

APLIKASI BIDIRECTIONAL ASSOSIATIF MEMORI (BAM) NETWORK PADA PENGENALAN MODEL

Iskhaq Iskandar
Jurusan Fisika FMIPA Universitas Sriwijaya

ABSTRAK

Penelitian ini bertujuan untuk menyusun suatu simulasi komputer yang dapat dipergunakan untuk menguji kemampuan memori komputer dalam mengenali suatu model tertentu berdasarkan algoritma Bidirectional Assosiatif Memori Neural Network. Model yang digunakan dalam penelitian ini adalah huruf-huruf abjad yang dinyatakan dalam kode bipolar -1 dan +1 dalam bentuk matrik [5x3]. Hasil yang didapat dalam penelitian ini menunjukkan bahwa rancangan network yang disusun mampu mengenali model input yang diberikan walaupun model tersebut mengandung noise. Network masih tetap memberikan respon yang cukup baik untuk model input yang mengandung maksimum 4 (empat) buah noise untuk model input yang dinyatakan dalam vektor X, dan 1 (satu) buah noise untuk model input yang dinyatakan dalam vektor Y.

PENDAHULUAN

Neural Network merupakan bagian dari kemampuan komputer untuk mengingat model atau input yang diberikan. Ide ini ditujukan agar komputer dapat menyimpan memori sehingga komputer dapat memanggilnya kembali ketika input yang menyerupai memori yang telah disimpan, diberikan ke komputer^{1,2,3,9]}.

Salah satu bentuk Neural Network adalah *Bidirectional Assosiatif Memori* yang termasuk dalam bentuk algoritma *Assosiatif Memori Neural Network*. Assosiatif memori neural network adalah suatu bentuk network

yang bobot-bobotnya ditentukan sedemikian rupa sehingga network dapat menyimpan suatu kumpulan data yang membentuk suatu model tertentu^{2,4,5,6,10]}.

Berangkat dari fenomena di atas, maka timbul gagasan untuk membuat suatu simulasi komputer yang dapat digunakan untuk pembelajaran suatu model berdasarkan model tertentu yang telah disimpan dalam memori komputer. Model yang digunakan dalam penelitian ini adalah huruf-huruf abjad yang diwakili oleh kode bipolar -1 dan +1. Model ini dinyatakan dalam bentuk matrik [5 x 3]. Matrik bobot pada tiap-tiap model didapat dengan menggunakan *Aturan Hebb*.

Selanjutnya, pengujian kemampuan memori komputer dilakukan dengan memberikan model input yang sama atau hampir sama dengan model yang telah tersimpan dalam memori komputer.

Identifikasi masalah dalam penelitian ini adalah sebagai berikut; *inisialisasi matrik bobot dengan menggunakan Aturan Hebb (Hebb Rule); pemberian model input ke tiap-tiap unit pada tiap-tiap lapisan neuron; set aktivasi tiap-tiap unit dalam tiap lapisan; tes konvergen untuk fungsi aktivasi pada tiap-tiap lapisan.*

Permasalahan yang diteliti dibatasi pada aplikasi Bidirectional Asosiatif Memori Neural Network pada pengenalan model. Model yang digunakan dalam penelitian ini adalah huruf-huruf abjad yang diwakili oleh kode bipolar -1 dan +1. Model-model tersebut dinyatakan dalam bentuk matrik [5 x 3]. Matrik bobot pada tiap-tiap model ditentukan dengan menggunakan *Aturan Hebb (Hebb Rule)*.

TINJAUAN PUSTAKA

Asosiatif memori neural network adalah suatu bentuk network yang bobot-bobotnya ditentukan sedemikian rupa

sehingga network dapat menyimpan suatu kumpulan data yang membentuk satu model tertentu. Tiap-tiap kumpulan data merupakan pasangan vektor $(s(p), t(p))$, dengan $p = 1, 2, \dots, p$. Tiap-tiap vektor $s(p)$ mempunyai n komponen dan tiap-tiap vektor $t(p)$ mempunyai m komponen. Selanjutnya, bobot-bobot pada network dapat ditentukan dengan menggunakan *Aturan Hebb (Hebb Rule)* atau *Aturan Delta (delta Rule)*^{2,7,8,10}.

