PENERAPAN ALGORITMA FUNGSI *BOOLEAN* SEBAGAI EKSTRASI CIRI PADA JARINGAN SYARAF TIRUAN PROPAGASI BALIK UNTUK PENGENALAN SIMBOL PETA

Helmy Thendean
Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara
Jl. Letjen S. Parman No.1 Jakarta Barat 11440
Tlp. (021)5676260, 5677949 ext.114
E-mail: helmy.fti.untar@gmail.com, helmy@fti.utara.org

ABSTRAK: Penerapan Jaringan Syaraf Tiruan (JST) pada masalah klasifikasi citra sering membutuhkan waktu proses pembelajaran yang cukup lama yang disebabkan oleh besarnya jumlah data *input*. Penggunaan data seperti ini dikenal sebagai penggunaan data yang bersifat mentah. Data *input* sebuah citra diambil secara langsung untuk diberikan sebagai data pembelajaran dalam algoritma JST. Permasalahan ini menimbulkan berbagai metode ekstrasi ciri (*feature extraction*) yang diterapkan pada sebuah citra sebelum dijadikan sebagai data pembelajaran dalam sebuah algoritma JST. Algoritma Fungsi Boolean (AFB) menunjukkan bahwa JST dengan penerapan AFB memiliki nilai *output* yang bersifat identik. Sifat identik pada hasil pemrosesan sebuah data citra inilah yang kemudian diterapkan pada algoritma JST Propagasi Balik (JST PB) sebagai algoritma JST yang terbaik untuk permasalahan klasifikasi. Hasil percobaan menunjukkan bahwa JST AFB dapat diterapkan sebagai salah satu metode *feature extraction* pada citra. Kinerja JST AFB dengan pendekatan kedua memberikan hasil yang lebih baik dari tingkat pengenalan maupun waktu pembelajaran.

Kata kunci: jaringan syaraf tiruan, algoritma fungsi Boolean, propagasi balik, ekstrasi ciri, pengenalan pola.

***ABSTRACT**: Artificial neural networks (ANN) implementation in image classification problem requires a lot of training time which caused by enormous data size. This kind of data is known as raw data. An image data is extracted directly without any preprocessing. Many feature extraction techniques are offered to reduce the time consumed in training image data. Boolean function algorithm (BFA) in ANN (first and second approach) produces identical output values that can be used as feature of an image data. This feature can then be used as input in Back Propagation ANN that is known as the best ANN method in problems classification. The experiment shows that BFA can be used well as one of feature extraction methods. The second approach BFA in Back Propagation ANN shows better performance in recognition and is lesser time consuming in training than the first approach.*

***Keywords**: artificial neural networks, Boolean function algorithm, back propagation method, feature extraction, pattern recognition*

PENDAHULUAN

Jaringan Syaraf Tiruan (JST) telah diketahui memiliki kemampuan yang sangat baik untuk melakukan berbagai proses klasifikasi. Salah satu klasifikasi yang sering dilakukan dengan menerapkan JST adalah pada permasalahan klasifikasi citra. Penerapan JST pada masalah klasifikasi citra sering membutuhkan waktu proses pembelajaran yang cukup lama yang disebabkan oleh besarnya jumlah data *input*. Penggunaan data seperti ini dikenal sebagai penggunaan data yang bersifat mentah. Data *input* sebuah citra diambil secara langsung untuk diberikan sebagai data pembelajaran dalam algoritma JST. Permasalahan ini menimbulkan berbagai metode ekstrasi ciri (*feature extraction*) yang diterapkan pada sebuah citra sebelum dijadikan sebagai data pembelajaran dalam sebuah algoritma JST [1, 2, 7].

Penelitian JST dengan menerapkan Algoritma Fungsi Boolean (AFB) [4] dan [8] menunjukkan bahwa JST dengan penerapan AFB memiliki nilai *output* yang bersifat identik. Sifat identik pada hasil pemrosesan sebuah data citra inilah yang mendorong timbulnya keinginan untuk mengetahui akurasi klasifikasi sebuah JST apabila diterapkan "algoritma JST dengan penerapan AFB" sebagai salah satu bentuk ekstrasi ciri pada citra. Algoritma JST yang digunakan sebagai dasar pengujian adalah algoritma JST Propagasi Balik (PB) yang sudah cukup teruji dalam permasalahan klasifikasi [3, 6].

