

## **MODEL IDENTIFIKASI PETA SECARA OTOMATIS MENGUNAKAN KONSEP JARINGAN SARAF TIRUAN BACKPROPAGATION**

**Muhammad Erwin Ashari Haryono**

*Laboratorium Pemrograman dan Informatika Teori  
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia  
Jl. Kaliurang Km. 14 Yogyakarta 55501  
Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148  
E-mail: meah@fti.uii.ac.id*

### **ABSTRACT**

*Artificial Neural Network (ANN) is a model from many models in Artificial Intelligence field. ANN mimics the behaviour of human brain concept of learning. In this model, information will flow and distribute from one neuron to another neuron by using huge of computational process. Backpropagation is the one famous model in ANN. Most researcher in Indonesia using this model for forecasting and deciding data in many field of application like economic, natural science, law, psychology, social etc. The concept of backpropagation is a supervised model, it means inputs and outputs should decide and hold first before trained. Every point of neurons connection from one layer to another layer has a value associating by its weight value. This paper will introduce the behaviour of ANN model to recognize every place of region in map automatically. The case, we will take one sample region map from Daerah Istimewa Yogyakarta Province. After learning and training from 420 data, we have got a stable weight of all neuron connections, than from this ideal weight of connection we will continue testing every map segment by using more than 20 testing data. The result shows that accuracy at 80% of testings with less than 20% failure. The best ANN architecture is found with the following parameters: 0.02 learning factor, 0.01 threshold failure, 10 hidden units, and one layer of hidden layer.*

*Keywords : backpropagation, Artificial Neural Network, learning factor, threshold failure*

### **1. PENDAHULUAN**

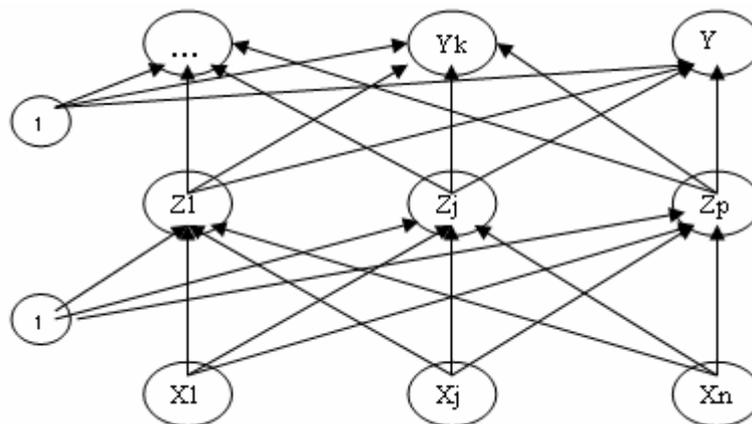
Era teknologi informasi semakin berkembang dengan cepat dan kompleks, kehandalan sistem yang mengolah data dengan baik akan menghasilkan informasi yang baik, begitu pula dengan pemakai sistem yang merupakan faktor utama suatu aplikasi sistem yang sedang dijalankan selain perangkat keras dan perangkat lunak. *Brainware* adalah konsep pemikiran yang dimiliki oleh manusia dalam kaitannya dengan sistem komputer secara luas. Konsep pemikiran manusia yang dapat melakukan pembelajaran terhadap segala sesuatu yang ditemuinya merupakan hal menarik yang menjadi landasan dibentuknya model Jaringan saraf tiruan (JST) dalam domain yang cukup bervariasi dalam sistem cerdas. Sistem

cerdas yang diaplikasikan untuk identifikasi objek dua dimensi pada pemetaan wilayah geografis merupakan sistem yang dapat memberikan pengetahuan dan informasi mengenai keadaan wilayah pada suatu tempat atau daerah yang diinginkan pemakai. Sistem yang dibangun dapat menentukan lokasi wilayah, objek wisata, dan hal-hal yang terkait di dalamnya. Pemakai hanya menunjuk bagian tertentu yang diinginkan dalam peta, kemudian sistem akan memberikan informasi tentang wilayah yang diinginkan pemakai tersebut. Penerapan sistem cerdas ini dapat diaplikasikan pada pola pembelajaran siswa dan mahasiswa pada pelajaran geografi dan hal yang menyangkut pemetaan wilayah, maupun tempat-tempat yang strategis, terutama di jantung kota budaya ini.