Secara umum, di dalam Aturan Hebb, untuk menyimpan pasangan vektor $s(p), t(p)$, dengan $p = 1, 2, \dots, p$, dimana :

$$s(p) = [s_1(p), \dots, s_i(p), \dots, s_n(p)] \dots\dots\dots (1)$$

dan

$$t(p) = [t_1(p), \dots, t_j(p), \dots, t_m(p)] \dots\dots\dots (2)$$

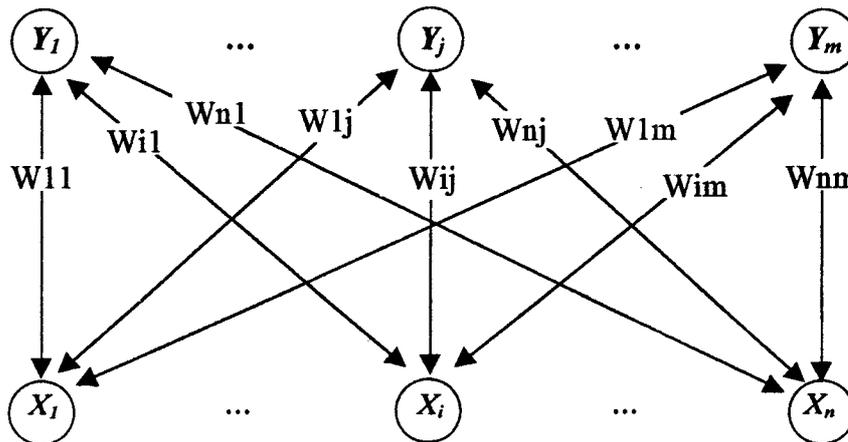
Matrik bobot $W = \{w_{ij}\}$ diberikan oleh :

$$w_{ij} = \sum_{p=1}^p s_i(p)t_j(p) \dots\dots\dots (3)$$

Selanjutnya, Aturan Hebb akan digunakan dalam simulasi pada usulan penelitian ini untuk menentukan matrik bobot.

Arsitektur Bidirectional Assosiatif Memori (BAM)

Arsitektur Bidirectional Assosiatif Memori (BAM) network ditunjukkan pada gambar 1. di bawah ini,



Gambar 1. Arsitektur BAM Network (Fausett, L., 1994)

Dari gambar 1. di atas nampak bahwa BAM network adalah network lapisan tunggal (*single-layer network*) yang bersifat nonlinear dan feedback. BAM network mempunyai n unit dalam lapisan-X dan m unit dalam lapisan-Y. Hubungan antar lapisan (*layer*) bersifat dua arah, yang artinya jika matrik bobot untuk sinyal-sinyal yang dikirim dari lapisan-X ke lapisan-Y adalah W , maka matrik bobot untuk sinyal-sinyal yang dikirim dari lapisan-Y ke lapisan-X adalah W^T .

Dalam iterasi BAM network, sinyal-sinyal dikirim secara dua arah, dari lapisan-X ke lapisan-Y secara bergantian dan terus-

menerus hingga seluruh neuron mencapai keseimbangan. Keseimbangan yang dimaksud di sini adalah suatu keadaan dimana tiap-tiap aktivasi neuron tetap konstan untuk beberapa langkah iterasi.

Penentuan Bobot-bobot Koneksi

Matrik bobot untuk menyimpan vektor input dan vektor target $[s(p), t(p)]$, dimana $p = 1, 2, \dots, p$ dengan ^{2,4,101};

$$s(p) = [s_1(p), \dots, s_i(p), \dots, s_n(p)]$$

dan

$$t(p) = [t_1(p), \dots, t_j(p), \dots, t_m(p)]$$

dapat ditentukan dengan *Aturan Hebb (Hebb Rule)*. Perumusan untuk masukan tergantung pada apakah vektor-vektor pelatihan dalam bentuk biner atau bipolar. Untuk vektor input yang berbentuk biner, matrik bobot $W = \{w_{ij}\}$ ditentukan dengan

