untuk membuat struktur Jaringan Syaraf Tiruan model perceptron. Sehingga dapat membantu user dalam memahami teori sekaligus dapat memahami struktur JST secara visual. Hasil akhir dari penelitian skripsi ini adalah untuk membangun sebuah tool untuk membuat struktur Jaringan Syaraf Tiruan model perceptron. Tool untuk struktur Jaringan Syaraf Tiruan (JST) model perceptron dapat digunakan untuk melakukan proses pelatihan dan pengujian serta menampilkan struktur jaringan perceptron.

Kata-kunci : Jaringan Syaraf Tiruan, Perceptron, Struktur Jaringan.

1. PENDAHULUAN

1.1 Latar Belakang

Jaringan Syaraf Tiruan (*Artificial Neural Networks*) adalah salah satu cabang ilmu dari bidang ilmu Kecerdasan Buatan. Salah satu model JST yang sering digunakan untuk pembelajaran adalah perceptron. Metode perceptron merupakan metode pembelajaran dengan pengawasan dalam sistem jaringan syaraf. Dalam merancang jaringan neuron yang perlu diperhatikan adalah banyaknya spesifikasi yang akan diidentifikasi. Jaringan neuron terdiri dari sejumlah neuron dan sejumlah masukan.

Salah satu permasalahan dalam JST yang dihadapi *user* adalah tidak dapat memahami struktur pada jaringannya. Hal ini terjadi karena tidak didukung dengan tersedianya suatu aplikasi yang dapat membantu *user* dalam memahami struktur JST. Untuk dapat memahami struktur JST selain memahami teori juga diperlukan pemahaman secara visual. Untuk itu diperlukan sebuah aplikasi handal yang dapat membantu *user* dalam memahami struktur JST model Perceptron.

Salah satu bahasa pemrograman yang banyak digunakan adalah Java. Java telah menerapkan konsep pemrograman berorientasi objek yang modern dalam implementasinya. Dengan kemudahannya, Java dapat digunakan dalam perancangan pembuatan *tool* untuk JST model Perceptron. Berdasarkan latar belakang tersebut, maka pada penelitian ini akan dibangun sebuah *tool*

untuk membuat Jaringan Syaraf Tiruan model perceptron.

1.2 Tujuan

Tujuan yang diharapkan dari penulisan tugas akhir ini adalah membangun aplikasi untuk mengimplementasikan struktur Jaringan Syaraf Tiruan model Perceptron dengan menggunakan bahasa pemrograman Java.

1.3 Batasan Masalah

Dalam melaksanakan suatu penelitian diperlukan adanya batasan agar tidak menyimpang dari yang telah direncanakan sehingga tujuan yang sebenarnya dapat tercapai. Batasan masalah yang diperlukan yaitu :

1. Aplikasi ini dibuat untuk dijalankan pada desktop.
2. Program yang akan dibuat nantinya akan menampilkan struktur JST model Perceptron.
3. Masukan yang diperlukan antara lain jumlah variabel *input*, nilai variabel *input*, bobot, alpha (*learning rate*), *threshold*, maksimum epoch dan target (*output*).
4. Iterasi dilakukan terus hingga semua pola memiliki keluaran jaringan yang sama dengan targetnya.

2. PERCEPTRON

2.1 Teori Jaringan Syaraf Tiruan (JST)

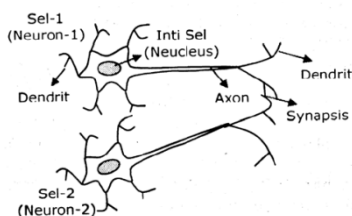
Elemen yang paling mendasar dari jaringan syaraf adalah sel syaraf. Sel-sel syaraf inilah membentuk bagian kesadaran manusia yang meliputi beberapa kemampuan umum. Pada dasarnya sel syaraf biologi menerima masukan dari sumber yang lain dan mengkombinasikannya dengan beberapa cara, melaksanakan suatu operasi yang non-linear untuk mendapatkan hasil dan kemudian mengeluarkan hasil akhir tersebut.