## 2. JARINGAN SARAF TIRUAN UMPAN BALIK

Banyak model jaringan saraf tiruan dalam dunia riset JST, salah satu modelnya adalah Jaringan saraf tiruan *Backpropagation*. Jaringan saraf tiruan *Backpropagation* (BP) pertama kali diperkenalkan oleh Rumelhart, Hinton dan William pada tahun 1986, kemudian Rumelhart dan Mc Clelland mengembangkannya pada tahun 1988. Jaringan ini merupakan model jaringan *multilayer*. *Backpropagation* paling banyak digunakan oleh pemakai jaringan saraf tiruan, bahkan diperkirakan lebih dari 80 % proyek jaringan saraf tiruan yang tengah dikembangkan menggunakan *Backpropagation*.

Dalam metode *Backpropagation*, biasanya, digunakan jaringan *multilayer*. Sebagai contoh, pada Gambar 1 dilustrasikan jaringan dengan sebuah *hidden layer*. Dalam jaringan, selain terdapat unit *input*, unit tersembunyi (*hidden units*) dan *output* juga terdapat bias yang diberikan pada unit-unit tersembunyi dan *output*.



Gambar 1. Jaringan saraf *Backpropagation* dengan satu *hidden layer*

Pada intinya, pelatihan dengan metode *Backpropagation* terdiri atas tiga langkah, yaitu sebagai berikut.

- a. Data dimasukkan ke *input* jaringan (*feedforward*)
- b. Perhitungan dan propagasi balik dari *error* yang bersangkutan
- c. Pembaharuan (*adjustment*) bobot dan bias.

Saat umpan maju (*feedforward*), setiap unit *input* ( $X_i$ ) akan menerima sinyal *input* dan akan menyebarkan sinyal tersebut pada tiap *hidden unit* ( $Z_j$ ). Setiap *hidden unit* kemudian akan menghitung aktivasinya dan mengirim sinyal ( $z_j$ ) ke tiap unit *output*. Kemudian, setiap unit *output* ( $Y_k$ ) juga akan menghitung aktivasinya ( $y_k$ ) untuk menghasilkan respon terhadap *input* yang diberikan jaringan.

Saat proses pelatihan (*training*) setiap unit *output* membandingkan aktivasinya ( $y_k$ ) dengan nilai target (*desired output*) untuk menentukan besarnya *error*. Berdasarkan *error* tersebut, dihitung faktor  $\delta_k$ . Faktor  $\delta_k$  digunakan untuk mendistribusikan *error* dari *output* kembali ke layer sebelumnya. Dengan cara yang sama, faktor  $\delta_i$  juga dihitung pada *hidden unit*  $Z_j$ . Faktor  $\delta_k$  digunakan untuk memperbaharui bobot antara *hidden layer* dan *input layer*.

Setelah semua faktor  $\delta$  ditentukan, bobot untuk semua layer di sesuaikan secara bersamaan. Pembaharuan bobot  $W_{jk}$  (dari *hidden unit*  $Z_j$  ke unit *output*  $Y_k$ ) dilakukan berdasarkan faktor  $\delta_k$  dan aktivasi  $z_j$  dari *hidden unit*  $Z_j$ . Sedangkan, pembaharuan bobot  $v_{ij}$  (dari *input unit*  $X_i$  ke *hidden unit*  $Z_j$ ) dilakukan berdasarkan faktor  $\delta_j$  dan aktivasi  $x_i$  dari *input*.

Secara detail, langkah-langkah pelatihan *Backpropagation* sebagai berikut:

- Langkah 1. Inisialisasi bobot dan bias  
Baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya, angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif).
- Langkah 2. Jika *stop condition* masih belum terpenuhi, jalankan langkah 3-10
- Langkah 3. Untuk setiap *data training*, lakukan langkah 4-9

#### **Umpan maju (*feedforward*)**

Langkah 4. Setiap unit *input* ( $X_i$ ,  $i=1, \dots, n$ ) menerima sinyal *input*  $x_i$  dan menyebarkan sinyal tersebut pada seluruh unit pada *hidden unit*. Perlu diketahui bahwa *input*  $x_i$  yang dipakai di sini adalah *input training data* yang sudah diskalakan.

Langkah 5. Setiap *hidden unit* ( $Z_j$ ,  $j=1, \dots, p$ ) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasanya,

$$z\_in_j = \sum_{i=1}^n x_i v_{ij} \quad (1)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan.

$$z_j = f(z\_in_j), \quad (2)$$

lalu mengirim sinyal *output* tersebut ke seluruh unit pada unit *output*.

