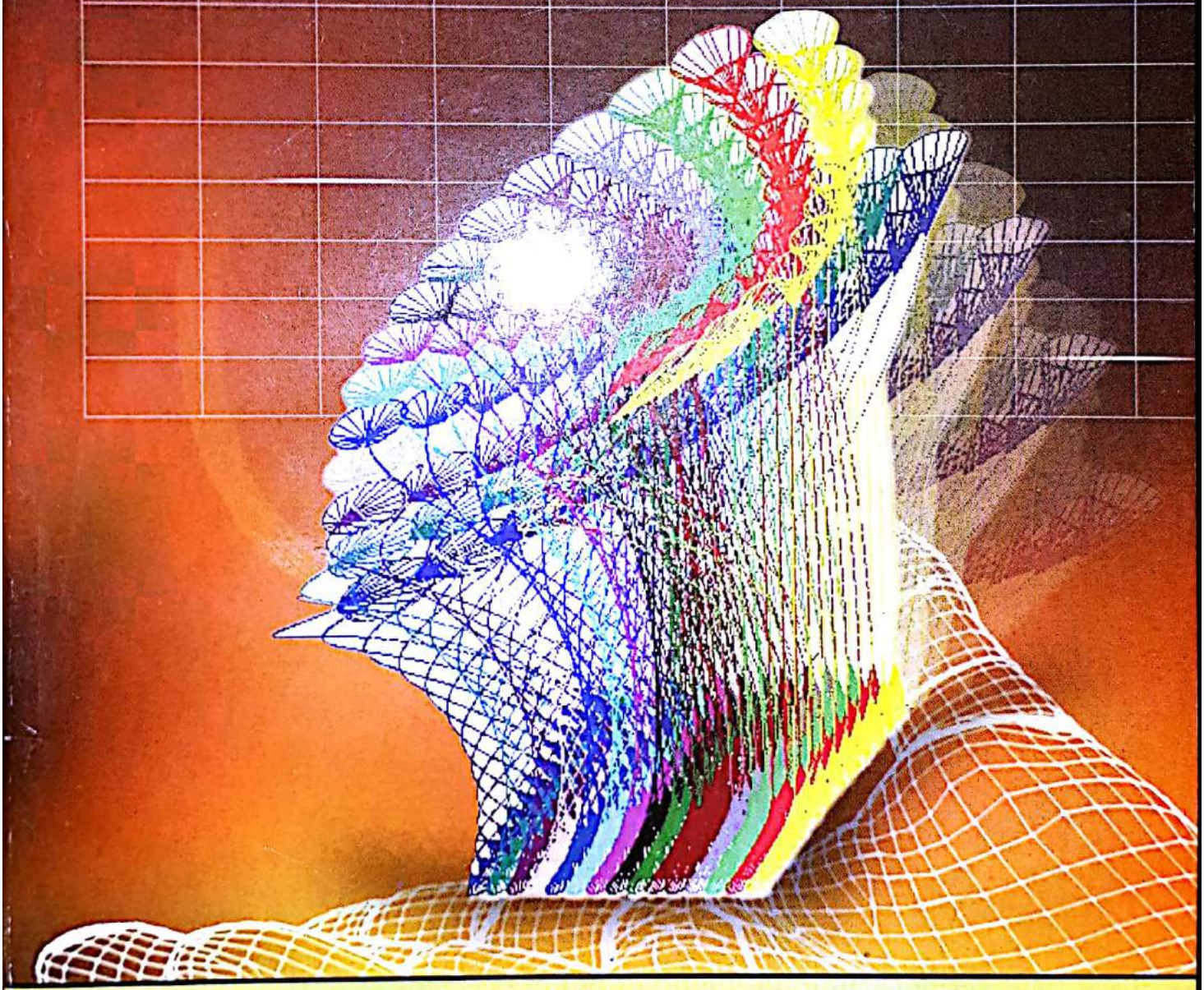
KAPITA SELEKTA STATISTIKA
NEURAL NETWORK



Oleh :
Budi Warsito, S.Si., M.Si.

