
PENERAPAN METODE JARINGAN SYARAF TIRUAN UNTUK MENGETAHUI KEBENARAN HASIL PERKAWINAN ANTAR GOLONGAN DARAH

Nopriadi

Dosen Program Studi Manajemen Informatika
Akademi Manajemen Informatika dan Komputer (AMIK) Depati Parbo Kerinci
Jl. Pancasila No.29 Sungai Penuh

ABSTRACT

Every human has a blood type. There are four types of human blood groups A, B, AB and O. In medicine, the human blood group classification is done by testing human blood with antigen A and antigen B. blood group could also serve as proof of verification child's of a couple. This is because the child's blood type consisting of a combination of their parents. the systems, a branch of the neural networks that is perceptron can be applied to determine the truth of the marriage between blood type or inheritance from parent to child's. The variables used are four blood groups on the cross-model results, the variables of the four blood groups can be separated by one in each a different output. Whereas, the parameter is the target of the marriage with the output we are looking for. If it matches said proven and if it does not it is said not proven. The Neural Network of the algorithm Perceptron consists of a layer of input units, and layers to Output. For input unit will be included husbands blood type and wife blood type are used as a matrix in each of typeface, then made the determination that the target output of the pattern of blood would have been trained in accordance with set targets indicating that blood group managed to prove so the network could ultimately decide the inheritance of blood groups proved or not proved.

Keywords: Artificial neural network, perceptron, blood type, inheritance.

PENDAHULUAN

Perkembangan teknologi komputer pada saat ini mengalami kemajuan yang sangat pesat, khususnya dalam bidang informasi, baik itu informasi kesehatan, bisnis, pendidikan dan informasi lainnya. Hasil kerja sistem komputer ini diharapkan dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia, itulah yang mendorong lahirnya teknologi AI (*Artificial Intelligence*). Perkembangan selanjutnya para ahli mencoba untuk mengembangkan sistem yang lebih baik, sehingga diharapkan suatu saat nanti akan tercipta suatu komputer yang dapat menimbang dan mengambil suatu keputusan sendiri sebagaimana layaknya meniru cara kerja otak manusia dalam menyelesaikan suatu masalah. Berdasarkan hal tersebut dikembangkan suatu jaringan yang memiliki konsep mirip dengan otak manusia. Jaringan tersebut adalah Jaringan Syaraf Tiruan.

JST atau *Artificial Neural Network (ANN)* adalah bagian dari sistem kecerdasan buatan (*Artificial Intelligence, AI*) yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah tiruan atau buatan dimaksudkan karena jaringan syaraf ini diimplementasikan dengan menggunakan komputer yang mampu menyelesaikan sejumlah proses pembelajaran. JST dimaksudkan untuk membuat model sistem komputasi yang dapat menirukan cara kerja jaringan syaraf biologis (Arief Heru Kuncoro dan Rinaldy Dalimi, 2005).

Jaringan Syaraf Tiruan diperkenalkan melalui beberapa komputasi menggunakan neuron-neuron sederhana yang dapat mengubahnya menjadi sebuah sistem baru yang mempunyai kemampuan komputasi yang lebih baik. Dengan seiring perkembangan Jaringan Syaraf Tiruan ini memiliki beberapa metode perhitungan secara sederhana seperti metode Perceptron, Hebb Rule, Delta Rule, dan metode lainnya.

Dalam bidang kesehatan, baik di negara maju maupun dinegara berkembang tenaga kesehatan dengan keahliannya dituntut untuk mampu dan dapat memberikan kontribusi dalam hal membedakan golongan darah manusia antara golongan darah A, B, AB dan O melalui perhitungan ABO dan perhitungan rhesus. Dengan pendekatan kecerdasann buatan, manusia diharapkan dapat menirukan bagaimana pola – pola dibentuk. Jaringan Syaraf Tiruan telah dikembangkan sebagai generalisasi model matematik dari pembelajaran manusia.

Jaringan syaraf tiruan (*Artificial Neural Network*) bisa diterapkan untuk memeriksa kebenaran hasil perkawinan antar golongan darah, yang selama ini masih dikerjakan dengan menggunakan sistem manual.

Dengan menggunakan Jaringan Syaraf Tiruan diharapkan pemeriksaan kebenaran hasil perkawinan antar golongan darah ini dapat dilakukan dengan cepat sehingga hasilnya pun dapat diketahui dalam waktu yang singkat dan mempunyai bobot yang tinggi.

Konsep Teoritis

Sri Kusumadewi (2003) mendefinisikan Jaringan Syaraf Tiruan (JST) 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 ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.

Menurut Eli Yani (2005), Jaringan Syaraf Tiruan adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi.

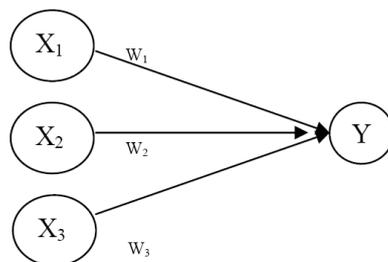
Menurut Elly Warni (2009), Jaringan Syaraf Tiruan merupakan model komputasi terdistribusi yang meniru cara kerja dan sistem syaraf biologis.

Dari definisi diatas dapat disimpulkan bahwa Jaringan Syaraf Tiruan adalah suatu program yang diciptakan sebagai informasi yang menyamakan dengan otak manusia.

JST ditentukan oleh 3 hal :

- a. Pola hubungan antar *neuron*(disebut arsitektur jaringan)
- b. Metode untu menentukan bobot penghubung(disebut metode *training/learning/algortma*)
- c. Fungsi aktivasi

Sebagai contoh, perhatikan *neuron* Y pada gambar 2.1



Gambar 2.1 Sebuah Sel Syaraf Tiruan

Y menerima input dari *neuron* x_1 , x_2 dan x_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 dan w_3 . Argumen fungsi aktivasi adalah net masukan (kombinasi linear masukan dan bobotnya). Ketiga sinyal simpul yang ada dijumlahkan $net = x_1w_1 + x_2w_2 + x_3w_3$. Besarnya sinyal yang diterima oleh Y mengikuti fungsi aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

Komponen Jaringan Syaraf

Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa *neuron*, dan ada hubungan antara *neuron-neuron* tersebut. *Neuron-neuron* tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarannya menuju ke *neuron-neuron* yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar 2.2 menunjukkan *struktur neuron* pada jaringan syaraf.



