

## PERANCANGAN DAN IMPLEMENTASI JARINGAN SYARAF TIRUAN UNTUK MENDIAGNOSA JENIS PENYAKIT KANDUNGAN

**Bambang Yuwono**

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta  
Jl. Babarsari No.2 Yogyakarta 55281  
Email: [bambangy@gmail.com](mailto:bambangy@gmail.com)

### **Abstrak**

*Jaringan syaraf tiruan merupakan salah satu cabang Artificial Intelligence yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Jaringan syaraf tiruan mampu menyelesaikan permasalahan dalam bidang kedokteran, yaitu mendeteksi suatu penyakit. Aplikasi ini digunakan untuk mendiagnosa jenis penyakit kandungan. Pelatihan yang dilakukan menggunakan metode Backpropagation yang merupakan sebuah pembelajaran terawasi. Aplikasi ini dibangun menggunakan pemrograman visual basic 6.0 dan Microsoft Access sebagai databasenya. Gejala-gejala penyakit kandungan yang digunakan sebagai input untuk mendiagnosa penyakit tersebut terdiri dari 16 variabel dengan kode target penyakit 00 untuk penyakit Myoma Uteri, 01 Kanker Serviks, 10 Kanker Ovarium, dan 11 Radang Panggul. Arsitektur jaringan syaraf tiruan dengan 16 variabel menggunakan 1 lapisan tersembunyi dengan 10 buah sel lapisan.*

### **1. PENDAHULUAN**

Seiring dengan perkembangan zaman banyak perubahan teknologi dan informasi yang mengalami kemajuan yang pesat. Peranan komputer sangat penting untuk membantu pekerjaan manusia sehari-hari dalam segala aspek bidang. Pemakai mulanya menggunakan komputer sebagai mesin ketik yang dapat bekerja lebih cepat, tepat maupun otomatis. Sejalan dengan perkembangan saat ini, para ahli mencoba menggantikan komputer menjadi suatu alat bantu yang dapat meniru cara kerja otak manusia, sehingga diharapkan suatu saat akan tercipta komputer yang dapat menimbang dan mengambil keputusan sendiri. Hal inilah yang mendorong lahirnya teknologi AI (*Artificial Intelligence*).

Salah satu teknik komputasi yang dikelompokkan dalam AI adalah jaringan syaraf tiruan (*Artificial Neural Network*). Jaringan syaraf tiruan merupakan salah satu sistem pemrosesan yang dirancang dan dilatih untuk memiliki kemampuan seperti yang dimiliki oleh manusia dalam menyelesaikan persoalan yang rumit dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Jaringan syaraf mensimulasi struktur proses-proses otak (fungsi syaraf

biologis) dan kemudian membawanya kepada perangkat lunak kelas baru yang dapat mengenali pola-pola yang kompleks serta belajar dari pengalaman-pengalaman masa lalu.

Di bidang kedokteran, peranan dokter dalam mendiagnosa penyakit pada pasien membutuhkan ketelitian dari jenis penyakit yang diderita. Untuk itu pada penelitian ini, dipelajari dan dicoba membuat aplikasi jaringan syaraf tiruan untuk mendiagnosa jenis penyakit kandungan.

Jaringan syaraf tiruan dalam mendiagnosa jenis penyakit kandungan ini merupakan sebuah memori yang menyimpan sejumlah data, meliputi informasi pada gejala, diagnosis, dan informasi lain sebagai suatu perawatan untuk hal-hal tertentu yang berhubungan dengan penyakit kandungan. Pelatihan jaringan dapat dipresentasikan dengan input yang terdiri dari serangkaian gejala yang diidap oleh penderita. Setelah itu jaringan syaraf akan melatih input gejala tersebut, sehingga ditemukan suatu akibat dari gejala tersebut yaitu jenis penyakit kandungan.

Sehingga aplikasi jaringan syaraf tiruan ini dapat membantu seorang dokter dalam memberikan keputusan klinis dimana perangkat ini memiliki pengetahuan berbagai data kasus hasil analisis komprehensif dari para dokter ahli yang berpengalaman.