$$w_{ij} = \sum_p [2s_i(p) - 1][2t_j(p) - 1] \dots(4)$$

dan untuk vektor input bipolar,

$$w_{ij} = \sum_p s_i(p)t_j(p) \dots\dots\dots(5)$$

Fungsi Aktivasi

Fungsi aktivasi BAM network adalah berbentuk fungsi langkah (*step function*), tergantung pada apakah vektor biner atau bipolar yang digunakan. Untuk vektor input biner, fungsi aktivasi lapisan-Y adalah ²¹ :

$$y_j = \begin{cases} 1. \text{ jika } y_in_j > 0 \\ y_j \text{ jika } y_in_j = 0 \\ 0. \text{ jika } y_in_j < 0 \end{cases} \dots\dots\dots (6)$$

dan fungsi aktivasi untuk lapisan-X adalah :

$$x_i = \begin{cases} 1. \text{ jika } x_in_i > 0 \\ x_j \text{ jika } x_in_i = 0 \\ 0. \text{ jika } x_in_i < 0 \end{cases} \dots\dots\dots (7)$$

Sedangkan untuk vektor input bipolar, fungsi aktivasi untuk lapisan-Y adalah :

$$y_j = \begin{cases} 1. \text{ jika } y_in_j > \theta_j \\ y_j \text{ jika } y_in_j = \theta_j \\ -1. \text{ jika } y_in_j < \theta_j \end{cases} \dots\dots\dots (8)$$

dan fungsi aktivasi untuk lapisan-X adalah :

$$x_i = \begin{cases} 1. \text{ jika } x_in_i > \theta_i \\ x_i \text{ jika } x_in_i = \theta_i \\ -1. \text{ jika } x_in_i < \theta_i \end{cases} \dots\dots\dots (9)$$

dimana θ_i adalah fungsi "*threshold*" yang nilainya biasanya diambil sama dengan 0 (nol).

METODE PENELITIAN

1. Persiapan

- 1. 1. Studi literatur
- 1. 2. Setup peralatan komputer

2. Pelaksanaan

2.1. Pemrograman

- 2.1.1. Penyusunan sub program 1 untuk menentukan matrik bobot berdasarkan Aturan Hebb.
- 2.1.2. Penyusunan sub program 2 untuk menentukan nilai vektor input dan output.
- 2.1.3. Penyusunan sub program 3 untuk menentukan nilai fungsi dari aktivasi

2.1.4. Penyusunan sub program 4 untuk simulasi bidirectional assosiatif memori neural network.

2.2. Algoritma Rancangan Program

Algoritma rancangan program secara garis besar dapat ditunjukkan pada langkah-langkah di bawah ini :

Step 1. Inisialisasi matrik bobot untuk menyimpan suatu vektor P
Inisialisasi seluruh aktivasi ke 0

Step 2. Untuk tiap-tiap vektor input, lakukan *step 3-7*.

Step 3a. Berikan model input x ke lapisan-X
(set aktivasi lapisan-X ke model input sekarang)

Step 3b. Berikan model input y ke lapisan-Y
(model input boleh vektor nol)

Step 4. Jika aktivasi tidak konvergen, lakukan *step 5-7*.

Step 5. Ubah aktivasi unit-unit dalam lapisan-Y
Hitung input network

$$y_in_j = \sum w_{ij} x_i$$

Hitung aktivasi

$$y_j = f(y_in_j)$$

Kirim sinyal ke lapisan-X.

