

# IDENTIFIKASI POLA SIDIK JARI DENGAN JARINGAN SYARAF TIRUAN BIDIRECTIONAL ASSOCIATIVE MEMORY (Finger Print Pattern Identification by Bidirectional Associative Memory Artificial Neural Network)

Anifuddin Azis<sup>1</sup>, dan Tanzil Kurniawan<sup>1</sup>

<sup>1</sup>Program Studi Ilmu Komputer FMIPA Universitas Gadjah Mada  
email : [anifudin@ugm.ac.id](mailto:anifudin@ugm.ac.id)

## ABSTRAK

Jaringan Syaraf Tiruan Bidirectional Associative Memory (BAM) memiliki kemampuan sebagai *associative memory* atau *content addressable memory* yaitu memori yang dapat dipanggil menggunakan bagian dari informasi yang tersimpan di dalamnya. Selain itu JST BAM memiliki 2 lapisan dan terhubung penuh dari satu lapisan ke lapisan yang lainnya, sehingga dimungkinkan adanya hubungan timbal balik antara lapisan input dan lapisan output. Untuk itu dalam penelitian ini, akan dibuktikan kemampuan JST BAM tersebut dengan merancang sebuah sistem untuk mengidentifikasi pola sidik jari. Dari hasil uji coba program menunjukkan bahwa JST BAM dapat mengidentifikasi pola sidik jari dengan tepat setelah diberi *noise* secara acak antara 0% - 30%. Di atas 30% beberapa output yang dihasilkan tidak dapat dikenali atau dikenali namun sebagai pola palsu, sehingga JST BAM tidak dapat mengenali pola secara tepat.

**Kata kunci:** Jaringan Syaraf Tiruan BAM, pola sidik jari

## ABSTRACT

Bidirectional Associative Memory Artificial Neural Network has the feature of associative memory or content addressable memory. In this research we prove the capability of the system by designing a system to recognize fringer print. The result of program trial shows that the system can identify correctly fringer print pattern with added noise 0 % - 30 %. Above 30 % the system cannot recognize the pattern correctly.

**Keywords:** Artificial Neural Network BAM, fringer print pattern

Makalah diterima 23 Juli 2006

## 1. PENDAHULUAN

Jaringan Syaraf tiruan (JST) telah banyak diterapkan dalam menyelesaikan masalah dalam berbagai bidang. Dalam bidang ilmu komputer JST dapat digunakan untuk menemukan keterangan dari simbol-simbol atau angka-angka yang dihasilkan oleh sistem informasi. JST mempunyai kemampuan untuk menyimpan pengetahuan

yang didapatkan dari hasil latihan. Kemampuan ini mirip dengan fungsi otak manusia, sehingga sistem JST dapat digunakan pada pekerjaan-pekerjaan yang membutuhkan kecerdasan manusia.

Dalam melakukan proses belajar, JST dapat memodifikasi tingkah lakunya sesuai dengan keadaan lingkungannya. JST dapat mengatur dirinya sendiri untuk menghasilkan suatu respon yang konsisten

terhadap serangkaian masukan. Beberapa JST memiliki kemampuan untuk mengabstraksikan intisari serangkaian masukan. Aplikasi-aplikasi jaringan syaraf selama beberapa tahun ini umumnya berpusat pada tiga bidang utama, yaitu analisis data, pengenalan pola, dan fungsi kendali.

JST memiliki kemampuan yang sangat baik dalam teknik pengenalan pola (*pattern recognition*). Mula-mula JST ditunjukkan dengan beberapa buah pola sebagai bahan pembelajaran. Kemudian, JST diberikan pola yang tidak utuh yang memiliki sedikit modifikasi akibat distorsi ataupun derau (*noise*), ternyata JST masih mampu mengenali pola tersebut sebagai pola sempurna yang telah dipelajarinya. Misalnya, sebuah jaringan dapat dilatih untuk mengenali beberapa versi huruf S. Setelah menjalani cukup latihan, jaringan tersebut dapat menghasilkan huruf S yang sempurna dari masukan huruf-huruf S yang terdistorsi.