## 2, LANDASAN TEORI

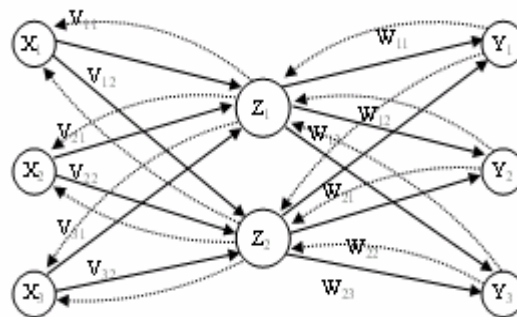
### 2.1. SISTEM JARINGAN SYARAF TIRUAN

Definisi JST menurut Sri Kusumadewi : “merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut”. Istilah buatan disini digunakan karena jaringan syaraf tiruan ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. *Neural Network* ini meniru otak manusia dari sudut : 1) Pengetahuan yang diperoleh oleh *Network* dari lingkungan, melalui suatu proses pembelajaran. 2) Kekuatan koneksi antar unit yang disebut *synaptic weight*, berfungsi untuk menyimpan pengetahuan yang telah diperoleh oleh jaringan tersebut “. Karakteristik lain yang penting dari jaringan syaraf tiruan, yang sama dengan sistem syaraf biologis adalah toleransi kesalahan (*fault tolerance*) (Kusumadewi, 2003). Sistem syaraf biologis memberikan toleransi kesalahan dalam dua hal. Pertama, dapat mengenali banyak sinyal input yang agak berbeda dengan sembarang sinyal input lain yang pernah dilihat sebelumnya. Ciri utama yang dimiliki oleh JST adalah kemampuannya untuk belajar. Belajar (*learning*) pada JST dapat diartikan sebagai proses penyesuaian parameter pembobot, karena

keluaran yang diinginkan tergantung pada harga pembobot interkoneksi yang dimiliki sel. Proses belajar akan dihentikan jika nilai kesalahan atau *error* sudah dianggap cukup kecil untuk semua pasangan data latihan. Jaringan yang sedang melakukan proses belajar disebut berada dalam tahap pelatihan (*training*). Pada tahap awal, pelatihan ini perlu dilakukan terlebih dahulu sebelum melakukan tahap pengujian (*testing*) suatu objek.

## 2.2. ARSITEKTUR BACKPROPAGATION

Arsitektur jaringan *backpropagation* terdiri dari banyak lapisan (*multilayer*) memiliki satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki satu atau lebih lapisan tersembunyi). (Fauset, 1994) Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Contoh arsitektur jaringan syaraf tiruan dengan banyak lapisan.



Gambar 2.1 Arsitektur Backpropagation

## 2.3. ALGORITMA PELATIHAN

Algoritma pelatihan jaringan *Backpropagation* :

1. Langkah 0: Inisialisasi bobot
2. Langkah 1: Selama kondisi berhenti bernilai salah, kerjakan :
  - a. Untuk masing-masing pasangan pelatihan, lakukan :
 

*Feedforward* (perambatan maju):

    1. Masing-masing unit input ( $x_i$ ,  $i = 1, 2, 3, \dots, n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
    2. Masing-masing unit tersembunyi ( $z_j$ ,  $j = 1, 2, 3, \dots, p$ ) menjumlahkan sinyal-sinyal input berbobot:

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad v_0 = \text{bias} ; v = \text{bobot} \quad \dots\dots\dots(2.1)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z\_in_j) \quad \dots\dots\dots(2.2)$$

Kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

3. Masing-masing unit output ( $y_k$ ,  $k = 1,2,3,\dots,m$ ) menjumlahkan sinyal-sinyal input berbobot:

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad w_0 = \text{bias} ; w = \text{bobot} \quad \dots\dots\dots(2.3)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y\_in_k) \quad \dots\dots\dots (2.4)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

- b. Untuk masing-masing pasangan pelatihan, lakukan :

*Backpropagation* (perambatan mundur):

1. Masing-masing unit output ( $y_k$ ,  $k = 1,2,3,\dots,m$ ) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi *errorny*

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad t = \text{target output} \quad \dots\dots\dots(2.5)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperoleh nilai  $w_{jk}$ ):

$$\Delta w_{jk} = \alpha \delta_k z_j \quad \alpha = \text{learning rate} \quad \dots\dots\dots(2.6)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{0k}$ ):

$$\Delta w_{0k} = \alpha \delta_k \quad \dots\dots\dots(2.7)$$

Kirimkan ini ke unit-unit yang ada di lapisan di bawahnya.