BP UNDIP
SEMARANG

Kapita Selekta Statist
Neural Network

Oleh :
Budi Warsito, S.Si., M.Si.

Perpustakaan Nasional / Katalog Dalam Terbitan **(KDT)**

Kapita Selekta Statistika Neural Network
oleh : Budi Warsito, S.Si, M.Si
Fakultas MIPA Universitas Diponegoro Semarang.
halaman; viii,184 hlm. uk; 23 x 15,5 cm

Kapita Selekta Statistika Neural Network

ISBN : 978 979 704 857-0

Cetakan Pertama, Desember 2009

Desain Cover : Thomas

Setting Lay Out : Tim Redaksi Widya Karya

Diterbitkan oleh : Badan Penerbit UNDIP Semarang

Untuk istriku tercinta dan anak

Husni Fadhi
Hasna Faria
Hanifah Khoirunn
Haidar Fa
Hashif Izzu

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan ke had
karena atas segala rahmat dan karunia-Nya sehingg
menyelesaikan penyusunan buku ini. Buku menge
neural network dan statistika ini diharapkan dapat
semua pihak yang berkepentingan.

Penulis menyadari bahwa penyusunan buku i
berkat bantuan, dorongan dan nasehat dari berbagai pih
itu pada kesempatan yang baik ini penulis menguca
kasih yang sebesar-besarnya kepada semua pihak yang
hingga terselesaikannya penulisan buku ini.

Penulis menyadari bahwa segala kemudahan
yang penulis peroleh selama ini semata-mata datang
Maha Kuasa. Semoga segala upaya makhluk yang ta
mendapatkan rahmat dan ridho-Nya. Semoga tu
memberikan manfaat yang sebesar-besarnya bagi
khususnya dan bagi ilmu pengetahuan pada umumnya.

Semarang,

DAFTAR ISI

POKOK BAHASAN I

DASAR-DASAR PEMODELAN NE NETWORK

I.1 PENGANTAR NEURAL NETWORK

1.1 Pendahuluan

Pada sub pokok bahasan ini akan dibahas p neural network kaitannya dengan sistem manusia. Dibahas pula beberapa istilah baru d pemodelan neural network sebagai suatu sistem buatan, termasuk jenis-jenis proses pembelajara jaringan.