Jaringan syaraf tiruan dapat dibayangkan seperti otak buatan yang dapat berpikir seperti manusia dan juga sependai manusia dalam menyimpulkan sesuatu dari potongan-potongan informasi yang diterima. Khayalan manusia tersebut mendorong para peneliti untuk mewujudkannya. Komputer diusahakan agar bisa berpikir sama seperti cara berpikir manusia. Caranya adalah dengan melakukan peniruan terhadap aktivitas-aktivitas yang terjadi di dalam sebuah jaringan syaraf biologis. [KUS03]

Salah satu contoh pengambilan ide dari jaringan syaraf biologis adalah adanya elemen-elemen pemrosesan pada jaringan syaraf tiruan yang saling terhubung dan beroperasi secara paralel. Ini meniru jaringan syaraf biologis yang tersusun dari sel-sel syaraf (neuron). Cara kerja dari elemen-elemen pemrosesan jaringan syaraf tiruan juga sama seperti meng-encode informasi yang diterimanya.

Jaringan syaraf biologis merupakan kumpulan sel-sel syaraf (neuron). Neuron mempunyai tugas mengolah informasi. Komponen-komponen utama dari sebuah neuron dapat dikelompokkan menjadi tiga bagian, yaitu:

1. *Dendrit*. Dendrit bertugas untuk menerima informasi.
2. *Badan sel (soma)*. Badan sel berfungsi sebagai tempat pengolahan informasi.
3. *Akson (neurit)*. Akson mengirimkan impuls-impuls ke sel syaraf lainnya.



Gambar 1. Sel Syaraf Biologis [KUS03]

2.2 Jaringan Syaraf Tiruan Model Perceptron

Model jaringan perceptron ditemukan pertama kali oleh Rosenbatt (1962) dan Minsky – Papert (1969). Model tersebut merupakan model yang memiliki aplikasi dan pelatihan yang paling baik pada era tersebut. Perceptron merupakan salah satu bentuk jaringan sederhana, perceptron biasanya digunakan untuk mengklasifikasikan suatu pola tipe

tertentu yang sering dikenal dengan pemisahan secara linear. Pada dasarnya perceptron pada jaringan syaraf dengan satu lapisan memiliki bobot yang dapat diatur. Dapat digunakan dalam kasus untuk mengenali fungsi logika “dan” dengan masukan dan keluaran bipolar. [SIA05]

Arsitektur jaringan perceptron mirip dengan arsitektur jaringan Hebb. Jaringan terdiri dari beberapa unit masukan (ditambah sebuah bias), dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasi bukan merupakan fungsi biner (atau bipolar), tetapi memiliki kemungkinan nilai -1, 0 atau 1.

Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Fungsi aktivasinya dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan negatif.

2.2.1 Algoritma Pelatihan Perceptron

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 0 : Inisialisasi semua bobot dan bias (umumnya $w_i = b = 0$). Set laju pembelajaran α ($0 < \alpha \leq 1$) (untuk penyederhanaan set $\alpha = 1$). Kemudian set epoch = 0.
- Langkah 1 : 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.
- Langkah 2 : 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 3 : Set aktivasi unit masukan $x_i = s_i$ ($i = 1, \dots, n$)
- Langkah 4 : Hitung respon untuk unit output :

$$net = \sum_i x_i w_i + b \quad \dots \dots \dots (3.3)$$

$$f(net) = \begin{cases} 1 & \text{jika } net > \theta \\ 0 & \text{jika } -\theta \leq net \leq \theta \\ -1 & \text{jika } net < -\theta \end{cases} \quad \dots \dots \dots (3.4)$$

- Langkah 5 : 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 :

$$\Delta w_i = \alpha t x_i = t x_i \text{ (karena } \alpha = 1 \text{).}$$

$$\text{Bobot baru} = \text{bobot(lama)} + \Delta w_i$$

Langkah 6 : Test kondisi berhenti, jika tidak terjadi perubahan bobot pada epoch tersebut maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE

2.2.2 Proses Pengujian

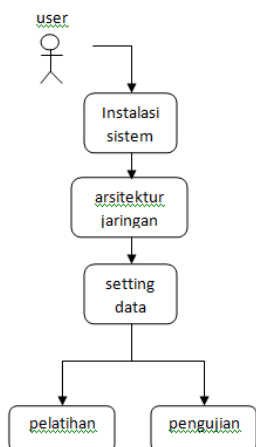
Proses pengujian merupakan tahap penyesuaian terhadap bobot yang telah terbentuk pada proses pelatihan. Algoritma untuk proses pengujian adalah sebagai berikut :

- Langkah 0 : Ambil bobot dari hasil pembelajaran,
- Langkah 1 : Untuk setiap vektor x , lakukan langkah 2 – 4,
- Langkah 2 : Set nilai aktivasi dari unit masukan, $x_i = s_i; i=1, \dots, n$,
- Langkah 3 : Hitung total masukan ke unit keluaran, $Net = \sum x_i w_i + b$,
- Langkah 4 : Gunakan fungsi aktivasi, $Y = f(net)$.

3. HASIL DAN PEMBAHASAN

3.1 Model Sistem

Gambaran umum perangkat lunak *tool* untuk struktur Jaringan Syaraf Tiruan (JST) model perceptron dapat dilihat pada gambar 2.



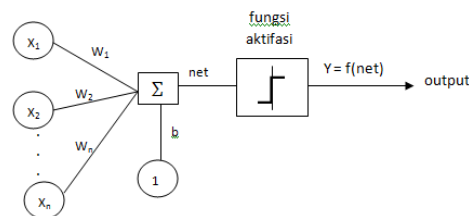
Gambar 2. Gambaran Umum Sistem

Dari gambar 2 diatas dapat dibagi menjadi beberapa bagian, yaitu instalasi sistem, arsitektur jaringan, *setting* data dan terakhir sistem dapat digunakan untuk membangun struktur jaringan melalui proses pengujian dan pelatihan.

3.2 Proses Pembuatan Arsitektur Jaringan

Pada proses pembuatan arsitektur jaringan, user harus mempunyai rancangan data yang akan digunakan untuk membangun sebuah struktur jaringan. Data (*input*) yang dibutuhkan seperti alpha, bobot, *threshold*, bias, jumlah variabel input, nilai variabel input dan target (*output*) yang nantinya akan digunakan dalam proses pelatihan dan pengujian sistem.

Pada gambar 3 menjelaskan bagaimana struktur akan dibangun berdasarkan *input* data yang akan dimasukan oleh *user*. *User* akan menentukan jumlah variabel input (X_1, X_2, \dots, X_n) yang akan dibangun, setelah itu *user* harus memasukan nilai variabel input berupa bilangan biner ataupun bipolar.



Gambar 3. Arsitektur JST Model Perceptron

Kemudian *user* memasukkan nilai bobot (umumnya nilai bobot adalah 0) dimana sistem akan memperbaharui nilai bobot selama masih ada *error* berdasarkan α adalah laju pembelajaran dan t adalah target keluaran (± 1). Fungsi aktivasi mempunyai kemungkinan nilai -1, 0, dan 1. Untuk nilai *threshold* θ yang ditentukan adalah berdasarkan pada persamaan 2.

$$Net = \sum x_i w_i + b \dots\dots\dots(1)$$

$$f(net) = \begin{cases} 1 & \text{jika } net > \theta \\ 0 & \text{jika } -\theta \leq net \leq \theta \\ -1 & \text{jika } net < -\theta \end{cases} \dots\dots\dots(2)$$

3.3 Analisis Kebutuhan Masukan

Kebutuhan masukan dari *user* dibutuhkan untuk menentukan struktur jaringan. Masukan yang dibutuhkan adalah :

- a. Jumlah variabel *input*.
- b. Nilai variabel *input*.
- c. Bobot awal.
- d. Alpha.
- e. *Threshold*.
- f. Maksimum epoch.
- g. Target (*output*).