2. Masing-masing unit tersembunyi ( $z_j$ ,  $j = 1,2,3,\dots,p$ ) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad \dots\dots\dots(2.8)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_j = \delta\_in_j f'(z\_in_j) \quad \dots\dots\dots(2.9)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ):

$$\Delta v_{ij} = \alpha \delta_j x_i \quad \dots\dots\dots(2.10)$$

Hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{0j}$ ):

$$\Delta v_{0j} = \alpha \delta_j \dots\dots\dots(2.11)$$

c. Perbaharui bobot dan bias nya:

1. Masing-masing unit output ( $y_k, k = 1,2,3,\dots,m$ ) memperbaiki bias dan bobotnya ( $j = 0,1,2,\dots,p$ ):

$$w_{jk}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (\text{bobot}) \quad \dots\dots\dots(2.12)$$

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (\text{bias}) \quad \dots\dots\dots(2.13)$$

2. Masing-masing unit tersembunyi ( $z_j, j = 1,2,3,\dots,p$ ) memperbaiki bias dan bobotnya ( $l = 0,1,2,\dots,n$ ):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (\text{bobot}) \quad \dots\dots\dots(2.14)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (\text{bias}) \quad \dots\dots\dots(2.15)$$

d. Tes kondisi berhenti

**2.4. PENGUJIAN DATA**

Setelah proses pembelajaran dilakukan berdasarkan langkah-langkah algoritma *backpropagation*, maka diperoleh nilai bobot akhir dan nilai bias akhir yaitu dari iterasi terakhir dengan nilai *error* atau galat  $\leq 0.01$ . Nilai bobot dan bias inilah yang akan digunakan dalam melakukan pengujian terhadap data dengan perambatan maju (*forward propagation*). Adapun langkah-langkah pengujian data adalah sebagai berikut:

*Feedforward* (perambatan maju):

1. Masing-masing unit input ( $x_i, i = 1,2,3,\dots,n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
2. Masing-masing unit tersembunyi ( $z_j, j = 1,2,3,\dots,p$ ) menjumlahkan sinyal-sinyal input berbobot:

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad \dots\dots\dots(2.16)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z\_in_j) \quad \dots\dots\dots(2.17)$$

Kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

3. Masing-masing unit output ( $y_k, k = 1,2,3,\dots,m$ ) menjumlahkan sinyal-sinyal input berbobot:

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad \dots\dots\dots(2.18)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k}) \quad \dots\dots\dots(2.19)$$

Kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

### 3. HASIL DAN PEMBAHASAN

Aplikasi ini digunakan oleh seorang dokter untuk mengoptimasikan deteksi jenis penyakit kandungan. Dimana proses perhitungan deteksi penyakit kandungan ini dilakukan setelah perolehan data-data gejala klinis dari hasil pemeriksaan dokter terhadap pasien.

Implementasi masukan/input yang dapat dimengerti oleh program adalah 16 variabel yang mampu menegakkan diagnosa penyakit kandungan *Myoma Uteri*, Kanker *Serviks*, Kanker *Ovarium*, dan Radang Panggul yang harus diinisialisasi terlebih dahulu.

1. Anemia (kurang darah)
2. Nyeri Haid
3. Susah hamil
4. Terdapat benjolan diperut
5. Terjadi pendarahan
6. Nyeri waktu berhubungan seksual
7. Cepat lelah
8. Kehilangan berat badan
9. Nyeri panggul
10. Terjadi gangguan pencernaan
11. Nyeri perut
12. Nyeri punggung
13. Nafsu makan berkurang
14. Demam tinggi
15. Sakit kepala
16. Perut kembung

Setelah variabel-variabel yang dominan dalam penegakan diagnosa *Myoma Uteri*, Kanker *Serviks*, Kanker *Ovarium*, dan Radang Panggul ditentukan, maka harus ditentukan pula

nilai dari variabel-variabel tersebut. Nilai yang diberikan berkisar antara 0 sampai 1, tergantung dari kasusnya per variabel. Hal ini juga memungkinkan bahwa pemberian nilai berdasarkan tingkat parahnya kasus, semakin parah kasusnya maka semakin tinggi pula nilainya.