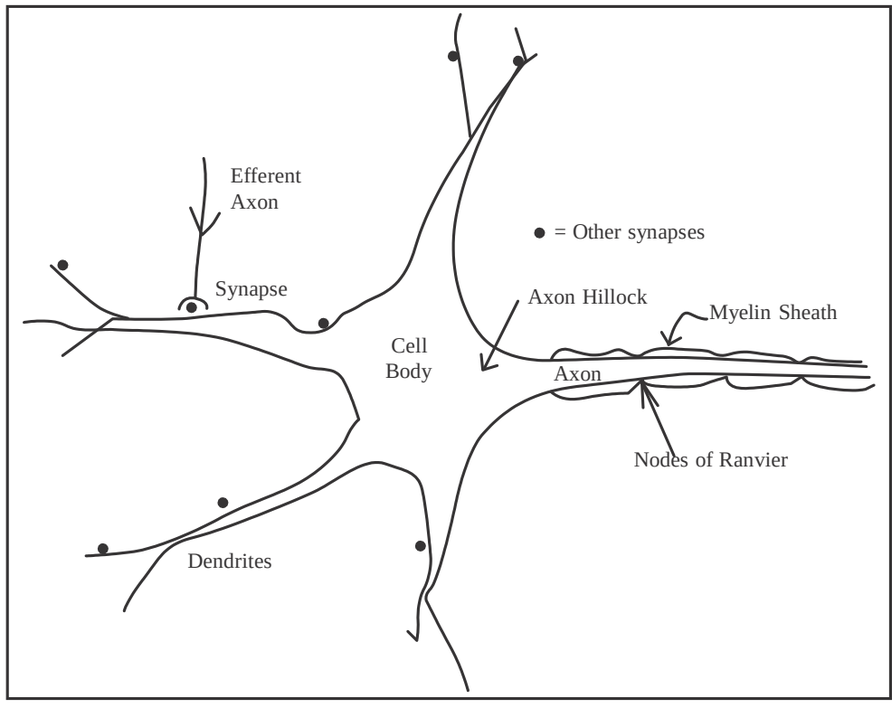
1.2 Penyajian

Jaringan Syaraf Tiruan (JST) atau yang *Artificial Neural Network (ANN)* merupakan s informasi yang memiliki karakteristik mirip d syaraf pada makhluk hidup (Fausset, perkembangan selanjutnya huruf "A" pada istil dihilangkan untuk mempersingkat penulisan populer dengan istilah Neural Network (NN) sa kali diperkenalkan oleh McCulloch dan Pitts (194 suatu model sederhana dari suatu syaraf nya manusia seperti suatu unit *threshold* yan dikembangkan sebagai model matematika ya pola pikir manusia atau jaringan syaraf makhluk asumsi bahwa :

1. Proses informasi terjadi pada banyak elemen disebut *neuron*.
2. Sinyal yang melewati antar *neuron* menggun tertentu.
3. Setiap penghubung antar *neuron* mempunya yang bersesuaian dengan mengalikan sinyal y Bobot ini dapat memperkuat maupun memper
4. Setiap *neuron* menggunakan fungsi aktivasi (jumlah sinyal input yang terboboti) untuk me output.

pemodelan NN pada saat ini secara luas banyak dilakukan dan dimotivasi oleh adanya kemungkinan untuk menggunakan NN sebagai suatu instrumen untuk menyelesaikan berbagai permasalahan aplikasi seperti *pattern recognition* dan *signal processing*, sedangkan pada pemodelan statistika banyak diterapkan pada masalah klasifikasi, clustering, regresi dan juga peramalan data *time series*.

Secara sederhana gambar sebuah sistem jaringan syaraf biologis manusia disajikan pada gambar 1.1.



Gambar 1.1 Sistem Jaringan Syaraf Manusia

Neural Network merupakan sebuah mesin pembelajaran yang dibangun dari sejumlah elemen pemrosesan sederhana yang disebut *neuron* atau *node* (Fausett, 1994). Istilah *unit* juga sering digunakan untuk menyebutkan neuron. Setiap neuron dihubungkan dengan neuron yang lain dengan link komunikasi langsung melalui pola hubungan yang disebut arsitektur jaringan. Bobot-bobot pada koneksi mewakili besarnya informasi yang digunakan jaringan. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakan dengan algoritma pembelajaran. Setiap *neuron* mempunyai tingkat

dan hanya dapat mengirim sekali dalam satu w
sinyal tersebut disebarkan pada beberapa neuron

Prinsip dari pemodelan *Neural Network* dik
karakteristik dan cara kerja otak manusia,
memproses informasi otak manusia terdiri dari
yang melakukan tugas sederhana. Ka
keterhubungan antar *neuron*, maka otak
melakukan fungsi pemrosesan yang san
Pemrosesan informasi hanya dapat dilakukan
proses pembelajaran sebelumnya. Pemrosesan
jaringan syaraf manusia bersifat adaptif, yang ar
antar *neuron* terjadi secara dinamis, kekuatan
neuron dapat berubah dari waktu ke wakt
mempunyai kemampuan untuk mempelaj
informasi baru.