JARINGAN SYARAF TIRUAN

JST merupakan salah satu bagian dari metode dalam bidang *Artificial Intelligence* yang dikenal sebagai *machine learning* [6]. *Machine leaarning*

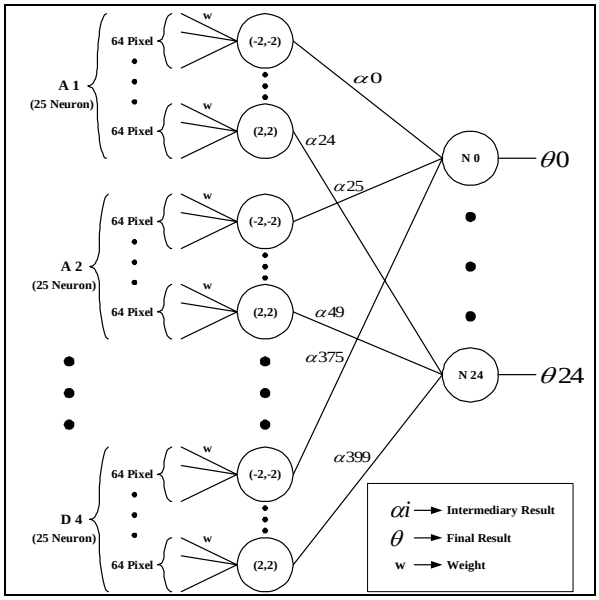
merupakan metode dalam ilmu komputer yang melibatkan mekanisme adaptasi sehingga mesin (komputer) dapat belajar dari pengalaman.

Secara umum, JST dapat digunakan sebagai salah satu metode untuk melakukan pengenalan pola pada sebuah citra [5]. Penggunaan JST untuk pengenalan pola pada sebuah citra ini erat kaitannya dengan permasalahan klasifikasi obyek yang sering muncul dalam kehidupan sehari-hari. Secara umum masalah pengenalan pola ini termasuk dalam bidang *computer vision* yang bertujuan untuk mengambil informasi dari citra atau data yang berdimensi banyak dengan menggunakan mesin [9]. Permasalahan dalam proses pengenalan pola adalah bagaimana sebuah data citra dapat dikenali sebagai dirinya atau diklasifikasikan dalam kelompok dengan citra tersebut sebagai anggota di dalamnya.

JST AFB

Pada penelitian tentang AFB [4, 8], terdapat dua arsitektur JST yang diajukan untuk proses pengenalan. Arsitektur dengan pendekatan pertama menghasilkan kinerja pengenalan yang relatif lebih buruk dibandingkan dengan arsitektur pendekatan kedua. Dari hasil pengamatan disebutkan bahwa kedua pendekatan tersebut sama-sama menghasilkan nilai yang bersifat identik pada lapisan keluaran untuk setiap citra yang dijadikan data pembelajaran.

Pendekatan pertama terdiri dari lapisan masukan sebanyak 400 neuron dengan setiap neuron akan menerima *input* 64 pixel. Lapisan masukan akan menghasilkan 400 keluaran sebagai *input* pada 25 neuron pada lapisan keluaran. Arsitektur Algoritma Fungsi Boolean dengan pendekatan pertama dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Algoritma Fungsi Boolean Pendekatan Pertama [8]

Proses pembelajaran dilakukan terhadap setiap *train set* untuk memperoleh θ_i dengan:

$$\theta_0 = (\alpha_0 + \alpha_{25} + \alpha_{50} + ... + \alpha_{375}) \quad (1)$$

$$\alpha_0 = \sum_{0}^{63} (w_{a1_{(-2,-2)}} \times A_{1_{(-2,-2)}}) \quad (2)$$

$$\alpha_{375} = \sum_{0}^{63} (w_{d4_{(-2,-2)}} \times D_{4_{(-2,-2)}}) \quad (3)$$

Untuk proses pengenalan, pola yang akan dikenali dicocokkan satu per satu dengan setiap *train set* dan dilakukan langkah-langkah sebagai berikut:

1. Hitung hasil keluaran P_i.

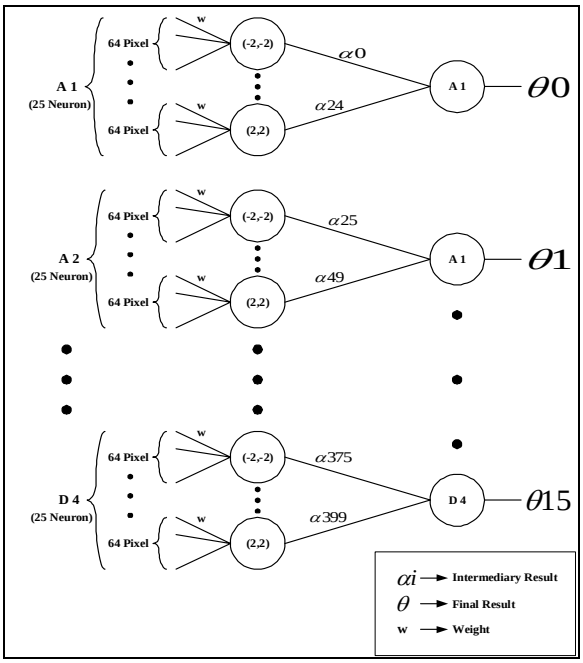
$$P_0 = (\beta_0 + \beta_{25} + \beta_{50} + ... + \beta_{375}) \quad (4)$$

$$\beta_0 = \sum_{0}^{63} (w^{"ENT"}_{a1_{(-2,-2)}} \times A^{"TEST"}_{1_{(-2,-2)}}) \quad (5)$$

$$\beta_{375} = \sum_{0}^{63} (w^{"ENT"}_{d4_{(-2,-2)}} \times D^{"TEST"}_{4_{(-2,-2)}}) \quad (6)$$

2. Bandingkan θ_i dan P_i dan hitung jumlah nilai $\theta_i = P_i$ dalam K.
3. Jika K>19 maka pola dikenali dan jika tidak maka pola tidak dikenali.

Pendekatan kedua terdiri dari lapisan masukan sebanyak 400 neuron dengan setiap neuron akan menerima *input* 64 pixel. Lapisan masukan akan menghasilkan 400 keluaran sebagai *input* pada 16 neuron pada lapisan keluaran. Arsitektur Algoritma Fungsi Boolean dengan pendekatan kedua dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur Algoritma Fungsi Boolean Pendekatan Kedua [8]

Proses pembelajaran dilakukan terhadap setiap *train set* untuk memperoleh θ_i dengan:

$$\theta_0 = \sum_{i=0}^{24} \alpha_i \quad (7)$$

$$\alpha_0 = \sum_{0}^{63} (w_{a1_{(-2,-2)}} \times A_{1_{(-2,-2)}}) \quad (8)$$

$$\alpha_{24} = \sum_{0}^{63} (w_{a1_{(2,2)}} \times A_{1_{(2,2)}}) \quad (9)$$

Untuk proses pengenalan, pola yang akan dikenali dicocokkan satu per satu dengan setiap *train set* dan dilakukan langkah-langkah sebagai berikut:

1. Hitung hasil keluaran P_i.

$$P_0 = \sum_{i=0}^{24} \beta_i \quad (10)$$

$$\beta_0 = \sum_{0}^{63} (w^{"ENT"}_{a1_{(-2,-2)}} \times A^{"TEST"}_{1_{(-2,-2)}}) \quad (11)$$

$$\beta_{24} = \sum_{0}^{63} (w^{"ENT"}_{a1_{(2,2)}} \times A^{"TEST"}_{1_{(2,2)}}) \quad (12)$$

2. Bandingkan θ_i dan P_i dan hitung jumlah nilai $\theta_i = P_i$ dalam K.
3. Jika K>12 maka pola dikenali dan jika tidak maka pola tidak dikenali

OBYEK PENELITIAN

Obyek penelitian berupa citra biner yang menampilkan simbol peta yang berukuran 32x32 *pixel*. Simbol peta yang digunakan adalah simbol umum pada peta Jogja. Untuk membatasi ruang lingkup penelitian, simbol peta yang dipakai terdiri dari lima jenis yaitu Airport, Tugu, Borobudur, Sultan Palace dan Bus Station. Borobudur, Bus Station dan Sultan Palace mewakili simbol peta yang didominasi oleh obyek di dalamnya. Airport dan tugu mewakili simbol peta yang didominasi oleh *background*.