**Tabel Nilai dari masing-masing variabel**

No.	Variabel-variabel	Nilai dari variabel
1.	Anemia ( $X_1$ )	Tidak Anemia = 0 Ya Anemia = 1
2.	Nyeri Haid ( $X_2$ )	Tidak nyeri haid = 0 Nyeri setelah haid = 0.5 Nyeri waktu haid = 1
3.	Susah Hamil ( $X_3$ )	Tidak susah hamil = 0 Susah hamil = 1
4.	Terdapat benjolan diperut ( $X_4$ )	Tidak terdapat benjolan = 0 Terdapat benjolan diperut = 1
5.	Terjadi pendarahan ( $X_5$ )	Tidak terjadi pendarahan = 0 Terjadi pendarahan menstruasi yang tidak normal = 0.5 Terjadi pendarahan spontan (tiba-tiba) = 1
6.	Cepat lelah ( $X_6$ )	Tidak cepat lelah = 0 Cepat lelah = 1
7.	Berat badan menurun ( $X_7$ )	Berat badan tidak menurun = 0 Berat badan menurun = 1
8.	Nyeri panggul ( $X_8$ )	Tidak nyeri panggul = 0 Ya nyeri panggul = 1
9.	Gangguan pencernaan ( $X_9$ )	Tidak mengalami gangguan pencernaan = 0 Mengalami gangguan pencernaan seperti susah buang air = 0.5 Mengalami gangguan pencernaan seperti diare = 1
10.	Nyeri perut ( $X_{10}$ )	Tidak mengalami nyeri perut = 0 Mengalami nyeri perut pada rongga perut = 0.25 Mengalami nyeri perut pada bagian bawah = 0.5 Mengalami nyeri perut pada bagian bawah dan daerah panggul = 1
11.	Nyeri punggung ( $X_{11}$ )	Tidak mengalami nyeri punggung = 0 Ya mengalami nyeri punggung = 1
12.	Nyeri waktu berhubungan seksual ( $X_{12}$ )	Tidak mengalami nyeri waktu berhubungan seksual = 0 Ya mengalami nyeri waktu berhubungan seksual = 1
13.	Nafsu makan berkurang ( $X_{13}$ )	Nafsu makan tidak berkurang = 0 Nafsu makan berkurang = 1
14.	Demam tinggi ( $X_{14}$ )	Tidak demam = 0 Demam = 1

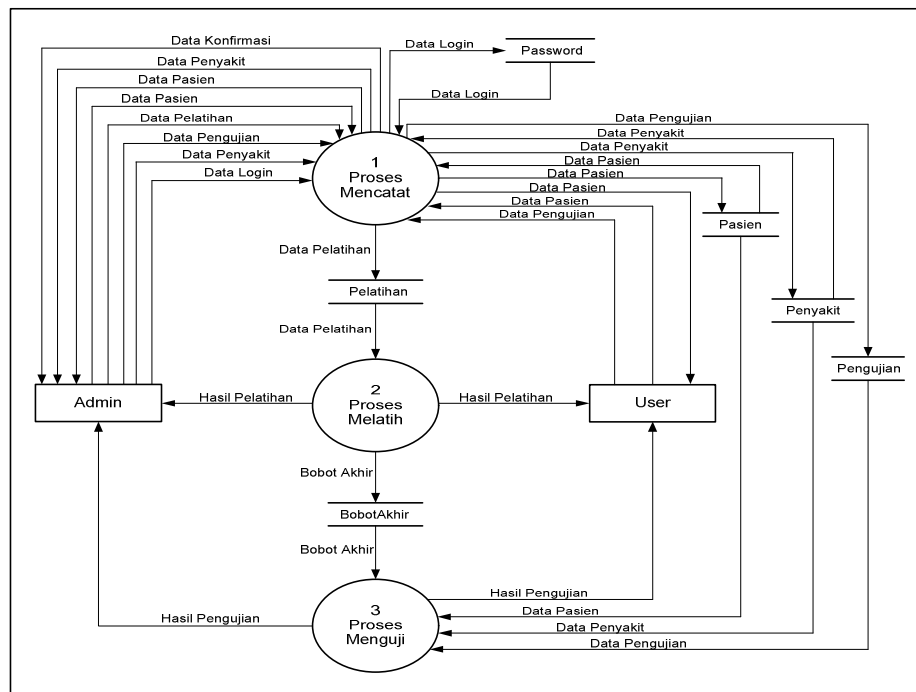
15.	Sakit kepala ( $X_{15}$ )	Tidak Sakit kepala = 0 Sakit kepala = 1
16.	Perut Kembung ( $X_{16}$ )	Tidak perut kembung = 0 Mengalami perut kembung = 1

Output yang dihasilkan adalah hasil diagnosa penyakit, node outputnya adalah 2 node. Hasil output yang diinginkan berupa nilai :

**Tabel Nilai Target Output**

No	Nilai target (Biner)	Nama Penyakit
1.	00	<i>Myoma Uteri</i>
2.	01	<i>Kanker Serviks</i>
3.	10	<i>Kanker Ovarium</i>
4.	11	<i>Radang Panggul</i>

Gambaran secara umum proses aplikasi JST untuk diagnosa penyakit kandungan, dapat dilihat pada gambar 3.1 berikut.



**Gambar 3.1** Gambaran umum proses aplikasi JST diagnosa penyakit kandungan

Menu input ini dapat menyimpan data pasien yang sudah diinput yang selanjutnya akan disimpan dalam bentuk tabel. Seperti terlihat pada gambar 3.2 berikut.