Dalam memproses informasi, jaringan
memiliki 3 elemen dasar sebagai berikut :

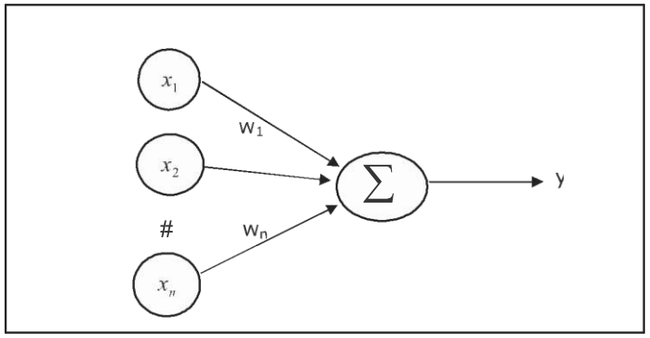
1. Himpunan Penghubung
   Himpunan Penghubung merupakan himp
   yang dihubungkan dengan suatu jalur kon
   tersebut memiliki bobot yang berbeda. Bob
   positif akan memperkuat sinyal sedangk
   bernilai negatif akan memperlemah sinyal. J
   dan pola hubungan antar unit-unit tersebut al
   arsitektur jaringan yang terbentuk.
2. Fungsi Penjumlah
   Fungsi Penjumlah merupakan suatu u
   menjumlahkan input-input sinyal yang su
   dengan bobot masing-masing.
3. Fungsi Aktivasi
   Fungsi aktivasi dapat digunakan sebaga
   mengubah bobot, yang dapat menjadikan ja
   lebih powerful. Fungsi aktivasi juga mengha
   yang akan dikirim ke neuron lain dengan bo
   tertentu.

Pada gambar 1.2 ditunjukkan struktur
input x_1, x_2,…,x_n yang bersesuaian dengan sinya
dalam saluran penghubung. Setiap sinyal yang
dengan bobot koneksinya yaitu w_1 w_2 … w_n seb

akan menjumlahkan semua input terbobot dan menghasilkan sebuah nilai yaitu y_in.

$$Y\_in = = x_1w_1 = x_1w_1 + x_2w_2 + \ldots + x_nw_n$$

Aktivasi *y* dari neuron *Y* ditentukan oleh fungsi input jaringannya, *y = f(y_in)* dimana *f* merupakan fungsi aktivasi yang digunakan.



Gambar 1.2 Struktur neuron buatan pada Jaringan Syaraf Tiruan dengan input $x_1, x_2, \ldots, x_n$ dan bobot koneksinya $w_1, w_2, \ldots, w_n$.

Karakteristik dari model neural network antara lain :

1. Memiliki kemampuan menghasilkan output terhadap pola yang belum pernah dipelajari *(generalization)*
2. Memiliki kemampuan untuk memproses *input* yang terdapat kesalahan di dalamnya dengan tingkat toleransi tertentu.
3. Mampu beradaptasi dengan perubahan yang terjadi terhadap nilai–nilai input dan output. Bentuk adaptasi ini diwujudkan dalam perubahan nilai bobot.

Secara garis besar neural network mempunyai dua tahap pemrosesan informasi, yaitu tahap pelatihan dan tahap pengujian.

1. Tahap Pelatihan

   Tahap pelatihan dimulai dengan memasukkan pola–pola pelatihan (data latih) ke dalam jaringan. Dengan menggunakan pola–pola ini jaringan akan mengubah–ubah bobot yang menjadi penghubung antar *node*. Pada setiap iterasi (dalam NN dikenal sebagai epoch) dilakukan evaluasi

bobot yang sesuai dimana nilai error yang
tercapai atau jumlah iterasi telah mencapai
yang ditetapkan sebelumnya. Selanjutnya bo
dasar pengetahuan pada tahap pengujian.

Berdasarkan permasalahan yang ditangani,
pada neural network dapat dibedakan menjad

a. Pelatihan Terawasi *(Supervised training)*

   Pada proses pelatihan terawas
   diharapkan dari input yang diterima
   diketahui sebelumnya. Metode pelatihan i
   jaringan untuk dapat mengenali pola m
   cara memetakan pola masukan (input) k
   (output) yang diinginkan. Tujuan dari pe
   adalah untuk menghasilkan pemetaan dar
   nilai output, sehingga akan diperoleh out
   dengan targetnya. Pengubahan nilai
   pembelajaran) dilakukan untuk mencapai t