PERCOBAAN

Dalam penelitian ini arsitektur JST PB yang akan digunakan akan disesuaikan dengan kedua jenis *feature vector* yang dihasilkan oleh JST AFB, masing-masing 24 *node input* untuk pendekatan pertama dan 15 *node input* untuk pendekatan kedua. Jumlah *hidden layer* yang merupakan lapisan perantara antara lapisan masukan dan keluaran adalah satu. Jumlah node pada *hidden layer* akan diperoleh berdasarkan percobaan yang dilakukan untuk memperoleh kinerja pengenalan terbaik. Jumlah *node* pada lapisan keluaran sebanyak jenis simbol peta yang akan dipakai sebagai contoh data.

Tabel 1. Contoh Hasil JST AFB Pendekatan Pertama

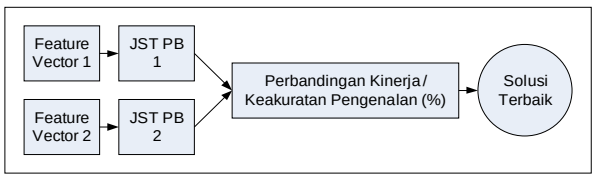
					
Tetha[0]= 48	Tetha[0]= 127	Tetha[0]= 102	Tetha[0]= 102	Tetha[0]= 101	Tetha[0]= 103
Tetha[1]= 42	Tetha[1]= 118	Tetha[1]= 94	Tetha[1]= 94	Tetha[1]= 93	Tetha[1]= 93
Tetha[2]= 37	Tetha[2]= 110	Tetha[2]= 84	Tetha[2]= 86	Tetha[2]= 85	Tetha[2]= 86
Tetha[3]= 22	Tetha[3]= 98	Tetha[3]= 75	Tetha[3]= 74	Tetha[3]= 90	Tetha[3]= 90
Tetha[4]= 252	Tetha[4]= 70	Tetha[4]= 46	Tetha[4]= 46	Tetha[4]= 67	Tetha[4]= 67
Tetha[5]= 19	Tetha[5]= 94	Tetha[5]= 70	Tetha[5]= 70	Tetha[5]= 69	Tetha[5]= 69
Tetha[6]= 13	Tetha[6]= 87	Tetha[6]= 65	Tetha[6]= 63	Tetha[6]= 62	Tetha[6]= 62
Tetha[7]= 8	Tetha[7]= 80	Tetha[7]= 55	Tetha[7]= 56	Tetha[7]= 55	Tetha[7]= 55
Tetha[8]= 249	Tetha[8]= 67	Tetha[8]= 45	Tetha[8]= 43	Tetha[8]= 59	Tetha[8]= 60
Tetha[9]= 224	Tetha[9]= 40	Tetha[9]= 17	Tetha[9]= 16	Tetha[9]= 37	Tetha[9]= 37
Tetha[10]= 246	Tetha[10]= 62	Tetha[10]= 39	Tetha[10]= 38	Tetha[10]= 37	Tetha[10]= 38
Tetha[11]= 241	Tetha[11]= 56	Tetha[11]= 34	Tetha[11]= 32	Tetha[11]= 31	Tetha[11]= 33
Tetha[12]= 236	Tetha[12]= 50	Tetha[12]= 26	Tetha[12]= 26	Tetha[12]= 25	Tetha[12]= 24
Tetha[13]= 221	Tetha[13]= 36	Tetha[13]= 14	Tetha[13]= 12	Tetha[13]= 28	Tetha[13]= 32
Tetha[14]= 196	Tetha[14]= 10	Tetha[14]= 244	Tetha[14]= 242	Tetha[14]= 7	Tetha[14]= 9
Tetha[15]= 18	Tetha[15]= 94	Tetha[15]= 71	Tetha[15]= 70	Tetha[15]= 69	Tetha[15]= 69
Tetha[16]= 12	Tetha[16]= 87	Tetha[16]= 64	Tetha[16]= 63	Tetha[16]= 62	Tetha[16]= 62
Tetha[17]= 8	Tetha[17]= 80	Tetha[17]= 55	Tetha[17]= 56	Tetha[17]= 55	Tetha[17]= 56
Tetha[18]= 248	Tetha[18]= 67	Tetha[18]= 45	Tetha[18]= 43	Tetha[18]= 59	Tetha[18]= 59
Tetha[19]= 223	Tetha[19]= 40	Tetha[19]= 17	Tetha[19]= 16	Tetha[19]= 37	Tetha[19]= 37
Tetha[20]= 244	Tetha[20]= 62	Tetha[20]= 39	Tetha[20]= 38	Tetha[20]= 37	Tetha[20]= 37
Tetha[21]= 239	Tetha[21]= 56	Tetha[21]= 33	Tetha[21]= 32	Tetha[21]= 31	Tetha[21]= 31
Tetha[22]= 235	Tetha[22]= 50	Tetha[22]= 25	Tetha[22]= 26	Tetha[22]= 25	Tetha[22]= 26
Tetha[23]= 219	Tetha[23]= 36	Tetha[23]= 14	Tetha[23]= 12	Tetha[23]= 28	Tetha[23]= 28
Tetha[24]= 194	Tetha[24]= 10	Tetha[24]= 243	Tetha[24]= 242	Tetha[24]= 7	Tetha[24]= 7