Gambar 2.2 Struktur *Neuron* Jaringan Syaraf

Arsitektur Jaringan Syaraf

Ada beberapa arsitektur jaringan syaraf, antara lain:

a. Jaringan Dengan Lapisan Tunggal (*Single Layer Net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengelolanya menjadi *output* tanpa harus melalui lapisan tersembunyi. Pada gambar 2.2 menjelaskan lapisan input memiliki 3 *neuron*, yaitu X1, X2 dan X3. Sedangkan pada lapisan *output* memiliki 2 *neuron* yaitu Y1 dan Y2. *Neuron-neuron* pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 *neuron* ditentukan oleh bobot yang bersesuaian. Semua unit *input* akan dihubungkan dengan setiap unit *output*.

b. Jaringan Dengan Banyak Lapisan (*Multilayer Net*)

Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki satu atau lebih lapisan tersembunyi). Umumnya, ada lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan

tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, banyak kasus pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.

c. Jaringan Dengan Lapisan Kompetitif (*Competitive Layer Net*)

Umumnya, hubungan antar *neuron* pada lapisan *competitive* ini tidak diperlihatkan pada diagram arsitektur.

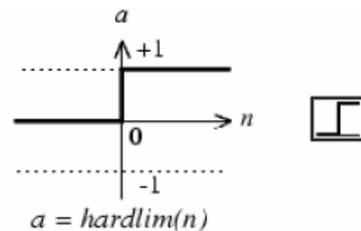
Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang mentransformasikan nilai penjumlahan menjadi sebuah nilai yang dapat diproses lebih lanjut. Terdapat beberapa fungsi aktivasi, yaitu:

1. Fungsi Undak Biner (*hard limit*).

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak biner (*step function*) untuk mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1). Gambar 2.7 menunjukkan fungsi undak biner (*hard limit*) yang dirumuskan sebagai:

$$y = \begin{cases} 0, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases}$$

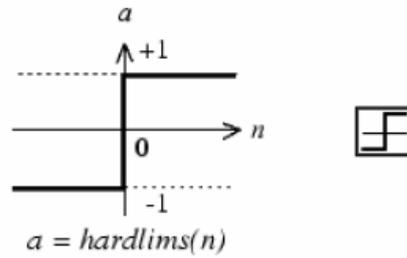


Gambar 2.3 Fungsi Aktivasi Undak Biner (*Hard Limit*)

2. Fungsi Bipolar (*symetric hard limit*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja *output* yang dihasilkan berupa 1 atau -1. Gambar 2.8 fungsi *Symetric Hard Limit* yang dirumuskan sebagai :

$$y = \begin{cases} 1, & \text{jika } x \geq 0 \\ -1, & \text{jika } x < 0 \end{cases}$$

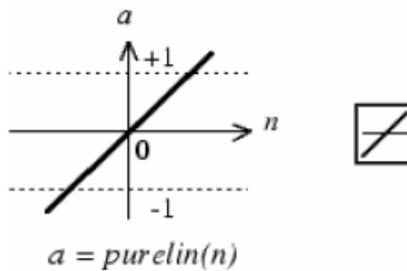
Gambar 2.4 Fungsi Aktivasi Bipolar (*Symmetric Hard Limit*)

3. Fungsi Linear (*purelin*)

Fungsi linear memiliki nilai *output* yang sama dengan nilai *inputnya*.

Gambar 2.9 fungsi linear yang dirumuskan sebagai:

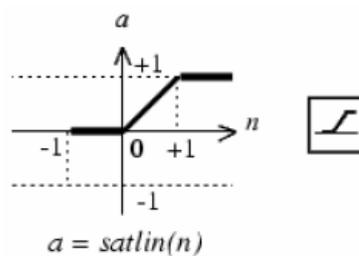
$$y = x$$

Gambar 2.5 Fungsi Aktivasi Linear (*Purelin*)

4. Fungsi Saturating Linear (*satlin*)

Fungsi ini akan bernilai 0 jika *inputnya* kurang dari -1/2 dan akan bernilai 1 jika *inputnya* lebih dari 1/2. Sedangkan jika nilai *input* terletak antara -1/2 dan 1/2 maka *outputnya* akan bernilai sama dengan nilai *input* ditambah 1/2. fungsi *saturating linear* yang dirumuskan sebagai :

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x, +0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0, & \text{jika } x \leq -0,5 \end{cases}$$

Gambar 2.6 Fungsi Aktivasi Saturating Linear (*Satlin*)

5. Fungsi Sigmoid Biner (*logsig*)

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai *outputnya* 0 atau 1. Fungsi sigmoid biner dirumuskan sebagai :

$$y = f(x) = \frac{1}{1 + e^{-\alpha}}$$

dengan $f'(x) = \sigma f(x)[1 - f(x)]$

6. Fungsi Sigmoid Bipolar (*tansig*)

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1. Fungsi sigmoid bipolar dirumuskan sebagai :

$$x_2 = \frac{1 - e^{-x}}{1 + e^{-x}}$$

dengan $f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$

Perceptron

Perceptron juga termasuk salah satu bentuk jaringan syaraf yang sederhana. *Perceptron* biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan pemisahan secara *linear*. Pada dasarnya, *perceptron* pada jaringan syaraf dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang (*thereshold*). *Algorima* yang digunakan oleh aturan *perceptron* ini mengatur parameter-parameter bebasnya melalui proses pembelajaran. Nilai *thereshold* (θ) pada fungsi aktivasi adalah *non negative*. Fungsi aktivasi ini dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif.

Garis pemisah antara daerah positif dengan daerah nol memiliki pertidaksamaan sebagai berikut :

$$w_1x_1 + w_2x_2 + b > \theta$$

dengan garis pembatasnya :

$$x_2 = \frac{-x_1 * w_1 - b + \theta}{w_2}$$

atau

$$x_2 = \frac{-w_1}{w_2} x_1 - \frac{b - \theta}{w_2}$$

Garis pemisah antara daerah negatif dengan daerah nol memiliki pertidaksamaan sebagai berikut :

$$w_1 x_1 + w_2 x_2 + b < -\theta$$

Dengan garis pembatasnya :

$$x_2 = \frac{-x_1 * w_1 - b + \theta}{w_2}$$

atau

$$x_2 = \frac{-w_1}{w_2} x_1 - \frac{b - \theta}{w_2}$$

Daerah *undecided bound* adalah yang memisahkan antara daerah respon positif dengan daerah respon negatif yaitu dengan pertidaksamaan :
 $-\theta \leq x_1 w_1 + w_2 x_2 + b \leq \theta$.