   Setiap input jaringan akan me
   outputnya dan selanjutnya output
   dibandingkan dengan vektor target ya
   Perbedaan antara output yang dihasilka
   output disebut sebagai error. Error
   dikembalikan ke dalam jaringan sebag
   menentukan bobot koneksi yang baru. Bob
   neuron akan diperbaiki berdasarkan algo
   tertentu, sampai seluruh rangkai
   menghasilkan error yang sedikit (minima
   pemodelan statistika, neural network deng
   terawasi ini dapat diterapkan pada beberap
   seperti pada model-model regresi, tir
   klasifikasi.

b. Pelatihan Tak Terawasi *(Unsupervised Train*

   Pelatihan tak terawasi dite
   permasalahan–permasalahan dimana o
   tidak diketahui sebelumnya. Pada metode
   nilai bobot dilakukan dengan sen
   menggunakan output yang diinginkan. Tu
   mengelompokkan unit–unit dengan ciri–c

Pelatihan tak terawasi tidak membutuhkan vektor target untuk output-nya, sehingga tidak ada pembandingan yang dilakukan dengan target yang ditentukan sebelumnya. Rangkaian pelatihan hanya berisi vektor input saja. Jaringan akan menempatkan input-input ke dalam beberapa kelompok Selanjutnya dari kelompok-kelompok tersebut ditentukan karakteristik kelompoknya, sehingga apabila jaringan menerima input yang baru dapat ditentukan ke dalam kelompok dimana input tersebut berada. Jaringan memperbaiki bobot koneksi antar neuron sedemikian sehingga pola-pola input yang mirip ditempatkan ke dalam kelompok yang sama. Kemudian jaringan akan membuat pola-pola input sebagai wakil bagi setiap kelompok yang terbentuk. Salah satu penerapan pada statistika adalah pada analisis cluster.

2. Tahap Pengujian
   Pada tahap ini dilakukan pengujian terhadap suatu pola masukan yang belum pernah dilatihkan sebelumnya (data uji) menggunakan bobot-bobot yang telah dihasilkan pada tahap pelatihan. Diharapkan bobot-bobot hasil pelatihan yang sudah menghasilkan error minimal juga akan memberikan error yang kecil pada tahap pengujian.

Dalam pemodelan Neural Network, kemampuan jaringan dalam menghasilkan model dipengaruhi oleh banyak faktor. Beberapa faktor yang berpengaruh terhadap kebagusan model diantaranya adalah sebagai berikut :

1. Arsitektur jaringan
2. Fungsi aktivasi yang digunakan pada *hidden layer* dan *output layer*
3. Metode pembelajaran yang digunakan
4. Input jaringan

Pemode
taranya adalah sebagai berikut :

1. Pengenalan Pola *(Pattern Recognition)*
   Model Neural Network dapat dipakai untuk mengenali pola (misalnya huruf atau angka) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih

seseorang yang belum dikenal hanya berd
yang diketahui.

2. *Signal Processing*
   Model Neural Network dapat dipakai
   noise dalam saluran telepon.
3. Peramalan
   Model Neural Network dapat dipakai un
   hal yang akan terjadi di masa yang
   berdasarkan pola kejadian sejenis yang
   lampau. Dalam hal ini data masa lampa
   sedangkan prediksi masa depan sebagai o
   dilakukan mengingat kemampuan Neural
   mengingat dan membuat generalisasi dari
   ada sebelumnya.
4. Klasifikasi Data/ Prediksi Kelas
   Neural Network dapat dipakai
   pengklasifikasian data/prediksi kel
   merupakan proses untuk menemukan sek
   yang menjelaskan dan membedakan k
   sehingga model tersebut dapat dig
   memprediksi nilai suatu kelas yang belum
   sebuah objek.