Tabel 2. Contoh Hasil JST AFB Pendekatan Kedua

					
Tetha[0]= 190	Tetha[0]= 89	Tetha[0]= 39	Tetha[0]= 89	Tetha[0]= 80	Tetha[0]= 68
Tetha[1]= 36	Tetha[1]= 56	Tetha[1]= 69	Tetha[1]= 56	Tetha[1]= 17	Tetha[1]= 219
Tetha[2]= 21	Tetha[2]= 41	Tetha[2]= 63	Tetha[2]= 56	Tetha[2]= 44	Tetha[2]= 15
Tetha[3]= 190	Tetha[3]= 89	Tetha[3]= 49	Tetha[3]= 89	Tetha[3]= 170	Tetha[3]= 14
Tetha[4]= 127	Tetha[4]= 147	Tetha[4]= 127	Tetha[4]= 147	Tetha[4]= 188	Tetha[4]= 195
Tetha[5]= 232	Tetha[5]= 241	Tetha[5]= 95	Tetha[5]= 78	Tetha[5]= 238	Tetha[5]= 177
Tetha[6]= 222	Tetha[6]= 231	Tetha[6]= 82	Tetha[6]= 74	Tetha[6]= 177	Tetha[6]= 254
Tetha[7]= 127	Tetha[7]= 147	Tetha[7]= 127	Tetha[7]= 147	Tetha[7]= 148	Tetha[7]= 135
Tetha[8]= 139	Tetha[8]= 159	Tetha[8]= 161	Tetha[8]= 139	Tetha[8]= 27	Tetha[8]= 115
Tetha[9]= 252	Tetha[9]= 191	Tetha[9]= 102	Tetha[9]= 79	Tetha[9]= 70	Tetha[9]= 115
Tetha[10]= 1	Tetha[10]= 196	Tetha[10]= 93	Tetha[10]= 78	Tetha[10]= 24	Tetha[10]= 53
Tetha[11]= 154	Tetha[11]= 179	Tetha[11]= 152	Tetha[11]= 159	Tetha[11]= 53	Tetha[11]= 137
Tetha[12]= 192	Tetha[12]= 91	Tetha[12]= 63	Tetha[12]= 61	Tetha[12]= 101	Tetha[12]= 197
Tetha[13]= 105	Tetha[13]= 126	Tetha[13]= 183	Tetha[13]= 145	Tetha[13]= 140	Tetha[13]= 222
Tetha[14]= 105	Tetha[14]= 125	Tetha[14]= 161	Tetha[14]= 140	Tetha[14]= 200	Tetha[14]= 193
Tetha[15]= 65	Tetha[15]= 91	Tetha[15]= 50	Tetha[15]= 61	Tetha[15]= 91	Tetha[15]= 185

Untuk menentukan apakah JST AFB dapat menghasilkan ekstrasi ciri yang baik terhadap suatu citra maka akan dilakukan percobaan terhadap hasil ekstrasi ciri dari AFB dengan pendekatan pertama dan kedua. Percobaan dilakukan dengan memberikan *feature vector* dari kedua pendekatan AFB tersebut ke JST PB. Data untuk percobaan dibagi dalam dua jenis, yaitu *training set* dan *test set*. Contoh data hasil dari JST AFB pendekatan pertama dan kedua dapat dilihat pada Tabel 1 dan Tabel 2.