Jika kita menggunakan pasangan vektor *input* dan vektor *output* sebagai pasangan vektor yang akan dilatih maka kita akan memiliki algoritma seperti :

Step 0 : Inisialisasi bobot(w_i) dan bias(b)

Set learning rate α ($0 < \alpha \leq 1$)

(untuk sederhananya set $\alpha = 1$)

Step 1 : Jika kondisi *false* tidak terpenuhi, kerjakan step 2 - 6

Step 2 : Untuk setiap pasangn *training* ($s : t$) kerjakan step 3 – 5

Step 3 : set aktivasi dari unit-unit *input* : $x_i = s_i$;

Step 4 : Hitung respon dari unit *output* :

$$y\text{-in} = b + \sum_{i=1}^n x_i w_i$$

$$y = \begin{cases} -1 & net < \theta \\ 0, & -\theta \leq net \leq \theta \\ 1, & net > \theta \end{cases}$$

Step 5 : Perbaiki bobot dan bias jika *error* terjadi

Jika $y \neq t$ maka :

$$w_i (\text{baru}) = w_i (\text{lama}) + \alpha * t * x_i$$

$$b (\text{baru}) = b (\text{lama}) + \alpha * t$$

Jika $y = t$ maka :

$$b (\text{baru}) = b (\text{lama})$$

$$w_i (\text{baru}) = w_i (\text{lama})$$

Step 6 : Tes untuk kondisi stop

Jika tidak ada bobot yang berubah maka stop, alternatif lain lanjut kan

Keterangan :

S : Sensor

T : Target

x_i : Unit input ke-i

s_i : Unit sensor ke-i

w_i : Bobot ke-i

B : Bias

Y : Unit respons (output)

α : Nilai pembelajaran

θ : Nili ambang

I : Banyaknya unit input (1...,n)

Algoritma di atas dapat digunakan untuk *input* biner maupun bipolar, dengan θ tertentu, dan bisa yang dapat diatur. Pada algoritma tersebut bobot-bobot yang berhubungan dengan *input* yang aktif ($x_i \neq 0$) dan bobot-bobot yang tidak menghasilkan nilai y yang benar.

Golongan Darah

Wiko Putrawan (2010) mengungkapkan Golongan darah adalah ciri khusus darah dari suatu individu karena adanya perbedaan jenis karbohidrat dan protein

pada permukaan membran sel darah merah. Sedangkan menurut Abdul Salam M. Sofro (2012), golongan darah adalah salah satu dari dua cairan tubuh luar sel selain cairan interstisial yang berada diantara sel – sel tubuh.

1. Golongan Darah A

Individu dengan golongan darah A memiliki sel darah merah dengan antigen A di permukaan membran selnya dan menghasilkan antibodi terhadap antigen B dalam serum darahnya. Sehingga, orang dengan golongan darah A-negatif hanya dapat menerima darah dari orang dengan golongan darah A-negatif atau O-negatif.

2. Golongan Darah B

Individu dengan golongan darah B memiliki antigen B pada permukaan sel darah merahnya dan menghasilkan antibodi terhadap antigen A dalam serum darahnya. Sehingga, orang dengan golongan darah B-negatif hanya dapat menerima darah dari orang dengan golongan darah B-negatif atau O-negatif

3. Golongan Darah AB

Individu dengan golongan darah AB memiliki sel darah merah dengan antigen A dan B serta tidak menghasilkan antibodi terhadap antigen A maupun B. Sehingga, orang dengan golongan darah AB-positif dapat menerima darah dari orang dengan golongan darah ABO apapun dan disebut *resipien universal*. Namun, orang dengan golongan darah AB-positif tidak dapat mendonorkan darah kecuali pada sesama AB-positif.

4. Golongan Darah O

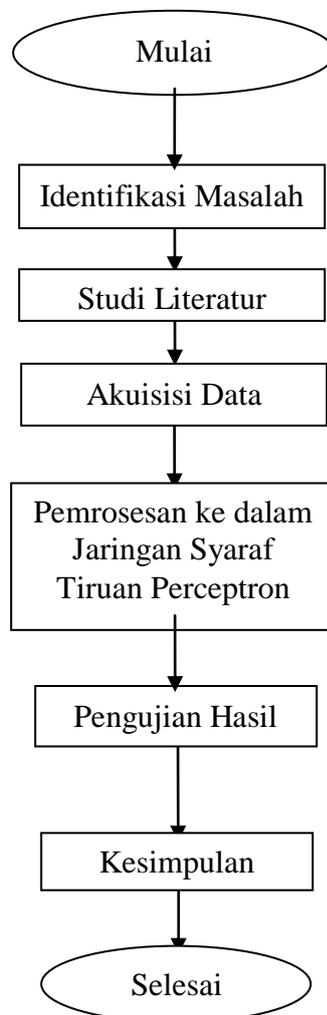
Individu dengan golongan darah O memiliki sel darah tanpa antigen, tapi memproduksi antibodi terhadap antigen A dan B. Sehingga, orang dengan golongan darah O-negatif dapat mendonorkan darahnya kepada orang dengan golongan darah ABO apapun dan disebut *donor universal*. Namun, orang dengan golongan darah O-negatif hanya dapat menerima darah dari sesama O-negatif.

Secara umum, golongan darah O adalah yang paling umum dijumpai di dunia, meskipun di beberapa negara seperti Swedia dan Norwegia, golongan darah A lebih dominan. Antigen A lebih umum dijumpai dibanding antigen B.

Karena golongan darah AB memerlukan keberadaan dua antigen, A dan B, golongan darah ini adalah jenis yang paling jarang dijumpai di dunia.

kerangka Kerja

Kerangka kerja berguna untuk membuat tahapan–tahapan yang akan dilakukan dalam melakukan penelitian. Setiap tahapan dilakukan sesuai dengan yang telah direncanakan. Oleh karena itu semua tahapan pada kerangka kerja penelitian ini berpengaruh pada tahapan selanjutnya. Model kerangka kerja untuk pemecahan masalah yang akan digunakan dalam penelitian ini dapat dilihat dalam diagram kerja dibawah ini.



Gambar 3.1 Kerangka Kerja

Agar langkah-langkah yang diambil penulis dalam penelitian dapat lebih mudah dipahami, berikut ini disertakan penjelasan terperinci mengenai sistematika model metodologi pemecahan masalah.

Identifikasi Masalah

Tahap identifikasi masalah adalah tahap menemukan permasalahan sebelum dilakukannya penelitian. Dengan menggali permasalahan yang ditemukan pada obyek yang diteliti guna mencari alternatif solusi yang terkait dengan permasalahan. Identifikasi masalah dilakukan untuk memperoleh gambaran yang lengkap tentang lingkup masalah dan langkah yang tepat dalam pemecahannya.

Studi Literatur

Studi literatur adalah tahap melakukan pembelajaran mengenai teknik klasifikasi jaringan syaraf tiruan menggunakan algoritma perceptron. Teknik-teknik dan algoritma yang dilakukan dalam proses klasifikasi berbasis jaringan syaraf tiruan berasal dari buku-buku dan jurnal dari *website (internet)* yang berkaitan dengan konsep Jaringan Syaraf Tiruan dan algoritma *perceptron*.

Untuk mencapai tujuan, maka dipelajari beberapa literatur-literatur yang diperkirakan dapat digunakan. Kemudian literatur – literatur yang dipelajari tersebut diseleksi untuk dapat ditentukan literatur – literatur mana yang akan digunakan dalam penelitian ini.