Arsitektur Neural Network

Neuron-neuron dalam Neural Network
bentuk lapisan–lapisan (layer) dan
keterhubungan baik dalam satu lapisan
lapisan. Susunan dari neuron-neuron dalam
keterhubungannya dalam dan antar lapisan
arsitektur jaringan. Neuron-neuron yang be
lapisan tertentu akan mempunyai pola keter
sama (Fausset, 1994). Suatu jaringan syaraf
atas lapisan-lapisan :

1. Lapisan input *(Input Layer)*
   Yaitu lapisan yang menerima masukan/in
   luar.
2. Lapisan output *(Output Layer)*

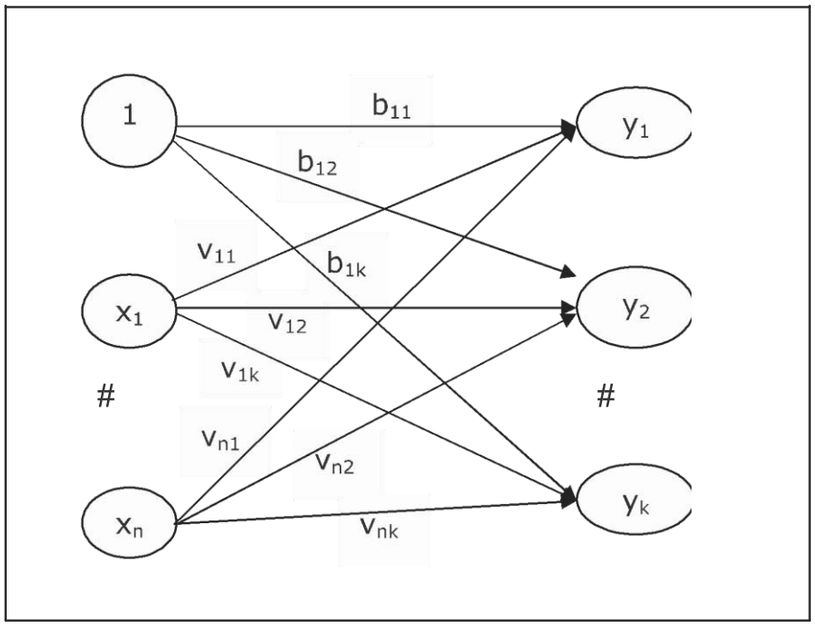
Yaitu lapisan yang menghasilkan output dari j

3. Lapisan tersembunyi *(Hidden Layer)*
   Yaitu lapisan yang terletak diantara lap

dan tidak berhubungan langsung dengan keadaan di luar jaringan.

Dalam hal ini lapisan input tidak melakukan proses perhitungan, hanya mendistribusikan sinyal ke semua neuron pada lapisan berikutnya sehingga untuk menentukan jumlah lapisan dalam suatu jaringan, lapisan input tidak dihitung karena tidak dianggap sebagai lapisan. Berdasarkan jumlah lapisan yang dimiliki neural network dapat dibedakan menjadi Jaringan Lapisan Tunggal *(Single Layer)* dan Jaringan Multi Lapis *(Multi Layer)*.

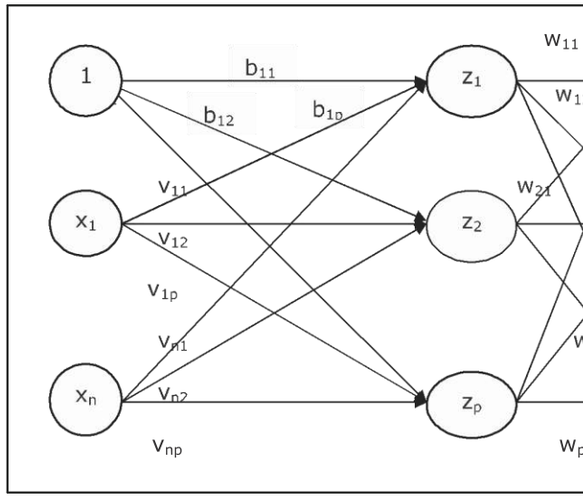
1. Jaringan Lapisan Tunggal *(Multi Layer)*
   Sebuah jaringan disebut jaringan lapis tunggal jika jaringan tersebut tidak mempunyai lapisan tersembunyi atau hanya mempunyai satu lapisan bobot koneksi (Fausett, 1994). Pada jaringan ini sekumpulan input neuron dihubungkan langsung dengan sekumpulan output (Siang, 2005). Arsitektur dari jaringan lapis tunggal digambarkan pada gambar 1.3 sebagai berikut:



Gambar 1.3 Jaringan lapis tunggal dengan n unit input (x_1, $x_2,\ldots,x_n$) dan k unit output (y_1, $y_2,\ldots,y_k$).