Konsep JST saat ini sangat berkembang dengan pesat. Banyak metode yang digunakan untuk membuat pemodelan JST, dan masing-masing metode mempunyai kelebihan dan kekurangannya. Beberapa diantaranya telah dikelompokkan berdasarkan kemampuan memori asosiasinya, yaitu kemampuan untuk menyimpan dan memanggil informasi yang telah disimpannya. Hal ini sama dengan kemampuan mengingat pada otak manusia. Yang memiliki kemampuan ini misalnya *Linier Associator*, JST Hopfield, dan *Bidirectional Associative Memory (BAM)*.

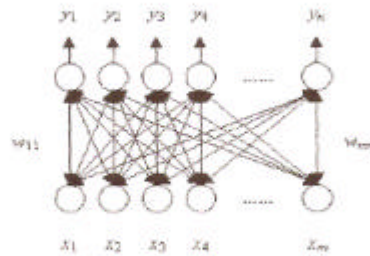
## 2. JARINGAN SYARAF TIRUAN BIDIRECTIONAL ASSOCIATIVE MEMORY (BAM)

Jaringan Syaraf Tiruan adalah sebuah sistem pengolahan informasi yang karakteristik kinerjanya menyerupai jaringan syaraf biologis. Jaringan Syaraf Tiruan telah berkembang sebagai model matematika dari pikiran manusia yaitu syaraf biologis, yang didasarkan pada asumsi sebagai berikut, (Fausett, 1994) :

a. Pemrosesan informasi terjadi pada banyak elemen sederhana yang disebut *neuron*

- b. Sinyal dilewatkan melalui sambungan antar *neuron*
- c. Setiap sambungan mempunyai bobot, yang menguatkan sinyal yang melaluinya.
- d. Setiap *neuron* menerapkan fungsi aktivasi ke lapisan input (jumlah sinyal input terboboti) untuk menentukan sinyal output.

Secara umum, arsitektur jaringan syaraf tiruan BAM ditunjukkan seperti pada Gambar 1. (Kusumadewi, 2003)



Gambar 1. Arsitektur Jaringan Syaraf Tiruan BAM

Ada 2 jenis jaringan syaraf tiruan BAM, yaitu (Kusumadewi, 2003);

### A. BAM Diskret

Pada BAM diskret ada 2 kemungkinan tipe data, yaitu biner dan bipolar. Matriks bobot awal dibuat sedemikian rupa sehingga dapat menyimpan pasangan vektor input dan vektor output  $s(p)-t(p)$ , dengan  $p = 1, 2, 3, \dots, P$ .

a. Untuk vektor input biner, matriks bobot ditentukan sebagai :

$$W_{ij} = \sum_p (2^* s_i(p) - 1)(2^* t_j(p) - 1)$$

Sedangkan fungsi aktivasi yang digunakan adalah :

? Untuk lapisan output :

$$y_j = \begin{cases} 1; & \text{jikay\_in}_j > 0 \\ y_j; & \text{jikay\_in}_j = 0 \\ 0; & \text{jikay\_in}_j < 0 \end{cases}$$

? Untuk lapisan input

$$x_i = \begin{cases} 1; & \text{jikax\_in}_i > 0 \\ x_i; & \text{jikax\_in}_i = 0 \\ 0; & \text{jikax\_in}_i < 0 \end{cases}$$

b. Sedangkan untuk vektor input bipolar, matriks bobot ditentukan sebagai :

$$W_{ij} = \sum_p (s_i(p) * t_j(p))$$

Sedangkan fungsi aktivasi yang digunakan adalah :

? Untuk lapisan output :

$$y_j = \begin{cases} 1; & \text{jikay\_in}_j > q \\ y_j; & \text{jikay\_in}_j = q \\ -1; & \text{jikay\_in}_j < q \end{cases}$$

? Untuk lapisan input

$$x_i = \begin{cases} 1; & \text{jikax\_in}_i > q \\ x_i; & \text{jikax\_in}_i = q \\ -1; & \text{jikax\_in}_i < q \end{cases}$$

Dengan catatan bahwa input hasil olahan pada jaringan ( $x\_in_i$  atau  $y\_in_j$ ) sama dengan nilai *threshold* nya (?), maka fungsi aktivasi akan menghasilkan nilai sama dengan nilai sebelumnya.