Akuisisi Data

Akuisisi Data berfungsi untuk pengambilan data sebagai bahan baku penelitian selanjutnya akan diimplementasikan untuk mengetahui kebenaran hasil perkawinan antar golongan darah berbasis Jaringan Syaraf Tiruan menggunakan algoritma *Perceptron*. Pengambilan data dilakukan dengan cara menyebarkan kuesioner. Dari hasil kuesioner yang telah didapatkan maka penulis mendapatkan informasi mengenai perkawinan antar golongan darah yang terdiri dari golongan darah ayah, golongan darah ibu yang kemudian akan menghasilkan golongan darah anak, sebagai sumber data yang penulis butuhkan.

Pemrosesan Ke Dalam JST Perceptron

Pemrosesan ke dalam JST *Perceptron* merupakan tindak lanjut dari akuisisi data golongan darah yang telah didapat dari kuesioner. Kemudian

diimplementasikan ke dalam algoritma *perceptron* agar nantinya kebenaran hasil perkawinan antar golongan darah bisa diklasifikasikan sesuai dengan percobaan penelitian menggunakan algoritma *perceptron*.

Dalam pemrosesan ke dalam JST *perceptron*, data yang sudah terkumpul akan di analisa terlebih dahulu. Proses ini terdiri dari :

1. Tahap persiapan

Pada tahap ini penulis melakukan pengecekan data ulang hasil dari kuesioner perkawinan antar golongan darah yang sudah didapatkan sebelumnya.

2. Tahap pemisahan data

Tahap pemisahan data ini dilakukan agar mempermudah penulis dalam melakukan analisis data nantinya.

3. Tahap penentuan metode yang akan digunakan

Tahap penentuan metode ini dilakukan agar mempermudah penulis dalam melakukan analisis nantinya. Adapun metode yang akan digunakan adalah metode jaringan syaraf tiruan *perceptron*.

4. Perancangan data

Setelah data dipersiapkan dan disusun, tahap selanjutnya adalah menentukan *input* dan *output*. Penulis menjadikan golongan darah sebagai *input* dan *output*.

Golongan darah yang akan digunakan adalah golongan darah A, B, AB dan O.

Penulis juga menentukan input dan output bilangan yaitu golongan darah A = 0; B = 0.25; AB = 0.5; dan O = 0.75.

Setelah ditentukan *input* dan *output* yang digunakan, penulis juga memasukkan angka 1 sebagai bias, 0.8 sebagai alfa(α), dan 0.5 sebagai teta(θ).

Pengujian Hasil

Pada tahapan pengujian hasil dilakukan untuk mengetahui cara sistem bekerja dalam mengolah data. Serta bisa mendapatkan hasil – hasil yang diinginkan. Tahapan pengujian dilakukan menggunakan *software*. Dengan cara data yang telah melalui pelatihan kemudian akan dilakukan pengujian secara *software*. Pengujian secara *software* dilakukan dengan menggunakan *software Matlab*.

ANALISA DATA

Faktor-faktor yang Mempengaruhi Pewarisan Golongan Darah

Dalam banyak penelitian sistem penggolongan darah khususnya sistem ABO menyatakan ada beberapa jumlah variabel yang mempengaruhi, dan pada penelitian ini faktor yang mempengaruhi pewarisan golongan darah sistem ABO yaitu :

1. Suami
 - a. Golongan darah A
 - b. Golongan darah B
 - c. Golongan darah AB
 - d. Golongan darah O
2. Istri
 - a. Golongan darah A
 - b. Golongan darah B
 - c. Golongan darah AB
 - d. Golongan darah O

Pemrosesan Data

Berdasarkan data-data di atas, pembuktian menghasilkan 4 (empat) buah prediktor kebenaran hasil perkawinan antar golongan darah.

1. Golongan Darah A
2. Golongan darah B
3. Golongan darah AB
4. Golongan darah O

Fungsi aktivasi yang digunakan untuk mengolah data di atas adalah fungsi *step* (biner), maka data harus ditransformasikan dulu karena *range* keluaran fungsi aktivasi *step* adalah $[0,1]$.

Data Input

Input data untuk melakukan pembuktian kebenaran hasil perawinan antar golongan darah berdasarkan pada data kuesioner yang telah disebar sebelumnya. Format dan contoh data input disajikan pada tabel dibawah ini :

Tabel 4.1 Format dan Contoh Data

No	Golongan Darah Suami	Golongan Darah Istri	Golongan Darah Anak
1	AB	B	AB
2	A	B	A
3	A	B	A
4	A	B	B
5	B	O	O
6	B	A	B
7	B	O	O
8	A	AB	A
9	AB	AB	AB
10	O	A	A
11	A	O	O
12	A	O	O
13	AB	A	A
14	AB	A	AB
15	AB	O	A
16	AB	O	B
17	AB	O	B
18	O	O	O
19	O	O	O
20	A	A	A
21	AB	B	AB
22	AB	AB	AB
23	AB	AB	B
24	A	A	A
25	AB	AB	A
26	AB	AB	AB
27	O	A	A
28	B	O	O
29	O	A	A
30	O	A	O
31	AB	B	A
32	O	B	B

Dari data di atas, kita dapat melakukan konversi menjadi berikut ini. Dalam perhitungan nanti, *input* yang digunakan adalah golongan darah dari kedua orang tua yang akan diubah menjadi bilangan angka yaitu :

Tabel 4.2 Golongan Darah Sebagai Input

Golongan Darah Orang Tua	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈
A	0	0	0	0	0	0	0	0
B	0	1	0	0	0	1	0	0
AB	0	0	1	0	0	0	1	0
O	0	0	0	1	0	0	0	1

Sedangan untuk *output* yang digunakan adalah golongan darah anak.

Tabel 4.3 Golongan Darah Sebagai Output

Golongan Darah Anak	Output			
	Y ₁	Y ₂	Y ₃	Y ₄
A	0	0	0	0
B	0	1	0	0
AB	0	0	1	0
O	0	0	0	1

Setelah *input* dan *output* ditentukan maka penulis melakukan perhitungan dengan menggunakan metode *perceptron*.

Aturan/Rule

Dalam analisis ini penulis juga menggunakan aturan/*rule* sebagai gambaran yang menjadikan perhitungan lebih mudah untuk memahaminya, dengan keterangan yang penulis berikan yaitu :

X₁ = Suami A

X₂ = Suami B

X₃ = Suami AB

X₄ = Suami O

X₅ = Istri A

X₆ = Istri B

X₇ = Istri AB

X₈ = Istri O

Y₁ = Anak A

Y₂ = Anak B

Y3 = Anak AB

Y4 = Anak O

Setelah mencapai *rule* yang di inginkan maka dapat kita lakukan perhitungan pada tiap – tiap tahap metode yang telah ditetapkan.