2. Jaringan Multilapis *(Multilayer)*

Jaringan multilapis merupakan perluas
single layer (Siang, 2005). Jaringan multilapis te
input *(input layer)*, lapisan tersembunyi *(hidden l*
output *(output layer)*. Lapisan tersembunyi t
lapisan input dan lapisan output. Output dari
tersembunyi akan menjadi input bagi lapi
Jaringan ini paling tidak mempunyai satu lapis
Arsitektur jaringan multilapis digambarkan p
sebagai berikut :



Gambar 1.4 Jaringan multilapis dengan n
$x_2,\ldots,x_n$); p unit tersembunyi ($z_1,z_2,$.
output (y_1, $y_2,\ldots,y_k$).

Latihan

1. Jelaskan jargon-jargon dalam pemodelan
   berikut :
   a. unit
   b. input
   c. output
   d. target
   e. arsitektur j
   f. bobot
   g. algoritma
   h. epoch
2. Jelaskan perbedaan pelatihan dengan pe
   pelatihan tanpa pengawasan
3. Apa kegunaan hidden layer dalam jaringan

1.3 Penutup

Pemodelan Neural Network dilandasi oleh algoritma yang bersumber pada sistem kerja jaringan syaraf manusia yang bekerja dengan melakukan pemrosesan berdasarkan informasi/sinyal yang masuk. Input model sebagai sumber informasi diproses dengan melakukan pembobotan terhadap koneksi antar neuron. Output model adalah jumlahan terboboti dari seluruh sinyal yang masuk seteleh kemudian dilakukan aktivasi. Dengan proses pelatihan terhadap jaringan diharapkan output yang dihasilkan akan memberikan hasil yang sesuai dengan target.

I.2 Fungsi Aktivasi

2.1 Pendahuluan

Pada sub pokok bahasan ini akan dibahas beberapa fungsi aktivasi yang sering digunakan pada pemodelan neural network. Sebagai aplikasi disajikan beberapa contoh fungsi aktivasi yang disertai dengan gambar fungsinya menggunakan bahasa pemrograman Matlab versi 7.1.

2.2 Penyajian

Pada dasarnya, Neural Network merupakan suatu kumpulan dari elemen-elemen pemrosesan sederhana yang saling berhubungan, yang disebut *neuron* (*unit, sel* atau *node*). Setiap neuron dihubungkan dengan neuron yang lain dengan link komunikasi langsung melalui pola hubungan yang disebut arsitektur jaringan. Tiap-tiap hubungan tersebut mempunyai bobot koneksi *(weight)* yang dilatih untuk mencapai respon yang diinginkan. Bobot-bobot pada koneksi mewakili besarnya informasi yang digunakan jaringan. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakan dengan algoritma pembelajaran. Masing-masing bobot koneksi dipropagasikan ke seluruh unit atau node. Dengan pelatihan terhadap data berdasarkan bobot-bobot koneksi tersebut diharapkan dapat diperoleh output sesuai dengan yang diinginkan.