## B. BAM Kontinu

BAM kontinu akan mentransformasikan input secara lebih halus dan kontinu ke kawasan output dengan nilai yang terletak pada range [0,1]. Fungsi aktivasi yang digunakan adalah fungsi sigmoid.

Algoritma jaringan syaraf tiruan BAM adalah sebagai berikut :

**Langkah 0.** Inisialisasi bobot (untuk menyimpan sekumpulan P vektor).

Inisialisasi semua aktivasi sama dengan 0.

**Langkah 1.** untuk tiap-tiap input, kerjakan langkah 2-6 :

**Langkah 2a.** Berikan input pola x kelapisan X (kita set aktivasi lapisan X sebagai pola input).

**Langkah 2b.** Berikan input pola y ke lapisan Y (salah satu dari vektor input tersebut biasanya diset sebagai vektor nol)

**Langkah 3.** Kerjakan langkah 3-6 jika aktivasi-aktivasi tersebut belum konvergen :

**Langkah 4.** Perbaiki setiap unit aktivasi di lapisan Y :

Hitung :  $y\_in_j = \sum_i W_{ij} * x_i$

Hitung :  $y_j = f(y\_in_j)$

Berikan informasi tersebut ke lapisan X.

**Langkah 5.** Perbaiki setiap unit aktivasi di lapisan X :

Hitung :  $x\_in_i = \sum_j W_{ij} * x_i$

Hitung :  $x_i = f(x\_in_i)$

Berikan informasi tersebut ke lapisan Y.

**Langkah 6.** Tes kekonvergenan. Jika vektor x dan y telah mencapai keadaan stabil, maka interaksi berhenti, jika tidak demikian lanjutkan interaksi.

## 3. ANALISIS DAN PERANCANGAN SISTEM

Perancangan proses menitikberatkan pada urutan-urutan kegiatan yang terjadi dalam program. Yaitu diawali dengan memasukkan input data berupa pola sidik jari kemudian menyimpannya ke dalam database sebagai lapisan input. Secara random, akan ditentukan pada lapisan output yang akan digunakan untuk menghitung matriks bobot serta sebagai target output yang diharapkan. Berikutnya dilanjutkan dengan proses pelatihan dan pengujian pola-pola yang telah disimpan tersebut. Hasil dari proses pengujian akan menentukan apakah pola sidik jari tersebut dapat dikenali lagi atau tidak baik sebelum diberi *noise* maupun setelah diberi *noise*.

### 3.1 Perancangan Proses Masukan (Input)

Proses masukan atau input merupakan proses awal dijalankannya program. Dalam model *Bidirectional Associative Memory*, dibutuhkan 2 lapisan input dan output yang saling terhubung penuh dari satu lapisan ke

lapisan yang lainnya. Sehingga pada proses masukan, pola sidik jari yang berupa file gambar akan disimpan sebagai lapisan input. Kemudian akan dipilih secara acak kode vektor yang disimpan sebagai output. Lapisan output inilah nantinya yang akan menjadi target output yang diharapkan. Kedua lapisan tersebut akan disimpan dalam database yaitu dalam tInput.

**Tabel 1. Struktur Tabel Input**

Nama Atribut	Type Field	Ukuran	Kunci	Keterangan
NoID	Text	2	PK	Nomer urut pola masukan
Nama	Text	15		Nama pola masukan
Pict	OLE			Nama pola masukan
Output	Object	8		Pola masukan berupa file gambar
Status	Text Yes/No			Kode vector sebagai lapisan output Status telah dilatih atau tidak

### 3.2 Perancangan Proses Pelatihan (Learning)

Pada proses pelatihan atau *learning* program akan memanggil data-data berupa sidik jari utuh (belum terdistorsi) yang telah disimpan dalam tabel tInput. Kemudian pola tersebut akan dilatih oleh JST BAM dengan tujuan agar JST memiliki pengetahuan yang cukup dalam mengenali pola-pola sidik jari baik yang terdistorsi maupun yang tidak terdistorsi. Pada proses pelatihan inilah diperoleh matriks bobot sesuai dengan yang telah dirumuskan dalam JST BAM.