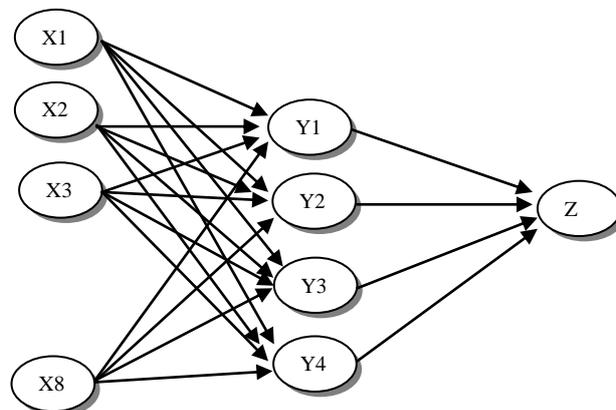
Analisa Perancangan Sistim Jaringan Optimum

Proses selanjutnya adalah proses untuk menentukan jumlah dari lapisan masukan (*input*), lapisan tersembunyi (*hidden layer*) serta jumlah lapisan keluaran (*output layer*). Penggunaan jaringan dengan dua atau lebih lapisan tersembunyi kebanyakan tidak akan memberikan pengaruh yang sangat besar terhadap prestasi jaringan namun akan melambatkan proses pelatihan yang disebabkan bertambahnya unit.

Dalam hal ini penulis menggunakan model jaringan 8-4-1 dengan menggunakan metode *perceptron* terdiri dari;

- Lapisan masukan (*input layer*) dengan 8 simpul ($x_1, x_2, x_3, \dots, x_8$)
- Lapisan tersembunyi (*hidden layer*) dengan jumlah simpul ditentukan oleh pengguna ($y_1, y_2 \dots y_n$).

Lapisan keluaran (*Output layer*) dengan 1 simpul (z).



Gambar 4.1 Arsitektur Jaringan

Menentukan *Learning Rate* dan *Momentum*

Tahapan selanjutnya adalah tahapan penentuan tingkat pembelajaran atau disebut juga *learning rate* dan penentuan momentum. Kedua tahapan ini digunakan untuk menginisialisasikan parameter Jaringan Syaraf Tiruan. *Learning Rate* dan *momentum* sangat berpengaruh terhadap hasil keputusan dari jaringan

yang akan dibuat. Untuk parameter pembelajaran, penulis menggunakan parameter pembelajaran *traingdm*. Dimana pada penggunaan parameter pembelajaran *traingdm* akan mempercepat pelatihan pada jaringan. Hal ini dikarenakan parameter pembelajaran *traingdm* merupakan metode *perceptron* yang dipercepat dengan menggunakan *momentum*. Pada fungsi *traingdm*, nilai dari *learning rate* dan *momentum* dapat dirubah selama proses pelatihan. Terdapat beberapa parameter yang harus diset terlebih dahulu sebelum dilakukan proses pelatihan. Dengan memberikan nilai pada parameter maka akan bisa menghasilkan hasil yang optimal. Parameter yang akan diset adalah :

- *Epoch* maksimum

Epoch maksimum atau disebut juga dengan perulangan maksimum adalah suatu proses untuk mencapai target yang telah ditentukan dimana proses perulangan akan dihentikan ketika nilai dari *epoch* melebihi nilai *epoch* maksimum. Pada matlab digunakan instruksi nilai *default* untuk maksimum *epoch* adalah 10.000, namun penulis akan menyesuaikan dengan kebutuhan jaringan. Instruksi pada matlab yaitu dengan menggunakan sintaks :

```
net.trainParam.epochs = nilai_max_epoch
```

- Kinerja tujuan

Kinerja tujuan atau disebut juga dengan toleransi *error* yang merupakan nilai toleransi dari *error* yang terjadi pada jaringan. Apabila nilai dari toleransi *error* lebih kecil dari nilai maksimum *epoch* maka proses iterasi akan dihentikan. Nilai *default* dari toleransi *error* adalah 0. Namun untuk jaringan ini nilai toleransi (tingkat *error*) yang akan dipakai penulis dalam rentang 0.01 (1%) dan 0.1 (0.1%). Instruksi pada matlab untuk kinerja tujuan adalah

```
net.trainParam.goal = nilai_target_error
```

- *Learning rate*

Learningrate atau disebut juga dengan laju pembelajaran yang mana memiliki fungsi untuk menghitung nilai nilai koreksi bobot pada waktu proses pelatihan. *Range* dari *learning rate* berada dikisaran 0 dan 1. Semakin besar *learning rate* maka proses pelatihan akan berlangsung cepat namun ketelitian jaringan terhadap *error* akan semakin besar sehingga akan memakan waktu dalam

proses pelatihan. Pada jaringan ini *learning rate* yang akan dicobakan yaitu : 0,01 . Instruksi pada matlab yaitu : *net.trainParam.lr = nilai_learning_rate*.

- Momentum

Menurut Siang (2005 : 267), modifikasi metode penurunan tercepat dilakukan dengan menambahkan momentum. Besarnya perubahan bobot ditentukan atau dipengaruhi oleh momentum. Nilai *default* dari momentum adalah 0 sampai dengan 1. Pada jaringan ini momentum yang akan dicobakan yaitu 0,01 dan 0,1. Instruksi pada matlab untuk *set n*.

Proses Pelatihan Jaringan Syaraf Tiruan

Pada metodologi penelitian, proses pelatihan akan dilakukan dengan 2 cara, yaitu pelatihan secara *software* dan secara manual. Pada proses pelatihan ini akan dilakukan pelatihan jaringan terhadap beberapa pola dan parameter yang telah dirancang sebelumnya. Oleh karena itu proses pelatihan Jaringan Syaraf Tiruan adalah sebagai berikut :

Inisialisasi Data

Proses pertama yang akan dilakukan adalah dengan menentukan nilai dari *input* serta pola yang akan digunakan untuk pelatihan Jaringan Syaraf Tiruan. Proses inisialisasi data untuk dijadikan sebagai variabel *input* adalah dengan menjumlahkan data dari kuesioner tersebut berdasarkan dimensi masing – masing.