Langkah 6. Setiap unit *output* ( $Y_k, k=1, \dots, m$ ) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasanya,

$$y\_in_k = w_{0k} \sum_{j=1}^p z_{W_{jk}} \quad (3)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit *output* yang bersangkutan

$$y_j = f(y\_in_k), \quad (4)$$

lalu mengirim sinyal *output* ini ke seluruh unit pada unit *output*.

### **Propagasi error (backpropagation of error)**

Langkah 7. Setiap unit *output* ( $Y_k, k = 1, \dots, m$ ) menerima suatu target *pattern* (*desired output*) yang sesuai dengan *input training pattern* untuk menghitung kesalahan (*error*) antara target dengan *output* yang dihasilkan jaringan.

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (5)$$

Sebagaimana *input training data*, *output training data*  $t_k$  juga telah diskalakan menurut fungsi aktivasi yang dipakai.

Faktor  $\delta_k$  digunakan untuk menghitung koreksi *error* ( $\Delta W_{jk}$ ) yang nantinya akan dipakai untuk memperbaharui  $W_{jk}$ , di mana

$$\Delta W_{jk} = \alpha \delta_k z_j \quad (6)$$

selain itu, juga dihitung koreksi bias  $\Delta W_{0k}$  yang nantinya akan dipakai untuk memperbaharui  $W_{0k}$ , di mana:

$$\Delta W_{0k} = \alpha \delta_k \quad (7)$$

Faktor  $\delta_k$  kemudian dikirimkan ke *layer* yang berada pada langkah 8.

Langkah 8. Setiap *hidden unit* ( $Z_j, j = 1, \dots, p$ ) menjumlahkan *input delta* (yang dikirim dari *layer* pada langkah 7) yang sudah berbobot.

$$\delta\_in_j = \sum_{k=1}^m \delta_k W_{jk} \quad (8)$$

Kemudian, hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi *error*  $j$ , di mana

$$\delta_j = \delta\_in_j f'(z\_in_j) \quad (9)$$

Faktor  $\delta_j$  digunakan untuk menghitung koreksi *error* ( $\Delta v_{ij}$ ) yang nantinya akan dipakai untuk memperbaharui  $v_{ij}$ , di mana:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (10)$$

Selain itu, juga dihitung koreksi bias  $\Delta v_{0j}$  yang nantinya akan dipakai untuk memperbaharui  $v_{0j}$ , di mana:

$$\Delta v_{0j} = \alpha \delta_j \quad (11)$$

### **Pembaharuan bobot (*adjustment*) dan bias**

Langkah 9. Setiap unit *output* ( $Y_k$ ,  $k = 1, \dots, m$ ) akan memperbaharui bias dan bobotnya dari setiap *hidden unit* ( $j = 0, \dots, p$ ),

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (12)$$

demikian pula, setiap *hidden unit* ( $Z_j$ ,  $j = 1, \dots, p$ ) akan memperbaharui bias dan bobotnya dari setiap unit *input* ( $i = 0, \dots, n$ )

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \Delta v_{ij} \quad (13)$$

Langkah 10. Memeriksa *stop condition*.

## **3. ANALISIS DAN PERANCANGAN SISTEM**

Perancangan piranti lunak jaringan saraf tiruan didasari oleh model *multilayer perceptron* dengan propagasi balik dan fungsi aktivasi *sigmoid biner* untuk setiap neuronnya. Ide dasar yang dapat dilakukan pada penelitian ini adalah menentukan masukan yang akan dilakukan proses pembelajaran, selain itu karena model yang dikembangkan ini adalah model pembelajaran terawasi (*supervised learning*), maka normalisasi *output* juga harus diberikan pada saat awal pengembangan sistem.

### **3.1 Analisis Masukan dan Keluaran**

Masukan utama adalah citra peta wilayah. Citra peta berwarna kemudian dirubah menjadi gambar dengan skala keabuan, metode yang dipakai untuk menjadikan peta tersebut berwarna *grayscale* adalah konversi yang dikeluarkan oleh *CCIR Recommendation 601-1* yaitu pada persamaan 14 di bawah ini.

$$Y = 0,299 * R + 0,587 * G + 0,114 * B \quad (14)$$

Dengan Y adalah nilai piksel yang baru pada mode *grayscale*, R adalah nilai piksel merah, G adalah nilai piksel hijau, dan B adalah nilai piksel biru

Masukan sistem adalah normalisasi dari segmentasi piksel terhadap wilayah per kecamatan se kotamadya Yogyakarta yang telah dibuat nilai *histogram* terhadap nilai *grayscale*. Gambar dibatasi dengan format BMP dengan jumlah piksel image 776 \* 521 piksel. Segmentasi dilakukan dengan tiga tahap, yaitu