Proses pelatihan diawali dengan memilih file gambar (.bmp) sebagai input. Kemudian gambar tersebut akan diubah menjadi monokrom dilanjutkan dengan proses mengubah gambar ke bentuk *bipolar*, yaitu dengan aturan apabila warna hitam maka nilainya adalah 1 sedangkan untuk warna putih nilainya adalah -1. Bentuk bipolar ini kemudian dikalikan dengan kode vektor yang disimpan sebagai lapisan output untuk mendapatkan matriks bobot yang telah tersimpan sebelumnya (jika proses pelatihan baru dilakukan pertama kali, nilai matriks bobot sama dengan 0). Karena itu, sebelum proses menghitung matriks bobot,

terlebih dahulu dilakukan proses membuka file teks yang berisi nilai matriks bobot yang telah tersimpan sebelumnya yaitu yang tersimpan dalam *Matrixbobot.txt*. Setelah itu, baru dilakukan proses mengubah matriks bobot, yaitu dengan menambahkan matriks bobot yang disimpan dalam file dengan matriks bobot yang didapatkan dari gambar baru yang akan dilatih. Matriks bobot yang telah diubah akan disimpan kembali dalam file *MatrixBobot.txt*. Status pola yang disimpan dalam database berubah menjadi 'True', sehingga dapat diketahui jumlah pola yang telah dilatih. Dengan demikian proses pelatihan untuk satu file gambar telah selesai. Apabila akan dilakukan proses pelatihan kembali, maka proses kembali pada proses memilih file gambar.

### 3.3 Perancangan Proses Pengujian (Testing)

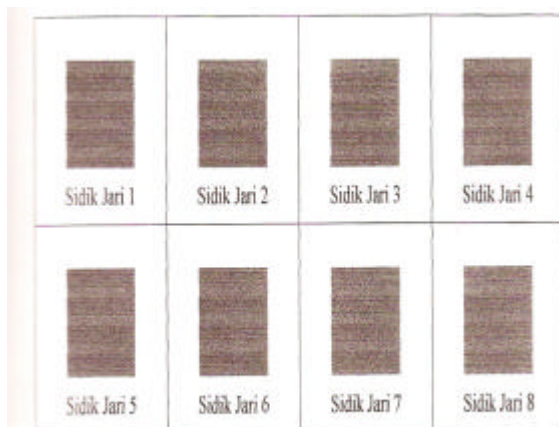
Pada proses pengujian JST BAM, dilakukan proses pengambilan matriks bobot yang tersimpan sebelumnya, kemudian setelah dihitung dengan matriks input pola baik yang terdistorsi maupun tidak terdistorsi, dapat diketahui apakah pola tersebut dapat dikenali atau tidak. Serta seberapa besar *noise* yang dapat diberikan hingga JST BAM tidak dapat mengenali pola dengan tepat.

Proses diawali dengan memiliki file gambar (.bmp) sebagai masukan, kemudian mengubahnya menjadi monokrom dilanjutkan dengan mengubah gambar ke bentuk *bipolar*, yaitu dengan aturan apabila warna hitam maka nilainya 1 sedangkan untuk warna putih nilainya -1. Selanjutnya dilakukan proses membuka file *MatriksBobot.txt* untuk mengetahui nilai matriks bobot yang tersimpan sebelumnya. Dengan algoritma pada JST BAM kemudian dilakukan perhitungan untuk mencari bentuk vektor sesuai dengan target yang diharapkan yaitu seperti pada lapisan output. Setelah kondisi *konvergen* maka akan didapatkan output berupa kode vektor, kemudian akan dibandingkan dengan lapisan output yang tersimpan dalam database. Jika ditemukan kecocokan maka pola tersebut telah dikenali. Sebaliknya, jika tidak ada

kecocokan data maka pola tersebut tidak dapat dikenali dengan tepat.