Setelah dilakukan penentuan nilai *input*, maka selanjutnya akan dilakukan penentuan parameter untuk pelatihan Jaringan Syaraf Tiruan. Berikut adalah parameter *input* berdasarkan pola dan parameter yang telah dirancang sebelumnya dengan menggunakan beberapa pola :

Pelatihan Menggunakan Pola 1

- Jumlah *neuron* pada *hidden layer* sebanyak 3 *neuron*.
- Frekuensi 8000 Hz
- Toleransi *error* sebesar 0.01
- *Learning rate* sebesar 0.01
- Momentum sebesar 0.1
- *Max epoch* sebanyak 10.000 *epoch*

Pelatihan Menggunakan Pola 2

- Jumlah *neuron* pada *hidden layer* sebanyak 3 *neuron*.
- Frekuensi 8000 Hz
- Toleransi *error* sebesar 0.01
- *Learning rate* sebesar 0.01
- Momentum sebesar 0.5
- *Max epoch* sebanyak 10.000 *epoch*

Pelatihan Menggunakan Pola 3

- Jumlah *neuron* pada *hidden layer* sebanyak 3 *neuron*.
- Frekuensi 8000 Hz
- Toleransi *error* sebesar 0.01
- *Learning rate* sebesar 0.1
- Momentum sebesar 0.1
- *Max epoch* sebanyak 10.000 *epoch*

Pelatihan Menggunakan Pola 4

- Jumlah *neuron* pada *hidden layer* sebanyak 3 *neuron*.
- Frekuensi 8000 Hz
- Toleransi *error* sebesar 0.01
- *Learning rate* sebesar 0.1
- Momentum sebesar 0.8
- *Max epoch* sebanyak 10.000 *epoch*

Aktivasi Data

Perhitungan Aktivasi menggunakan metode *perceptron*

$$\text{Fungsi } Y(p) = \sum_{i=1}^n X_i(p) * W_i(p) - \theta$$

$$P_1=1, \quad W = 1,1$$

$$X_1=0, X_2=0, X_3=1, X_4=0, X_5=0, X_6=1, X_7=0, X_8=0$$

$$\theta=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(1).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(1).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+(1.1,1)+ 0.1,1)+(0.1,1) -1] \\ &= \text{Step} [1,2] \end{aligned}$$

P₂=1, W= 1,1

X₁=0, X₂=0, X₃=0, X₄=0, X₅=0, X₆=1, X₇=0, X₈=0

Ø=1,

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(1).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(1.1,1)+ 0.1,1)+(0.1,1) -1] \\ &= \text{Step} [\mathbf{0,1}] \end{aligned}$$

P₃=1, W= 1,1

X₁=0, X₂=0, X₃=0, X₄=0, X₅=0, X₆=1, X₇=0, X₈=0

Ø=1,

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(1).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(1.1,1)+ 0.1,1)+(0.1,1) -1] \\ &= \text{Step} [\mathbf{0,1}] \end{aligned}$$

P₄=1, W= 1,1

X₁=0, X₂=0, X₃=0, X₄=0, X₅=0, X₆=1, X₇=0, X₈=0

Ø=1,

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(1).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(1.1,1)+ (0.1,1)+(0.1,1) -1] \\ &= \text{Step} [\mathbf{0,1}] \end{aligned}$$

P₅=1, W= 1,1

X₁=0, X₂=1, X₃=0, X₄=0, X₅=0, X₆=0, X₇=0, X₈=1

Ø=1,

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(1).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(1.1,1)+ -1] \\ &= \text{Step} [\mathbf{1,2}] \end{aligned}$$

P₆=1, W= 1,1

X₁=0, X₂=1, X₃=0, X₄=0, X₅=0, X₆=0, X₇=0, X₈=0

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(1).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -2] \\ &= \text{Step} [(0.1,1)+(1.1,1)+(0.1,1) + (0.1,1)+(0.1,1)+(0.1,1)+ (0.1,1)+(0.1,1) -1] \\ &= \text{Step} [\mathbf{0,1}] \end{aligned}$$

$$\mathbf{P}_7=1, \quad W=1,1$$

$$X_1=0, X_2=1, X_3=0, X_4=0, X_5=0, X_6=0, X_7=0, X_8=1$$

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(1).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(1.1,1) + -1] \\ &= \text{Step} [\mathbf{1,2}] \end{aligned}$$

$$\mathbf{P}_8=1, \quad W=1,1$$

$$X_1=0, X_2=0, X_3=0, X_4=0, X_5=0, X_6=0, X_7=0, X_8=1$$

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1) + (0.1,1)+(0.1,1)+(0.1,1)+ (0.1,1)+(1.1,1) -1] \\ &= \text{Step} [\mathbf{0,1}] \end{aligned}$$

$$\mathbf{P}_9=1, \quad W=1,1$$

$$X_1=0, X_2=0, X_3=1, X_4=0, X_5=0, X_6=0, X_7=1, X_8=0$$

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(1).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(1).W_7.(1)+X_8.(0).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+ (1.1,1)+(0.1,1) -1] \\ &= \text{Step} [\mathbf{1,2}] \end{aligned}$$

$$\mathbf{P}_{10}=1, \quad W=1,1$$

$$X_1=0, X_2=0, X_3=0, X_4=1, X_5=0, X_6=0, X_7=0, X_8=0$$

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(1).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1] \end{aligned}$$

$$= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+ (0.1,1)+(0.1,1) -1]$$

$$= \text{Step} [0,1]$$

$$P_{11}=1, W= 1,1$$

$$X_1=0, X_2=0, X_3=0, X_4=0, X_5=0, X_6=0, X_7=0, X_8=1$$

$$\dot{O}=1,$$

$$Y(1)=\text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+$$

$$X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1]$$

$$= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+ (1.1,1) -1]$$

$$= \text{Step} [0,1]$$

$$P_{12}=1, W= 1,1$$

$$X_1=0, X_2=0, X_3=0, X_4=0, X_5=0, X_6=0, X_7=0, X_8=1$$

$$\dot{O}=1,$$

$$Y(1)=\text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(0).W_3.(1)+X_4.(0).W_4.(1)+$$

$$X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1]$$

$$= \text{Step} [(0.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+ (0.1,1)+(0.1,1)+(0.1,1)+(1.1,1) -1]$$

$$= \text{Step} [0,1]$$

$$P_{13}=1, W= 1,1$$

$$X_1=0, X_2=0, X_3=1, X_4=0, X_5=0, X_6=0, X_7=0, X_8=0$$

$$\dot{O}=1,$$

$$Y(1)=\text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(1).W_3.(1)+X_4.(0).W_4.(1)+$$

$$X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1]$$

$$= \text{Step} [(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+ (0.1,1)+(0.1,1)+(0.1,1)+(0.1,1) -1]$$

$$= \text{Step} [0,1]$$

$$P_{14}=1, W= 1,1$$

$$X_1=0, X_2=0, X_3=1, X_4=0, X_5=0, X_6=0, X_7=0, X_8=0$$

$$\dot{O}=1,$$

$$Y(1)=\text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(1).W_3.(1)+X_4.(0).W_4.(1)+$$

$$X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(0).W_8.(1) -1]$$

$$= \text{Step} [(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+ (0.1,1)+(0.1,1)+(0.1,1)+(0.1,1) -1]$$

$$= \text{Step} [0,1]$$

$$P_{15}=1, W= 1,1$$

$$X_1=0, X_2=0, X_3=1, X_4=0, X_5=0, X_6=0, X_7=0, X_8=1$$

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(1).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+ (0.1,1)+(1.1,1) -1] \\ &= \text{Step} [0,1] \end{aligned}$$

$$P_{16}=1, W=1,1$$

$$X_1=0, X_2=0, X_3=1, X_4=0, X_5=0, X_6=0, X_7=0, X_8=1$$

$$\dot{O}=1,$$

$$\begin{aligned} Y(1) &= \text{Step}[X_1.(0).W_1.(1)+X_2.(0).W_2.(1)+X_3.(1).W_3.(1)+X_4.(0).W_4.(1)+ \\ &\quad X_5.(0).W_5.(1)+X_6.(0).W_6.(1)+X_7.(0).W_7.(1)+X_8.(1).W_8.(1) -1] \\ &= \text{Step} [(0.1,1)+(0.1,1)+(1.1,1)+(0.1,1)+(0.1,1)+(0.1,1)+ (0.1,1)+(1.1,1) -1] \\ &= \text{Step} [0,1] \end{aligned}$$