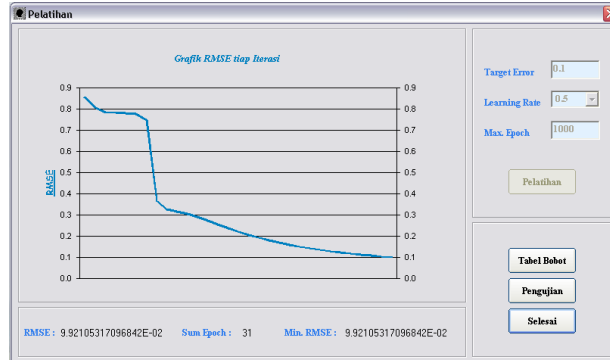




The screenshot shows a window titled "Input Pasien" with a sub-header "IDENTITAS PASIEN". It contains several input fields: "Tgl. Masuk" with value "18-01-07", "No. Pasien" with value "0015", "Nama" with value "nuri", "Umur" with value "22" and a unit dropdown set to "Tahun", "Alamat" with value "Kaliwajahan", and "No. Tlp" with value "684234". At the bottom, there are four buttons: "Tambah", "Tabel", "Batal", and "Diagnosa".

**Gambar 3.2** Menu input data pasien

Pada proses pelatihan menggunakan konstanta belajar yang digunakan yaitu 0.1 sampai dengan 0.9 dengan kenaikan 0.1. Proses pelatihan dilakukan sebelum melakukan pengujian. Nilai target *error* yang diizinkan pada proses pelatihan adalah 0.1 dan 0.01, dengan iterasi maksimum 100.000. Setelah pengisian lengkap maka data akan dihitung dengan menekan tombol *Learning*. Pada proses ini akan terjadi *looping*, dan berhenti jika sudah mencapai kondisi yang diinginkan sesuai dengan masukan pada *target error* dan *learning rate*. Hasil penghitungan akhir menghasilkan bobot akhir yang selanjutnya akan disimpan dalam bentuk tabel bobot. Seperti terlihat pada tampilan gambar 3.2 menu pelatihan:



**Gambar 3.3** Menu pelatihan

Setelah melakukan pelatihan, maka cari data pasien yang akan diperiksa. Kemudian pilih gejala-gejala yang diidap pasien tersebut. Lalu lakukan pengujian dengan menekan tombol *Deteksi*. Hasil pengujian akan ditampilkan pada *frame* pengujian. Pengujian seperti terlihat pada gambar 3.4 berikut.

**Gambar 3.4** Menu Penguujian

Penguujian JST dilakukan terhadap 16 variabel gejala-gejala dari penyakit kandungan yang dominan. Nilai bobot yang akan dipakai dalam penguujian diambil dari nilai bobot hasil pelatihan, kemudian mengubah-ubah nilai dari *learning rate* 0,1 sampai 0,9 dan target *error* pada masing-masing data. Data yang diujikan menggunakan nilai target *error* 0,1 dan 0,01, nilai *learning rate* 0,1 sampai 0,9. Hasil penguujian diagnosa penyakit kandungan untuk data yang dilatihkan dengan hasil optimum menggunakan *learning rate* 0,8, ketepatan diagnosa mencapai lebih dari 85%.

#### 4. KESIMPULAN

Dalam penelitian ini telah berhasil dirancang dan diimplementasikan sistem jaringan syaraf tiruan untuk mendiagnosa jenis penyakit kandungan. Penguujian jaringan syaraf tiruan berhasil dengan baik, dimana ketepatan penguujian lebih dari 85% dengan menggunakan *target error* 0,1 dan 0,01 dengan *learning rate* 0,1 sampai 0,9 dan 10 unit pengolah. Dengan sistem diagnosa jenis penyakit kandungan ini bisa memberikan suatu alternatif baru untuk mendiagnosa jenis penyakit kandungan pada wanita khususnya *Myoma Uteri*, *Kanker Serviks*, *Kanker Ovarium*, dan *Radang Panggul*.

#### DAFTAR PUSTAKA

- Fausett, L., 1994, *Fundamentals Of Neural Networks Architectures, Algorithms, and Applications*, Prentice-Hall New Jersey.
- Haykin, S., 1994, *Neural Networks (A Comprehensive Foundation)*, Macmillan Collage Publishing Company New York.
- Kusumadewi, S., 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu Yogyakarta.
- Mansjoer, A., dkk, 2003, *Kapita Selekt Kedokteran Jilid 1 Edisi Ketiga FKUI*, Media Aesculapius Jakarta.
- Yasim, F., 2005, *Penyakit Kandungan*, Pustaka Populer Obor, Jakarta.