3.4 Analisis Kebutuhan Keluaran

Sedangkan kebutuhan keluaran yang dihasilkan oleh *tool* untuk JST model perceptron yaitu hasil perhitungan dan gambar struktur Jaringan Syaraf Tiruan (JST) model perceptron.

3.5 Analisis Kebutuhan Proses

Kebutuhan proses dalam pembuatan *tool* untuk struktur Jaringan Syaraf Tiruan (JST) model perceptron antara lain proses *input* data untuk pembentukan jaringan. Kebutuhan proses *tool* untuk JST model perceptron adalah sebagai berikut :

- a. Proses *input* data ke sistem. *Input* berupa parameter JST model perceptron seperti jumlah

variabel *input*, nilai variabel *input*, bobot, alpha (*learning rate*), *threshold*, maksimum epoch dan target (*output*).

- b. Proses perhitungan data dan pembentukan jaringan. Setelah semua data dimasukkan kemudian diproses untuk menghasilkan perhitungan perceptron dan struktur jaringan dapat divisualisasikan ke bidang yang ada

3.6 Kebutuhan Antar Muka

Kebutuhan antar muka yang diperlukan antara lain :

- a. Antar muka halaman *home*, sebagai halaman utama sistem.
- b. Antar muka untuk *input* data, halaman untuk memasukkan data seperti jumlah variabel *input*, nilai variabel *input*, bobot, alpha (*learning rate*), *threshold*, maksimum epoch dan target (*output*).
- c. Antar muka untuk informasi hasil perhitungan, halaman untuk menampilkan hasil perhitungan epoch.
- d. Antar muka untuk tabel perhitungan epoch terakhir, halaman untuk menampilkan tabel hasil perhitungan epoch yang terakhir.
- e. Antar muka untuk proses pengujian, halaman yang berguna untuk melakukan proses pengujian terhadap proses pembelajaran perceptron.
- f. Antar muka untuk struktur jaringan, halaman yang menampilkan struktur jaringan.
- g. Antar muka halaman *about*, merupakan halaman untuk memberikan informasi mengenai program dan *programmer*.
- h. Antar muka halaman *help*, merupakan halaman untuk panduan penggunaan *tool* untuk JST model perceptron.

3.7 Perancangan Sistem

Pada proses perancangan, diawali dengan membuat rancangan antarmuka perangkat lunak. Rancangan antarmuka yang dikembangkan pada perangkat lunak menggunakan sistem yang *user friendly*.

Tool untuk JST model perceptron ini sudah berbasis *GUI* dimana terdapat *icon-icon* yang dapat mempermudah user dalam pemakaian dan membuat menarik aplikasi perangkat lunak ini. Pemilihan bahasa pemrograman Java karena memiliki kemudahan dalam pemakaian, kelengkapan fungsi, stabil, aman dan *cross-platform* serta masih banyak lagi kelebihan dari Java.

Rancangan antarmuka dalam tool untuk JST model perceptron ini antara lain :

1. Antar Muka Halaman *Home*
2. Antar Muka Halaman *Input Data*
3. Antar Muka Halaman Hasil Data
4. Antar Muka Halaman Tabel Epoch Terakhir
5. Antar Muka Halaman Pengujian Data
6. Antar Muka Halaman Struktur Jaringan

7. Antar Muka Halaman *About*
8. Antar Muka Halaman *Help*

3.8 Kinerja Yang Diharapkan

Dari hasil analisis yang sudah dijelaskan sebelumnya, diharapkan aplikasi yang dibuat nantinya mampu menangani berbagai macam model Jaringan Syaraf Tiruan.

4. PENGUJIAN

Proses pengujian tool untuk JST model perceptron ini meliputi proses pelatihan dan pengujian pada data masukan oleh user.

Pada tahap pengujian dan analisis *tool* untuk struktur Jaringan Syaraf Tiruan (JST) model Perceptron, dilakukan perbandingan antara kebenaran serta kesesuaian program dengan kebutuhan sistem. Setelah sistem melakukan proses pelatihan dan pengujian, user dapat melihat struktur jaringan yang telah terbentuk.