Analisa Hasil Pelatihan Jaringan Syaraf Tiruan

Proses terakhir dari pelatihan Jaringan Syaraf Tiruan adalah dengan membandingkan keempat pola yang telah dilakukan pelatihan sebelumnya. Sehingga setelah keempat pola dikumpulkan maka akan dilihat pola mana yang memiliki nilai *error* yang rendah. Jika pola tersebut memiliki nilai *error* yang rendah, maka pola tersebut akan dilanjutkan ke proses pengujian.

Dengan dilakukannya kombinasi pada pelatihan diatas dapat dilihat bahwa nilai *error* tidak tergantung pada nilai dari *learning rate* dan momentum hanya membuat sedikit perubahan pada pelatihan. Jumlah *neuron* pada *hidden layer*lah yang mempengaruhi pada pelatihan jaringan syaraf tiruan ini.

Korelasi atau hubungan dari pelatihan ini dengan pewarisan golongan darah dilihat dengan hasil akhir masing-masing pelatihan. Sehingga dengan adanya pelatihan ini, akan mempermudah proses analisa kebenaran hasil perkawinan antar golongan darah.

Implementasi Dan Pengujian

Bagian ini merupakan lanjutan dari proses pelatihan yang dilakukan pada bagian sebelumnya. Proses selanjutnya yang akan dilakukan pada bab implementasi dan hasil adalah proses pengujian Jaringan Syaraf Tiruan dengan

menggunakan data yang dipersiapkan untuk diuji. Proses Pengujian dilakukan dengan menggunakan *software* matlab 7.0

Matlab merupakan perangkat lunak yang cocok dipakai sebagai alat komputasi yang melibatkan penggunaan matriks dan vektor. Matlab adalah singkatan dari *Matrix Laboratory*, merupakan bahasa pemrograman yang dikembangkan oleh *The MathworkInc*. Matlab dibuat untuk memudahkan perhitungan persamaan linear, program linier, sistem yang kompleks seperti pengolahan citra dan lain sebagainya. Matlab menyediakan fungsi-fungsi khusus untuk menyelesaikan model jaringan syaraf tiruan, pemakai hanya tinggal memasukan vektor masukan, target, model dan parameter yang diinginkan.

Tujuan Pelatihan dan Pengujian

Proses pelatihan digunakan sebagai bagian dari proses penentuan *output* dari sebuah jaringan yang mana setiap pelatihan akan berguna untuk proses pengujian. Sehingga proses pelatihan akan mempengaruhi setiap hasil pada proses pengujian.

Adapun proses dari implementasi dan hasil adalah berupa pengujian, yang mana bertujuan :

1. Melihat hasil dari Jaringan Syaraf Tiruan yang di latih. Sehingga dapat diketahui *output* dari Jaringan Syaraf Tiruan dapat digunakan oleh penulis untuk menguji kebenaran hasil perawinan antar golongan darah.
2. Untuk mengetahui kinerja jaringan pada data uji yang sesungguhnya apakah sudah tepat dengan hasil yang diharapkan.

Implementasi Sistem

Untuk pembagian data yang digunakan adalah 50 % untuk proses pelatihan dan 50 % untuk proses pengujian, yang mana nantinya data akan dilatih adalah sebanyak 16 data dari 32 data dan 8 buah *input*. Data yang akan dilatih dapat dilihat pada tabel data pengujian dan pelatihan di bawah ini.

Tabel 5.1 Data Pelatihan Dan Pengujian

0	0	1	0	0	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0	1
0	1	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	1	0	1	0	0
0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1	0	0

Analisa tersebut menjelaskan sample data golongan darah yang sudah dirubah dalam bentuk matrik 0/1. Dengan dilengkapi data pembandingan dan target.

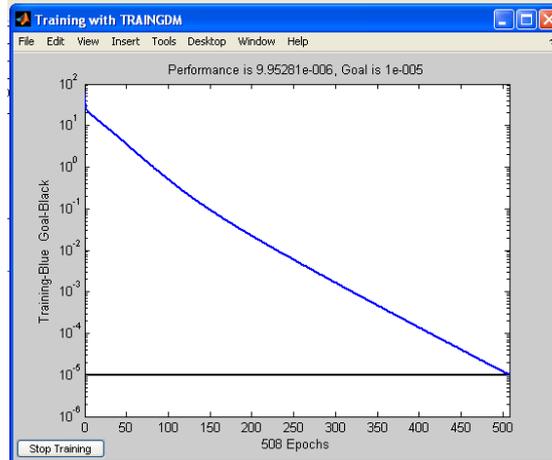
Pelatihan Secara Software Menggunakan Pola 1

Pelatihan dengan menggunakan pola 1 adalah dengan rincian sebagai berikut :

- Pelatihan menggunakan 4 *hidden layer*
- Frekuensi 8000 Hz
- *Learning rate* sebesar 0.01
- Momentum sebesar 0.1

- *Max epoch* sebanyak 10.000 *epoch*

Dengan menggunakan *learning rate* 0,01 dan momentum, 0,1 hasil (*goal*) diperoleh yaitu pada *epoch* ke 508 dari total 1000 *epoch*. Proses pelatihan dengan menggunakan *syntx* pada matlab yaitu dengan `:net=train(net,pn,tn)`. Grafik dari Jaringan Syaraf Tiruan pola 1 dapat dilihat pada gambar 5.1.