Jumlah *train set* yang direncanakan adalah lima (Sultan Palace, Airport, Tugu, Bus Station dan Candi Borobudur), masing-masing mewakili simbol peta yang dijadikan obyek percobaan. Dari hasil percobaan akan dilihat seberapa akurat JST PB dapat mengenali citra yang berasal dari *test set*. Proses penentuan solusi terbaik dari kedua pendekatan tersebut dapat dilihat pada Gambar 3.



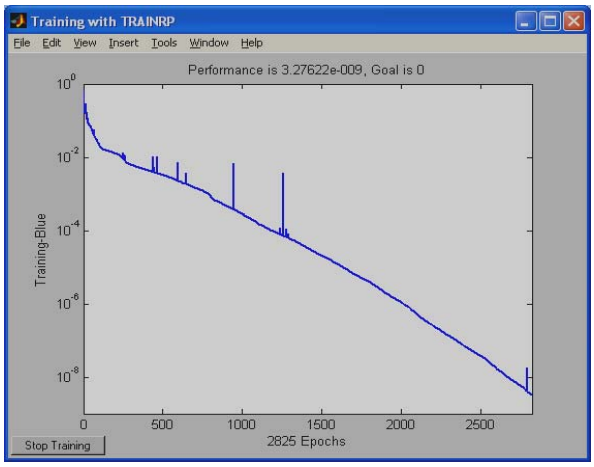
Gambar 3. Diagram Balok Proses Penentuan Solusi Terbaik

HASIL PERCOBAAN DAN PEMBAHASAN

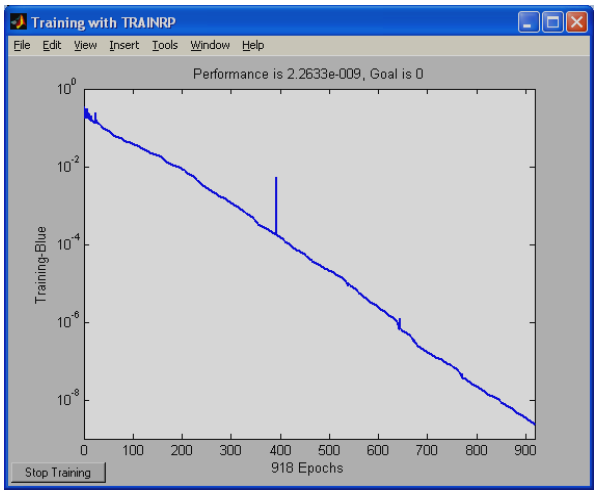
Percobaan pada masing-masing JST PB dimulai dengan menggunakan 1 lapisan tersembunyi dan 10 neuron di dalamnya. Adanya pertimbangan untuk mempercepat proses pembelajaran maka *learning rate* dirancang berubah-ubah dalam setiap *epoch* yang dilakukan oleh JST PB. Perubahan ini berdasarkan *error rate* yang terjadi pada setiap *epoch*. Bila *error rate* saat ini lebih besar dibandingkan dengan *error rate* pada *epoch* sebelumnya maka *learning rate* diturunkan nilainya. Sebaliknya bila *error rate* saat ini lebih kecil dibandingkan dengan *error rate* sebelumnya maka *learning rate* akan dinaikkan nilainya. Grafik kinerja JST PB 1 dan 2 dapat dilihat pada Gambar 4 dan Gambar 5.

Dari grafik hasil percobaan dapat dilihat bahwa proses pembelajaran JST PB1 berhenti pada saat *epoch* ke 2825 dan error rate $3{,}28.10^{-9}$ sedangkan JST PB 2 berhenti pada saat *epoch* ke 918 dan error rate $2{,}26.10^{-9}$. Hal ini menunjukkan bahwa JST PB 2 jauh lebih cepat dalam melakukan proses pembelajaran karena disebabkan *feature vector* yang berjumlah lebih sedikit.

Pada saat pengenalan dengan menggunakan Test Set, diperoleh bahwa hasil *feature vector* yang diperoleh dengan JST AFB 2 memberikan tingkat keakuratan yang lebih baik yaitu sebesar 73.33 % dibandingkan dengan JST PB 2 yang hanya sebesar 50 %. JST PB 1 gagal total untuk mengenali citra *Bus Station* dan *Sultan Palace*. JST PB 2 mampu mengenali setiap jenis data citra walaupun pada beberapa citra yang mengandung *noise* masih terdapat kegagalan pengenalan. Hasil pengenalan dapat dilihat pada Tabel 3.