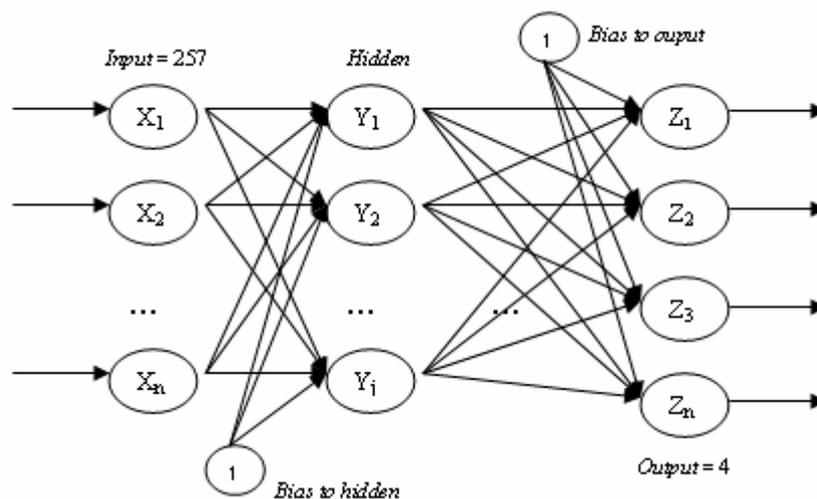
- Segmentasi bujur sangkar, yaitu dengan melakukan pengelompokan piksel dengan jumlah 20 piksel horisontal dan 20 piksel vertikal, sehingga didapat jumlah piksel 400 buah untuk satu segmentasinya.
- Segmentasi horisontal, yaitu dengan melakukan pengelompokan piksel dengan jumlah piksel 20 secara horisontal.
- Segmentasi vertikal, yaitu dengan malakukan pengelompokan piksel dengan jumlah piksel 20 secara vertikal.

Dari pengelompokan tersebut, kemudian di buat frekuensi nilai penyebaran nilai terang dan nilai gelap suatu warna piksel. Sehingga, bila frekuensi warna terang lebih besar dari frekuensi warna gelap akan dinormalisasi menjadi nilai

terang atau 1. Begitu juga sebaliknya. Dari hasil normalisasi tersebut diatas, maka didapat jumlah masukan yang sama untuk seluruh pola wilayah kecamatan yang diberikan (tabel 1) dan jumlah masukan yang akan masuk pada jaringan adalah 257 (jumlah bit biner hasil normalisasi). Karena model *Backpropagation* yang dikembangkan menggunakan model *supervised*, maka nilai keluaran juga harus diberikan dan diidentifikasi dahulu. *Output* adalah nama wilayah kecamatan yang ada di kotamadya Yogyakarta, sehingga jumlah yang didapat adalah 14. Nilai 14 dalam domain *biner* yang dapat terlingkupi menjadi 4 buah digit, sehingga dapat ditentukan nilai biner untuk tiap-tiap kecamatan tersebut. Sehingga (tabel 1).

Tabel 1. Contoh nilai biner input dan nilai biner output pelatihan

No.	Kecamatan	Pola input	Pola output
1	Jetis	100000101101111000111110000...	0001
2	Tegalrejo	100000101101111100111110000...	0010
3	Gondomanan	000000101011111001111100000...	0011
4	Mergangsan	100000101111111100111110000...	0100
5	Wirobrajan	000000101111111001111100000...	0101
6	Mantrijeron	000000101111111100111110000...	0110
7	Umbulharjo	010100100011000000111100000...	0111
8	Kraton	010101100011000000011100000...	1000
9	Pakualaman	000000101111111100111110000...	1001
10	Gondokusuman	000000101111111100011110000...	1010
11	Ngampilan	010101100011000000011100000...	1011
12	Gedung Tengen	010101100011000000011100000...	1100
13	Danurejan	100000101101111000111110000...	1101
14	Kotagede	100000111101111000111010000...	1110



Gambar 2. Rancangan jaringan saraf tiruan untuk pengenalan peta

### 3.2 Perancangan Arsitektur Sistem yang akan Dibangun

Dari hasil perancangan masukan dan keluaran yang telah di analisa, kemudian ditentukan jumlah *hidden*, baik jumlah layernya ataupun jumlah unit pada tiap *layer* yang ada. Jumlah *hidden* ditentukan dengan cara *trial and error*, dalam arti hasil pembelajaran yang tercepat dan terbaik itulah yang akan menentukan jumlah *hidden layer* tersebut. Adapun perancangan jaringan saraf tiruan *backpropagation* adalah seperti Gambar 2.