Gambar 5.1 Training dengan TRAINGDA Pola 1

Arsitektur hasil pengujian *software*

numInputs: 2
 numLayers: 4
 biasConnect: [1; 1; 1; 1]
 inputConnect: [1; 0; 0; 0]
 layerConnect: [4x4 boolean]
 outputConnect: [0 0 0 1]
 targetConnect: [0 0 0 1]

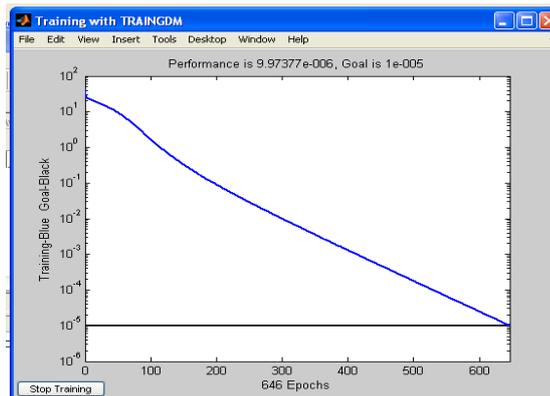
Pelatihan Secara Software Menggunakan Pola 2

Pelatihan dengan menggunakan pola 2 dengan rincian sebagai berikut :

- Pelatihan menggunakan 4 *hidden layer*
- Frekuensi 8000 Hz
- *Learning rate* sebesar 0.01
- Momentum sebesar 0.5
- *Max epoch* sebanyak 10.000 *epoch*

Dengan menggunakan *learning rate* 0,01 dan momentum 0,5 hasil (*goal*) diperoleh yaitu pada *epoch* ke 646 dari total 1000 *epoch*. Proses pelatihan dengan menggunakan *syntx* pada matlab yaitu dengan `:net=train(net,pn,tn)`. Grafik dari Jaringan Syaraf Tiruan pola 2 dapat dilihat pada gambar 5.2

Grafik dari Jaringan Syaraf Tiruan pola 2 dapat dilihat pada gambar 5.2.



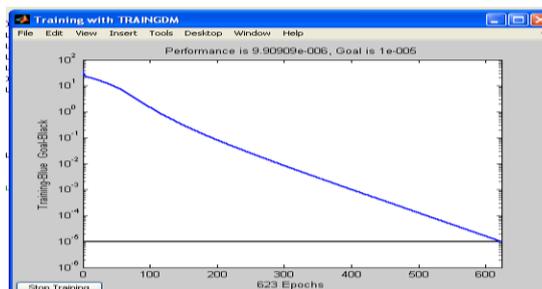
Gambar 5.2 Training dengan TRAINGDA Pola 2

Pelatihan Secara Software Menggunakan Pola 3

Pelatihan dengan menggunakan pola 3 dengan rincian sebagai berikut :

- Pelatihan menggunakan 4 *hidden layer*
- Frekuensi 8000 Hz
- *Learning rate* sebesar 0.1
- Momentum sebesar 0.1
- *Max epoch* sebanyak 10.000 *epoch*

Dengan menggunakan *learning rate* 0,1 dan momentum 0,1 hasil (*goal*) diperoleh yaitu pada *epoch* ke 623 dari total 1000 *epoch*. Proses pelatihan dengan menggunakan *syntx* pada matlab yaitu dengan `:net=train(net,pn,tn)`. Grafik dari Jaringan Syaraf Tiruan pola 3 dapat dilihat pada gambar 5.3



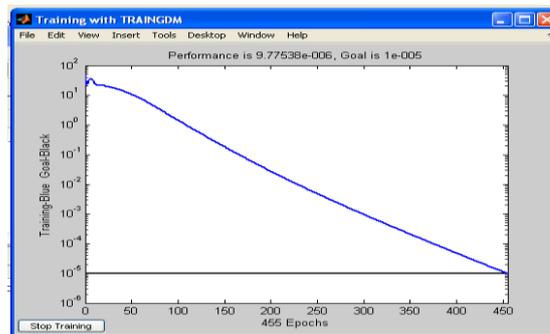
Gambar 5.3 Training dengan TRAINGDA Pola 3

Pelatihan Secara Software Menggunakan Pola 4

Pelatihan dengan menggunakan pola 4 dengan rincian sebagai berikut :

- Pelatihan menggunakan 3 *hidden layer*
- Frekuensi 8000 Hz
- *Learning rate* sebesar 0.1
- Momentum sebesar 0.8
- *Max epoch* sebanyak 10.000 *epoch*

Dengan menggunakan *learning rate* 0,1 dan momentum 0,8 hasil (*goal*) diperoleh yaitu pada *epoch* ke 455 dari total 1000 *epoch*. Proses pelatihan dengan menggunakan *syntx* pada matlab yaitu dengan `:net=train(net,pn,tn)`. Grafik dari Jaringan Syaraf Tiruan pola 4 dapat dilihat pada gambar 5.4



Gambar 5.4 Training dengan TRAINGDA Pola 4

Dari pelatihan *software* diatas dapat disimpulkan sebagai berikut:

Pola	Learning Rate	Momen tum	Epoch	Output	Error
	<i>Lr</i>	<i>M</i>			<i>e</i>
1	0.01	0.5	686	0.9579479	0.0420521
2	0.01	0.1	508	0.9951145	0.0048855
3	0.1	0.1	623	0.9954918	0.0045082
4	0.1	0.8	455	0.9948947	0.0051053

Tabel 5.2 Hasil Pelatihan

Dari penjelasan tabel 5.2 dapat disimpulkan bahwa, dengan menggunakan pelatihan pola 3 yaitu *Learning Rate* 0.1, *momentum* 0.1 dapat menghasilkan *output* yang lebih baik dibandingkan dengan pola pelatihan yang lainnya. Dengan hanya menyisakan *error* yang lebih kecil dengan target yang ditentukan.

DAFTAR PUSTAKA

- Ardi Pujiyanta, *Pengenalan Citra Objek Sederhana Dengan Jaringan Syaraf Tiruan Metode Perceptron*, Jurnal Informatika Vol.3.No.1. Januari 2009.
- Arief Heru Kuncoro dan Rinaldy Dalimi, *Aplikasi Jaringan Syaraf Tiruan Untuk Peramalan Beban Tenaga Listrik Jangka Panjang Pada Sistem Kelistrikan Di Indonesia*, Jurnal Teknologi, Edisi No.3. Tahun XIX, Sept.2005: 211-217.
- Eli Yani, *Pengantar Jaringan Syaraf Tiruan*, materikuliaah.com,2005.
- Elly Warni, *Penentuan Morfologi Sel Darah Merah(Eritrosit) Berbasis Pengolahan Citra Dan Jaringan Syaraf Tiruan*, Elektrikal Enjiniring Vol.07.No.3. Oktober-Desember:2009
- Kusumadewi, Sri, *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta, 2003
- Setyo Nugroho, Agus Harjoko, *Penerapan Jaringan Syaraf Tiruan Untuk Mendeteksi Posisi Wajah Manusia Pada Citra Digital*, Seminar Nasional Teknologi Informasi 2005 (SNATI 2005) Yogyakarta, 18 Juni 2005.
- Siang, Jong Jek, *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan Matlab*, Andi, Yogyakarta, 2005
- Wahidin Wahab, *Aplikasi Algoritma Back-Propagation Multi Layer Perceptron Pada Identifikasi Dinamika Sistem Nonlinier*, Jurnal Teknologi, Edisi No.1. Tahun XVIII, Maret 2004: 1-62.
- Wiryanawan, Wiko, *Aplikasi Pohon Dalam Penentuan Golongan Darah Manusia*, makalah IF2091 Struktur Diskrit – Sem. I:2010/2011.