#### 4. PENGUJIAN PROGRAM DAN PEMBAHASAN

Setelah dilakukan proses pelatihan dengan 8 pola masukan dan proses pengujian dengan pola asli dan terdistorsi, maka dilanjutkan dengan proses pengujian seberapa besar *noise* yang dapat diberikan pada pola masukan hingga JST BAM tidak mampu mengenali pola tersebut dengan tepat. Pada proses ini diambil 8 buah pola yang sudah dilatih kemudian masing-masing pola dikenai *noise* secara random yang besarnya antara 0% sampai dengan 100%. Namun dalam uji coba yang telah dilakukan, pemberian *noise* secara random hanya sampai sebesar 91%. Hal ini disebabkan karena kecepatan aplikasi sangat lambat sehingga tidak memungkinkan untuk dilakukan pemberian *inoise* di atas 91%. Gambar 2 menunjukkan pola sidik jari yang digunakan dalam proses pengujian. Sedangkan hasil pengujian ditunjukkan pada tabel 2 dan tabel 3. Jika output yang dihasilkan tidak sesuai dengan target yang diharapkan berarti JST BAM tidak berhasil mengenali pola dengan sempurna, sebaliknya jika output sama dengan target berarti JST BAM berhasil mengenali pola.



Gambar 2. Pola sidik jari yang digunakan dalam proses pengujian

Tabel 2. Hasil pengujian pola dengan *noise* berkisar 0 – 50% .

Nama Pola Sidik Jari	Persentase <i>noise</i> yang diberikan					
	0%	10%	20%	30%	40%	50%
Sidik Jari 1	Sidik Jari 1	Sidik Jari 1	Sidik Jari 1	Sidik Jari 1	Sidik Jari 1	-
Sidik Jari 2	Sidik Jari 2	Sidik Jari 2	Sidik Jari 2	Sidik Jari 2	Sidik Jari 2	-
Sidik Jari 3	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3
Sidik Jari 4	Sidik Jari 4	Sidik Jari 4	Sidik Jari 4	Sidik Jari 4	Sidik Jari 4	Sidik Jari 3
Sidik Jari 5	Sidik Jari 5	Sidik Jari 5	Sidik Jari 5	Sidik Jari 5	Sidik Jari 1	Sidik Jari 4
Sidik Jari 6	Sidik Jari 6	Sidik Jari 6	Sidik Jari 6	-	-	Sidik Jari 1
Sidik Jari 7	Sidik Jari 7	Sidik Jari 7	Sidik Jari 7	Sidik Jari 7	Sidik Jari 7	Sidik Jari 1
Sidik Jari 8	Sidik Jari 8	Sidik Jari 8	Sidik Jari 8	Sidik Jari 8	Sidik Jari 4	Sidik Jari 1

Tabel 3 Hasil pengujian pola dengan *noise* berkisar 60 – 91% .

Nama Pola Sidik Jari	Persentase <i>noise</i> yang diberikan				
	60%	70%	80%	90%	100%
Sidik Jari 1	Sidik Jari 8	-	-	-	-
Sidik Jari 2	Sidik Jari 5	Sidik Jari 5	Sidik Jari 5	Sidik Jari 5	Sidik Jari 5
Sidik Jari 3	Sidik Jari 6	Sidik Jari 6	Sidik Jari 6	Sidik Jari 6	Sidik Jari 6
Sidik Jari 4	Sidik Jari 5	-	-	-	-
Sidik Jari 5	Sidik Jari 2	Sidik Jari 2	Sidik Jari 2	Sidik Jari 2	Sidik Jari 2
Sidik Jari 6	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3	Sidik Jari 3
Sidik Jari 7	Sidik Jari 4	Sidik Jari 8	Sidik Jari 8	Sidik Jari 8	Sidik Jari 8
Sidik Jari 8	Sidik Jari 7	Sidik Jari 7	Sidik Jari 7	Sidik Jari 7	Sidik Jari 7