4.1 Input Data Awal

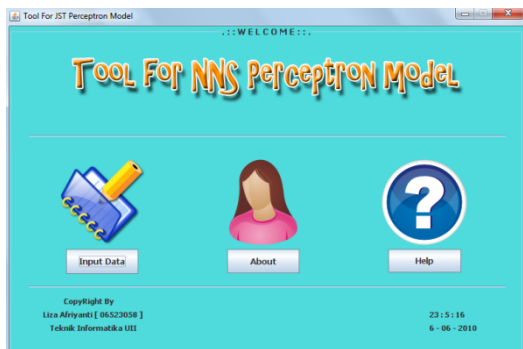
Untuk melakukan proses pelatihan dan pengujian perceptron, langkah pertama yang harus dilakukan oleh *user* adalah memasukkan data awal seperti jumlah variabel *input*, nilai variabel *input*, bobot, alpha (*learning rate*), *threshold*, maksimum epoch dan target (*output*).

Misalkan untuk inisialisasi awal tiap parameter yang digunakan adalah jumlah variabel *input* (x) = 2, bobot awal (w) = 0, alpha (α) = 0.8, *threshold* (θ) = 0.5 dan maksimum epoch adalah 11 dengan *input* biner dan target bipolar.

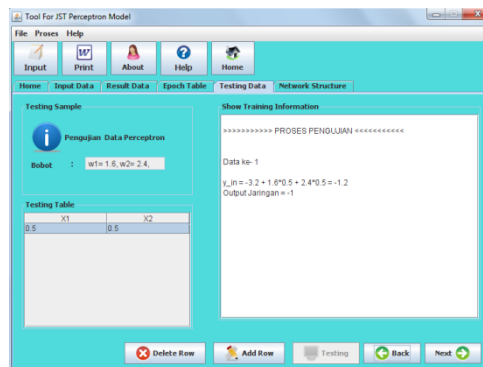
4.2 Implementasi

Pada implementasi *tool* untuk Jaringan Syaraf Tiruan (JST) model perceptron akan dijelaskan bagaimana sistem ini bekerja, dengan memberikan tampilan form-form dan tampilan output yang dibuat. Gambar 4 merupakan halaman home. Gambar 5 adalah halaman untuk input data awal seperti jumlah variabel *input*, nilai variabel *input*, bobot, alpha (*learning rate*), *threshold*, maksimum epoch dan target (*output*).

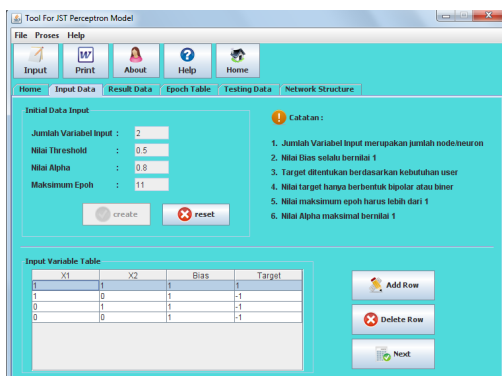
Gambar 6 adalah halaman hasil perhitungan proses pelatihan perceptron. Gambar 7 merupakan tabel epoch hasil iterasi terakhir pada proses pelatihan.