Gambar 4. Kinerja JST PB 1



Gambar 5. Kinerja JST PB 2

Tabel 3. Hasil Pengenalan

	Citra	Tipe	JST PB1	JST PB2
	Airport	1	Dikenali	Gagal
		2	Dikenali	Dikenali
		3	Dikenali	Dikenali
		4	Dikenali	Dikenali
		5	Dikenali	Dikenali
		6	Dikenali	Gagal
	Borobudur	1	Dikenali	Dikenali
		2	Dikenali	Dikenali
		3	Dikenali	Dikenali
		4	Dikenali	Dikenali
		5	Dikenali	Dikenali
		6	Dikenali	Dikenali
	Bus Station	1	Gagal	Gagal
		2	Gagal	Dikenali
		3	Gagal	Gagal
		4	Gagal	Dikenali
		5	Gagal	Dikenali
		6	Gagal	Gagal

Tabel 3. Hasil Pengenalan (lanutan)

	Citra	Tipe	JST PB1	JST PB2
	Sultan Palace	1	Gagal	Gagal
		2	Gagal	Dikenali
		3	Gagal	Dikenali
		4	Gagal	Dikenali
		5	Gagal	Gagal
		6	Gagal	Gagal
	Tugu	1	Dikenali	Dikenali
		2	Dikenali	Dikenali
		3	Dikenali	Dikenali
		4	Dikenali	Dikenali
		5	Dikenali	Dikenali
		6	Dikenali	Dikenali

Keterangan :
1 : Citra akibat perpotongan garis hitam
2 : Citra akibat Perpotongan garis putih
3 : Citra akibat *noise* bertipe *salt & pepper*
4 : Citra asli
5 : Citra akibat rotasi +10°
6 : Citra akibat rotasi -10°

KESIMPULAN

Hasil percobaan di atas dapat ditarik kesimpulan sebagai berikut:

1. JST AFB dapat diterapkan sebagai salah satu metode *feature extraction* pada citra.
2. Kinerja JST PB 2 pada saat melakukan pembelajaran jauh lebih efisien karena membutuhkan jumlah *epoch* yang jauh lebih kecil untuk mencapai error rate yang diizinkan.
3. Kinerja JST PB 2 lebih baik pada saat pengenalan terhadap data lain yang berbeda dengan data pembelajaran. Setiap jenis data citra dapat dikenali walaupun gagal untuk pengenalan beberapa data citra yang mengandung *noise*.

DAFTAR PUSTAKA

1. Chen, Yuehui., Jiang, Shuyan. & Abraham, Ajith., *Face Recognition Using DCT and Hybrid Flexible Neural Tree*, http://cilab.ujn.edu.cn/paper/338_26_16_510.pdf, akses Maret 2006.

2. Chikkerur, Sharat., Wu, Chaohang. & Govindaraju, Venu., *A Systematic Approach for Feature Extraction in Fingerprint Images*, http://www.cubs.buffalo.edu/pdf/finger-systematic.pdf, akses Maret 2006.

3. Fausett, Lauren., *Fundamentals of Neural Networks–Architectures, Algorithms, and Applications*, Englewood Cliffs, NJ : Prentice Hall, 1994.

4. Hadjar, Karim. & Khalfallah, Majed., *Improvement of Pattern Recognition inthe Field of Mapping Using Neural Networks*, Proceeding IASTED, Mexico, 1998.

5. Menendez, Anne., *Image Recognition Using Neural Networks*, <http//www.general-vision.com/publications/WP_Image%20recognition%20using%20neural%20networks.pdf>, akses Februari 2006.

6. Negnevitsky, Michael., *Artificial Intelligence: A Guide to Intelligent System*, Boston: PWS-KENT, 2002.

7. Sorwar, Golam. & Abraham, Ajith., *DCT Based Texture Classification Using Soft Computing Approach*, <http://arxiv.org/pdf/CS.AI/0405013>, akses Agustus 2006.

8. Thendean, Helmy. & Prijodiprodjo, Widodo., *Metode Pembangkitan Bobot untuk Jaringan Syaraf Tiruan dengan Algoritma Fungsi Boolean*, Sains dan Sibernatika 2004, Volume 17, No.4.

9. Wikipedia, *Computer Vision*, <http://en.wikipedia.org/wiki/computer_vision>, akses Maret 2006.

.