Step 6. Ubah aktivasi unit-unit dalam lapisan-X

Hitung input network

$$x_in_i = \sum w_{ij} y_j$$

Hitung aktivasi

$$x_j = f(x_in_i)$$

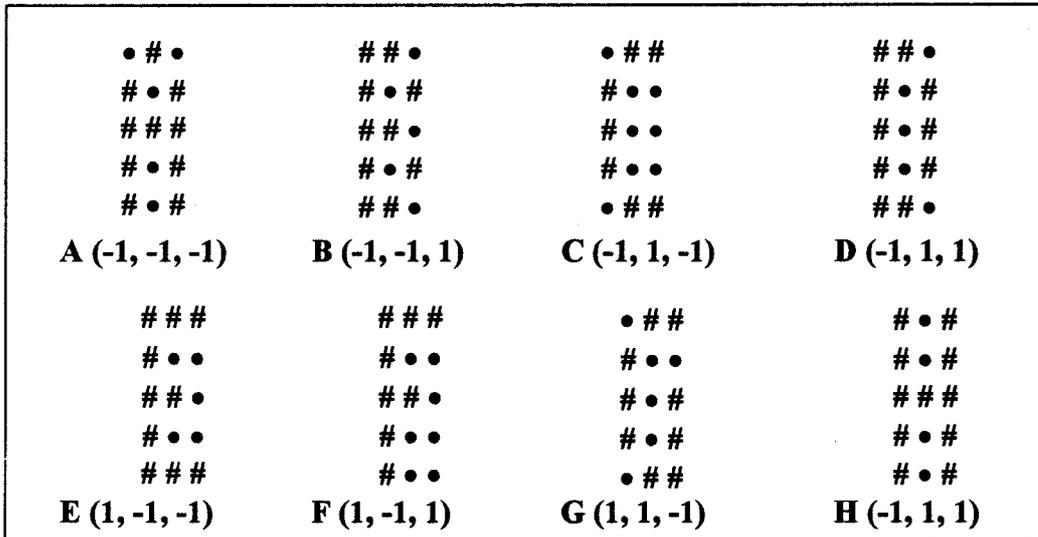
Kirim sinyal ke lapisan-Y.

Step 7. Tes konvergen aktivasi.

Jika aktivasi vektor x dan y telah mencapai keseimbangan, iterasi dihentikan. Jika tidak maka iterasi dilanjutkan.

HASIL DAN PEMBAHASAN

Model yang digunakan dalam simulasi komputer adalah huruf-huruf abjad dan dibatasi hanya untuk huruf-huruf A, B, C, D, E, F, G dan H. Model-model tersebut direpresentasikan dalam kode bipolar dalam bentuk matrik $[5 \times 3]$. Berikut disajikan model-model yang digunakan dalam penelitian ini.



Gambar 2. Model yang digunakan dalam simulasi komputer
 (# = 1 dan • = -1)

Proses simulasi diawali dengan menentukan matrik bobot $W = \{w_{ij}\}$ untuk tiap-tiap model. Penentuan matrik bobot ini menggunakan *Aturan Hebb (Hebb Rule)* yang dinyatakan dalam *persamaan (5)*. Sebagai analog akan dibahas cara menentukan matrik bobot untuk salah satu model penelitian, yaitu model huruf abjad A (-1, -1, -1). Model huruf abjad A jika direpresentasikan dalam bentuk vektor akan menjadi :

$$\begin{pmatrix} -1, 1, -1 \\ 1, -1, 1 \\ 1, 1, 1 \end{pmatrix}$$

Adanya spasi antar komponen-komponen vektor tersebut menandakan pergantian baris seperti yang ditunjukkan pada model dalam bentuk matrik pada *gambar 2* di atas. Dengan menggunakan *Aturan Hebb* akan didapat matrik bobot untuk model huruf abjad A (-1, -1, -1) sebagai berikut :

$$\begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

A
W

Gambar 3. Matrik bobot untuk Model Huruf Abjad A (-1, -1, -1)

Matrik bobot untuk model-model yang lain ditentukan dengan cara yang sama dengan di atas, yaitu dengan menggunakan *Aturan Hebb*, sehingga tiap-tiap model memiliki matrik bobot sendiri agar model dapat disimpan dalam memori komputer.