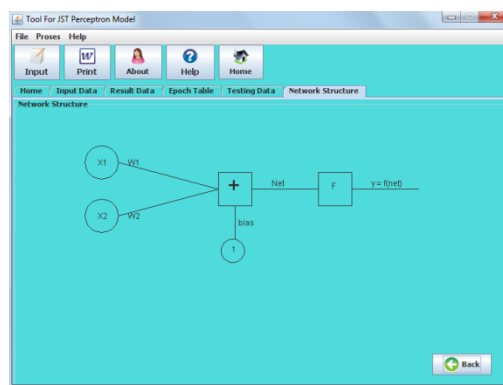
Gambar 4. Home



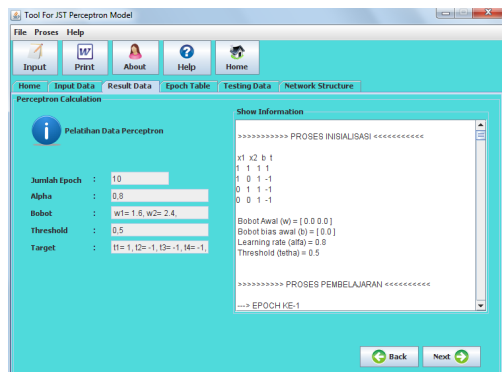
Gambar 8. Proses Pengujian



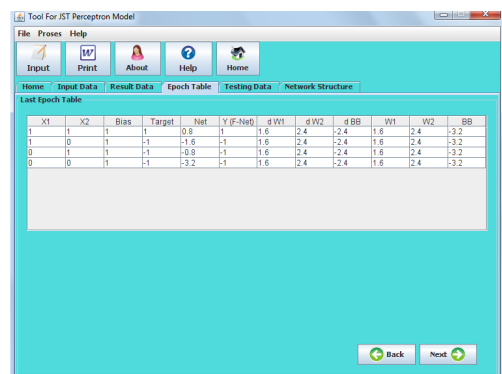
Gambar 5. Input Data Awal



Gambar 9. Struktur Jaringan



Gambar 6. Result Data



Gambar 7. Tabel Epoch Terakhir

Gambar 8 menunjukkan proses pengujian pada sistem. Bobot yang digunakan pada proses pengujian adalah bobot terakhir proses pelatihan dan user memasukkan nilai variabel input yang baru untuk proses pengujian. Gambar 9 merupakan gambar struktur jaringan. Jumlah neuron dibentuk berdasarkan jumlah variabel input oleh user pada halaman input data awal.

5. KESIMPULAN

Dari hasil penelitian dan pembahasan yang telah dilakukan, dapat diambil kesimpulan bahwa *tool* untuk struktur Jaringan Syaraf Tiruan (JST) model perceptron :

1. Mampu melakukan proses pelatihan data berdasarkan nilai masukkan dari user seperti jumlah variabel *input*, nilai variabel *input*, bobot, alpha (*learning rate*), *threshold*, maksimum epoch dan target (*output*).
2. Mampu melakukan proses pengujian data berdasarkan nilai bobot yang didapat dari proses pelatihan.
3. Sistem dapat membuat struktur Jaringan Syaraf Tiruan model perceptron. Jumlah neuron terbentuk berdasarkan jumlah variabel *input*.

Diharapkan dalam pengembangan sistem selanjutnya dapat mencakup beberapa metode jaringan syaraf tiruan (jst) seperti model hebb,

model adaline, back propagation, dan lain-lain. Penyajian data akan lebih baik jika dapat menampilkan keseluruhan hasil perhitungan termasuk tabel epoch pada setiap iterasi.

PUSTAKA

- [BAM03] Hariyanto, Bambang. *Esensi-esensi Bahasa Pemrograman JAVA*. Penerbit Informatika. Bandung. 2003.
- [RAC09] Hakim, Rachmad. *Mastering Java (Konsep Pemrograman Java dan Penerapannya untuk membuat software aplikasi)*. Elex Media Komputindo. 2009.
- [KUS03] Kusumadewi, Sri. *Artificial Intelligence (teknik dan aplikasinya)*. Graha Ilmu, Yogyakarta. 2003.
- [NAS09] Nasution, Fithlail Gudie. *Jaringan saraf tiruan (Neural Network)*. <http://fithlail.web.id/category/neuralnetwork/> diakses tanggal 17 Maret 2009.
- [ISA02] Rickyanto, Isak. *Dasar Pemrograman Berorientasi Objek Dengan JAVA 2 (JDK 1.4)*. Penerbit Andi. Yogyakarta. 2002.
- [SIA05] Siang, Jong Jek. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Penerbit Andi. Yogyakarta. 2005.