## 4. HASIL PERCOBAAN

Dari hasil perancangan yang telah dibuat dihasilkan percobaan yang terbaik. Terbaik disini diartikan mempunyai hasil ketepatan atau kebenaran yang cukup baik (lebih dari 82 %) dan pembelajaran (waktu maupun kestabilan) yang cukup baik. Dari percobaan yang dilakukan didapat jumlah iterasi maksimal yang dicapai mencapai 25000 iterasi, dan diperlukan pembelajaran 14 jam terhadap seluruh pola yang diberikan. Selain hal tersebut juga ditetapkan konstanta pembelajaran terbaik adalah 0.02 dengan toleransi kesalahan 0.01 dan lapisan tersembunyi 1 layer dengan jumlah unit 10 (lihat Tabel 1 dan 2)s.

Tabel 2. Hasil pengujian dengan konstanta belajar 0.7

<b>Toleransi kesalahan</b>	<b>Jumlah percobaan</b>	<b>Banyak pola</b>	<b>Prosentase kesuksesan</b>
0.1	100	200	40 %
0.1	100	420	75 %
0.05	100	200	60 %
0.05	100	420	77 %
0.01	100	200	61 %
0.01	100	420	82 %

Tabel 3. Hasil pengujian dengan konstanta belajar 0.02

<b>Toleransi kesalahan</b>	<b>Jumlah percobaan</b>	<b>Banyak pola</b>	<b>Prosentase kesuksesan</b>
0.1	100	200	78 %
0.1	100	420	81 %
0.05	100	200	78 %
0.05	100	420	83 %
0.01	100	200	81 %
0.01	100	420	83 %

## 5. KESIMPULAN DAN SARAN

JST *backpropagation* dapat melakukan pembelajaran dan pengenalan terhadap suatu pola dengan tingkat generalisasi yang cukup tinggi. Sistem identifikasi menghasilkan tingkat generalisasi sebesar 82,330%. Tingkat kegagalan (*failure*) terhadap data percobaan sebesar 17 % dari total 100 percobaan.

Penelitian ini masih dapat dikembangkan lebih jauh lagi dan lebih dalam lagi yang nantinya diharapkan dapat terbentuk suatu sistem yang lebih baik. Saran-saran bagi penelitian ini lebih lanjut antara lain adalah:

1. Penelitian identifikasi wilayah menggunakan objek citra normal. Hal ini dimaksudkan bahwa data hasil foto satelit dapat langsung diaplikasikan dengan aplikasi yang dilatih di dalamnya.
2. Penggunaan model normalisasi wilayah kota yang lebih baik, yaitu melibatkan proses point nomor 4 dibawah.
3. Penggunaan JST yang bersifat *incremental learning* sehingga JST dapat mengenali pola baru dengan lebih cepat.
4. Penggunaan momentum untuk menentukan bobot jaringan saraf tiruan yang akan dilatih akan mempercepat dan memperingkas waktu pembelajaran.
5. Penggunaan teknik *filtering, noise reduction, dan end point detection* sehingga sinyal suara *digital* yang dihasilkan akan lebih baik dari segi kualitas maupun dalam jumlah besarnya data.

#### **PUSTAKA**

- Chester, M. (1991). *Neural Network A Tutorial*. New Jersey: Prentice Hall.
- Haryono, M. E. A. (2005). *Diktat Kuliah Jaringan Saraf Tiruan*. Yogyakarta: Jurusan Teknik Informatika Universitas Islam Indonesia.
- Haryono, M. E. A. (2002). Sistem Cerdas Identifikasi Objek 2-Dimensi pada Pemetaan Wilayah Geografis dengan Metode Jaringan Saraf Tiruan, *Tugas Akhir*; Jurusan Teknik Informatika, Universitas Islam Indonesia.
- Fauset, L. (1994). *Fundamental of Neural Network*. New Jersey: Prentice Hall.
- Haykin, S. (1995). *Neural Network:: A Comprehensive Foundation*. New York: Macmillan College Publishing.
- Low, A. (1991) *Introductory Computer Vision and Image Processing*. London: Mc Graw-Hill.
- Mardi, A., dan Harjoko, A. (2000). Pemahaman Citra Teks Menggunakan Jaringan Saraf Tiruan, *Prosiding Seminar Nasional Aplikasi Sistem Cerdas dalam Rekayasa dan Bisnis*, 189-1999.
- Suparman (1991). *Mengenal Artificial Intelligence*. Yogyakarta: Andi Offset.