Matrik bobot yang dapat digunakan untuk menyimpan 4 (empat) buah model,

masing-masing adalah model *A, B, C, D* dan model *E, F, G, H*, didapat dengan cara menjumlahkan matrik bobot tiap-tiap model tersebut. Hasilnya dinyatakan dalam matrik-matrik berikut.

$$\begin{array}{cc}
 \begin{bmatrix} 0 & 0 & 4 \\ -4 & 0 & 0 \\ 2 & 2 & -2 \\ -4 & 0 & 0 \\ 4 & 0 & 0 \\ -2 & -2 & 2 \\ -4 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 4 \\ -4 & 0 & 0 \\ 4 & 0 & 0 \\ -2 & -2 & 2 \\ -2 & -2 & 2 \\ -2 & 2 & 2 \\ 0 & 0 & -4 \end{bmatrix} &
 \begin{bmatrix} 2 & -2 & 2 \\ 2 & -2 & -2 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ -4 & 0 & 0 \\ -2 & 2 & 2 \\ 4 & 0 & 0 \\ 2 & -2 & 2 \\ 0 & 4 & 0 \\ 4 & 0 & 0 \\ -4 & 0 & 0 \\ 0 & 4 & 0 \\ 2 & -2 & 2 \\ 0 & 0 & -4 \\ 2 & 2 & -2 \end{bmatrix} \\
 \text{(a)} & \text{(b)}
 \end{array}$$

Gambar 4. (a) Matrik bobot untuk Model Huruf Abjad A, B, C, D
 (b) Matrik bobot untuk Model Huruf Abjad E, F, G, H

Setelah matrik bobot didapat, proses selanjutnya adalah menguji kemampuan network. Pertama-tama akan kita lihat bahwa network akan memberikan respon yang baik, berupa vektor Y yang benar, ketika diberikan input model, berupa vektor X . Sebagai

contoh, diberikan input model huruf abjad **B** yang dinyatakan oleh vektor X berikut [1 1 -1 1 -1 1 1 1 -1 1 -1 1 1 1 -1]. Proses pengenalan model oleh network dapat dilihat dalam uraian berikut :

$$\begin{array}{l}
 \text{model input} \\
 [11-1 \ 1-11 \ 11-1 \ 1-11 \ 11-1]
 \end{array}
 \begin{array}{l}
 \text{matrik bobot} \\
 \begin{bmatrix}
 0 & 0 & 4 \\
 -4 & 0 & 0 \\
 2 & 2 & -2 \\
 -4 & 0 & 0 \\
 4 & 0 & 0 \\
 -2 & -2 & 2 \\
 -4 & 0 & 0 \\
 0 & -4 & 0 \\
 0 & 0 & 4 \\
 -4 & 0 & 0 \\
 4 & 0 & 0 \\
 -2 & -2 & 2 \\
 -2 & -2 & 2 \\
 -2 & 2 & 2 \\
 0 & 0 & -4
 \end{bmatrix}
 \end{array}
 = [-32, -10, 14]$$

Sesuai dengan fungsi aktivasi yang dinyatakan dalam persamaan 8, dimana jika $y_{in} < \theta$ maka $y_{out} = -1$, jika $y_{in} = \theta$ maka $y_{out} = y_{in}$ dan jika $y_{in} > \theta$ maka $y_{out} = 1$ (dalam penelitian ini θ diambil sama dengan nol), sehingga hasil respon dari network adalah [-1, -1, 1] yang merupakan model huruf abjad **B** yang dinyatakan dalam vektor **Y**.

Sifat *bidirectional* network dapat ditunjukkan dengan cara menjadikan vektor **Y**

sebagai input dan hasilnya dinyatakan dalam vektor **X**. Matrik bobot untuk sinyal yang dikirim dari lapisan **Y** ke lapisan **X**, merupakan *transpose* matrik bobot yang dikirim dari lapisan **X** ke lapisan **Y**. Sebagai contoh, *transpose* matrik bobot untuk model **A, B, C, D** adalah :

$$W^T = \begin{bmatrix}
 0 & -4 & 2 & -4 & 4 & -2 & -4 & 0 & 0 & -4 & 4 & -2 & -2 & -2 & 0 \\
 0 & 0 & 2 & 0 & 0 & -2 & 0 & -4 & 0 & 0 & 0 & -2 & -2 & 2 & 0 \\
 4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 & 2 & 2 & 2 & -4
 \end{bmatrix}$$