Dari hasil pengujian di atas, dapat dilihat bahwa JST BAM dapat mengenali pola sidik jari dengan tepat setelah pola dikenai *noise* berkisar antara 0 – 30%,

kecuali satu pola yaitu pola sidik jari 6. Sedangkan di atas 30%, yaitu sebesar 40 – 91%, beberapa pola sidik jari ada yang tidak dapat dikenali dengan tepat bahkan ada pula yang tidak bisa dikenali sama sekali, seperti pada pola sidik jari 1, 2, 4 dan 6. Hal tersebut dikarenakan output yang dihasilkan JST BAM bukan merupakan salah satu pola yang dilatih sebelumnya atau pola yang dihasilkan konvergen dengan pola terlatih yang lain.

## 5. KESIMPULAN DAN SARAN

### Kesimpulan

Dari hasil percobaan pembuatan dan pengujian program, terdapat beberapa kesimpulan yang dapat diambil antara lain :

1. Jaringan Syaraf Tiruan Bidirectional Associative Memory dapat digunakan dalam proses pengenalan pola, dalam penelitian ini pola yang digunakan adalah pola sidik jari.
2. Dari hasil pengujian, menunjukkan bahwa JST BAM tidak dapat mengenali secara tepat beberapa pola sidik jari yang dikenai *noise* antara 40% - 91%. Output yang dihasilkan JST BAM berupa pola palsu atau output yang dihasilkan bukan merupakan salah satu dari pola sidik jari yang telah dilatih sebelumnya.
3. Selain itu pada kondisi tertentu, output yang dihasilkan JST BAM dengan *noise* antara 40% - 91% konvergen dengan pola terlatih lainnya. Hal tersebut merupakan keterbatasan JST BAM yang disebabkan karena input yang diberikan sebagai pelatihan memiliki banyak kesamaan, sehingga JST BAM menjadi tidak stabil.

### Saran

1. Dalam penelitian ini output yang diharapkan menggunakan matriks dengan 4 vektor. Untuk data yang lebih

banyak dapat menggunakan ukuran matriks yang lebih besar. Hal ini tentunya dapat mempengaruhi kecepatan program.

2. Menambah format gambar yang digunakan sebagai pola masukan seperti GIF, JPG, TIF, dan lain-lain..
3. Khusus untuk masalah pencocokan sidik jari, idealnya percobaan dan pengujiannya menggunakan *sample* sidik jari yang didapatkan langsung dengan *scanner*.

## DAFTAR PUSTAKA

- Castleman, K.R., 1996. *Digital Image Processing*, Prentice-Hall Inc., New Jersey.
- Fausett, J.A., 1994. *Fundamentals of Neural Network, Architecture, Algorithm and Applications*, Prentice-Hall Inc, New Jersey.
- Freeman, J.A., Skapura, D.M., 1991. *Algorithms, Applications and Programming Techniques*, Addison Wesley, New York.
- Jogiyanto, H.M, 2000. *Pengenalan Komputer, Dasar Ilmu Komputer, Pemrograman, Sistem Informasi dan Inteleksi Buatan*, Andi Offset, Yogyakarta.
- Kusumadewi, S, 2003. *Artificial intelligence*, Graha Ilmu, Yogyakarta.
- Setiawan, S, 1993. *Artificial Intelligence*, Andi Offset, Yogyakarta.
- Shi, H, Zhao, Y, Zhuang, X, 1998. A General Model for Bidirectional Associative Memory, *IEEE Transactions On Systems, man, And Cybernetics*, Vol. XX, No. Y, August, 1998.
- Tim Penelitian dan Pengembangan Wahana Komputer, 2001. *Tip dan Pemrograman Visual Basic 6.0*, Andi Offset, Yogyakarta.