Jika input yang diberikan berupa vektor Y , yaitu model huruf abjad C yang dinyatakan oleh $[-1, 1, -1]$, maka respon yang akan

diberikan oleh network adalah sebagai berikut :

$$\begin{aligned}
 [-1 \ 1 \ -1] & \begin{bmatrix} 0 & -4 & 2 & -4 & 4 & -2 & -4 & 0 & 0 & -4 & 4 & -2 & -2 & -2 & 0 \\ 0 & 0 & 2 & 0 & 0 & -2 & 0 & -4 & 0 & 0 & 0 & -2 & -2 & 2 & 0 \\ 4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 & 2 & 2 & 2 & -4 \end{bmatrix} \\
 & = [-4 \ 4 \ 2 \ 4 \ -4 \ -2 \ 4 \ -4 \ -4 \ 4 \ -4 \ -2 \ -2 \ 2 \ 4]
 \end{aligned}$$

Sesuai dengan fungsi aktivasi yang dinyatakan dalam persamaan 9, dimana jika $x_{in} < \theta$ maka $x_{out} = -1$, jika $x_{in} = \theta$ maka $x_{out} = x_{in}$ dan jika $x_{in} > \theta$ maka $x_{out} = 1$ (dalam penelitian ini θ diambil sama dengan nol), sehingga hasil respon dari network adalah $[-1 \ 1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1]$ yang

merupakan model huruf abjad C yang dinyatakan dalam vektor X .

Selanjutnya, jika input yang diberikan mengandung noise, misal input dinyatakan dalam vektor $Y [-1, -1, 0]$, dimana komponen "0" dianggap sebagai noise, maka respon yang diberikan oleh network adalah sebagai berikut :

$$\begin{aligned}
 [-1 \ -1 \ 0] & \begin{bmatrix} 0 & -4 & 2 & -4 & 4 & -2 & -4 & 0 & 0 & -4 & 4 & -2 & -2 & -2 & 0 \\ 0 & 0 & 2 & 0 & 0 & -2 & 0 & -4 & 0 & 0 & 0 & -2 & -2 & 2 & 0 \\ 4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 & 2 & 2 & 2 & -4 \end{bmatrix} \\
 & = [0 \ 4 \ -4 \ 4 \ -4 \ 4 \ 4 \ 4 \ 0 \ 4 \ -4 \ 4 \ 4 \ 0 \ 0] \\
 & = [0 \ 1 \ -1 \ 1 \ -1 \ 1 \ 1 \ 1 \ 0 \ 1 \ -1 \ 1 \ 1 \ 0 \ 0]
 \end{aligned}$$

Karena hasil yang didapat belum konvergen, dimana masih mengandung komponen yang belum dikenal oleh network, komponen "0",

maka dilakukan iterasi dengan cara mengirim hasil yang didapat ke lapisan Y , sehingga akan didapat :

$$\begin{bmatrix} 0 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 0 & 1 & -1 & 1 & 1 & 0 & 0 \end{bmatrix}
 \begin{bmatrix} 0 & 0 & 4 \\ -4 & 0 & 0 \\ 2 & 2 & -2 \\ -4 & 0 & 0 \\ 4 & 0 & 0 \\ -2 & -2 & 2 \\ -4 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 4 \\ -4 & 0 & 0 \\ 4 & 0 & 0 \\ -2 & -2 & 2 \\ -2 & -2 & 2 \\ -2 & 2 & 2 \\ 0 & 0 & -4 \end{bmatrix}$$

$$= [-32, -12, 8] \Rightarrow [-1 \ -1 \ 1]$$

Jika hasil yang didapat tersebut dikirimkan lagi ke lapisan *X*, maka akan didapat hasil yang dinyatakan dalam vektor *X* sebagai berikut :

$$[1 \ 1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1]$$

yang merupakan model huruf abjad **B** yang dinyatakan dalam vektor *X*. Jadi model input

berupa vektor *Y* yang mengandung 1 (satu) buah noise masih mampu dikenali oleh network.

Begitu juga jika model input yang diberikan dalam vektor *X* yang mengandung noise, maka respon yang diberikan oleh network adalah sebagai berikut:

$$\begin{bmatrix} 1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & 1 & -1 \end{bmatrix}
 \begin{bmatrix} 0 & 0 & 4 \\ -4 & 0 & 0 \\ 2 & 2 & -2 \\ -4 & 0 & 0 \\ 4 & 0 & 0 \\ -2 & -2 & 2 \\ -4 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 4 \\ -4 & 0 & 0 \\ 4 & 0 & 0 \\ -2 & -2 & 2 \\ -2 & -2 & 2 \\ -2 & 2 & 2 \\ 0 & 0 & -4 \end{bmatrix}
 = [-28, 0, 20] \Rightarrow [-1 \ 0 \ 1]$$

Karena hasil yang didapat belum konvergen, masih mengandung komponen yang belum dikenali oleh network, *komponen "0"*, maka dilakukan iterasi dengan cara mengirim hasil yang didapat ke lapisan *X* sehingga akan didapat hasil sebagai berikut

[1 1 -1 1 -1 1 1 0 1 1 -1 1 1 -1]. Hasil ini juga belum konvergen, maka dikirim kembali ke lapisan *Y*, sehingga akan didapat hasil [-34, -6, 22] \Rightarrow [-1, -1, 1] yang merupakan model huruf abjad **B** yang dinyatakan dalam vektor *Y*. Jadi, jika input yang diberikan mengandung noise, maka

network akan melakukan iterasi dalam proses pengenalannya hingga model input mampu dikenali.

KESIMPULAN DAN SARAN

1. Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut :

- 1.1. Secara umum penelitian ini berhasil mencapai sasaran, dalam arti rancangan sistem perangkat lunak komputer (*software*) yang

dihasilkan dapat dipergunakan untuk proses pengenalan model menggunakan algoritma *Bidirectional Assosiatif Memori* (BAM) network.

- 1.2. Rancangan network yang dihasilkan dalam penelitian ini mampu mengenali model input yang mengandung noise. Network masih dapat memberikan respon yang cukup baik untuk model input yang mengandung maksimum 4 (empat) buah noise untuk model input yang dinyatakan dalam vektor X , dan 1 (satu) buah noise untuk model input yang dinyatakan dalam vektor Y .

2. Saran

Penelitian ini dapat dikembangkan dengan mencoba menerapkan beberapa algoritma neural network yang lain, seperti *back propagation network*, *Hopfield network*, atau *adaptive resonance theory network*. Dari segi aplikasi, penelitian ini dapat dikembangkan dengan menerapkan algoritma yang telah digunakan dalam

penelitian ini untuk kasus-kasus yang lain, seperti *traveling salesmen problem*, *wave recognition*, dan lain-lain.

DAFTAR PUSTAKA

- Ashida, Y dan Sassa, K, 1994, *Recognition of Waveform by use of Neural Network*, Butsiritansa 47, pp : 42 - 48
- Fausett, L, 1994, *Fundamentals of Neural Networks* Prentice Hall, Englewood Cliffs
- Gould, H dan Tobocnik, J, 1996, *An Introduction to Computer Simulation Methods*, Addison - Wesley Publishing Company, Inc.
- Hertz, J, Krogh, A dan Palmer, G, 1991, *Introduction to the Theory of Neural Computation*, Addison - Wesley Publishing Company, Inc.
- Iskandar, I, 1999, *Neural Network : Teori Dasar dan Contoh Aplikasi Sederhana* Jurnal Ilmiah MIPA, BKS - PTN Wilayah Barat
- Iskandar, I, 1999, *Model Hopfield sebagai Salah Satu Model Assosiatif Memori Neural Network*, Proceeding Jurusan Fisika, FMIPA Universitas Sriwijaya
- L. Chester, Michael, 1993, *Neural Network; A Tutorial* Prentice Hall Englewood Cliffs

Palaz, I. dan Weger, R.C., 1990, *Waveform Recognition using Neural Network* The Leading Edge 9, pp : 28 - 31

P.M.C. de Oliveira, 1997, *Computer Simulation of Neural Network*

Computer in Physics, Vol 11 No. 5, pp : 443 - 449

Soulie, F. 1992, *Neural Network for Pattern Recognition* World Scientific, New Jersey