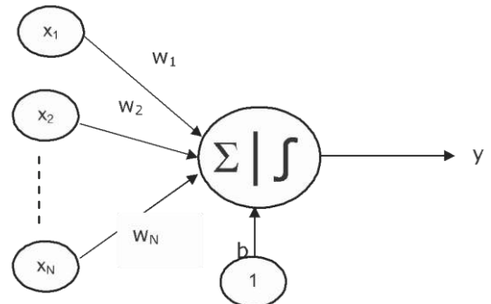
Setiap neuron mempunyai tingkat
merupakan fungsi dari input yang masuk pa
yang dikirim suatu neuron ke neuron yang lain b
hanya dapat mengirim sekali dalam satu waktu,
tersebut disebarkan pada beberapa neuron lain.
neuron Y menerima input dari neuron-neuro
dengan aktivasi (sinyal) masing-masing x_1, x_2,
koneksi w_1, w_2, …, w_n. Input jaringan y_in pada
jumlahan dari sinyal-sinyal terboboti yaitu

$$y\_in = \sum_{i=1}^{n} w_i x_i$$

Aktivasi y dari neuron Y ditentukan oleh
jaringannya, $y = f(y\_in)$ dimana f merupakan fun
digunakan.

Misalkan neuron Y dihubungkan dengan
bobot v, maka neuron Y mengirimkan sinya
tersebut. Dalam hal ini aktivasi z dari neuron Z
pada beberapa neuron, yaitu unit-unit input da
tersembunyi (*hidden unit*). Dengan adanya unit
jaringan untuk mencari bobot optimal menjadi
jaringan dengan fungsi aktivasi nonlinear pada
mempunyai kemampuan lebih dibandingkan
hanya mengandung input dan output saja.

Operasi dasar sebuah neuron buatan ya
suatu jaringan syaraf meliputi penjumlahan da
antara sinyal input dengan bobot antar ne
menggunakan fungsi aktivasi untuk mengh
neuron yang diinginkan. Dalam hal ini fungsi
untuk menetapkan output neuron yang mengind
aktifitas pada input neuron tersebut.



Ada beberapa fungsi aktivasi yang sering digunakan pada pemodelan NN sesuai dengan output yang menjadi tujuan pelatihan jaringan, antara lain disajikan pada bagian berikut. Untuk memudahkan pengerjaan sebaiknya penulisan setiap perintah disusun dalam m-file dan diberi nama sesuai dengan fungsi aktivasi yang diplot.

1. Fungsi hardlim *(Undak Biner)*

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

   Syntax : `y=hardlim(x)`

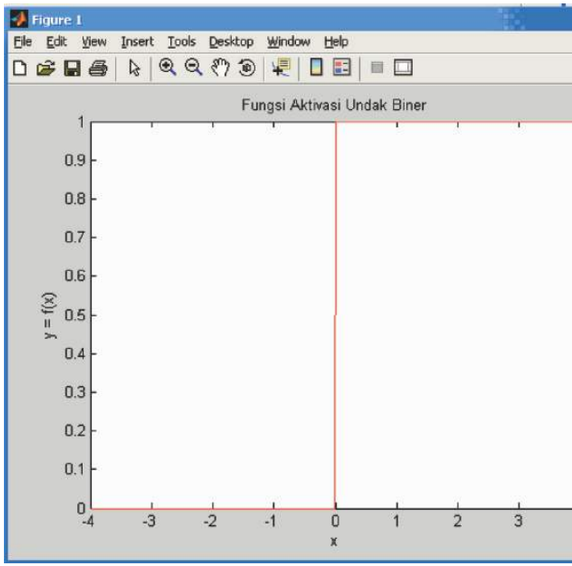
   Catatan : fungsi undak biner akan mengkonversi input dari variabel bernilai kontinu ke output biner 0 atau 1 dengan treshold 0.

   Contoh :
   ```
   x=-4:0.01:4;
   y=hardlim(x);
   plot(x,y,'r-')
   title('Fungsi Aktivasi Undak
   Biner')
   ylabel('y = f(x)')
   xlabel('x')
   ```

   Silahkan ditulis dalam script m-file dan beri nama fungsi yang tersusun sebagai undakbiner.m kemudian jalankan fungsi tersebut dengan mengetikkan `undakbiner` pada Command Window

   atau klik tanda berikut pada jendela editor :

Akan muncul tampilan berikut :



Gambar 1.7 Fungsi Aktivasi Undak

2. Fungsi Bipolar

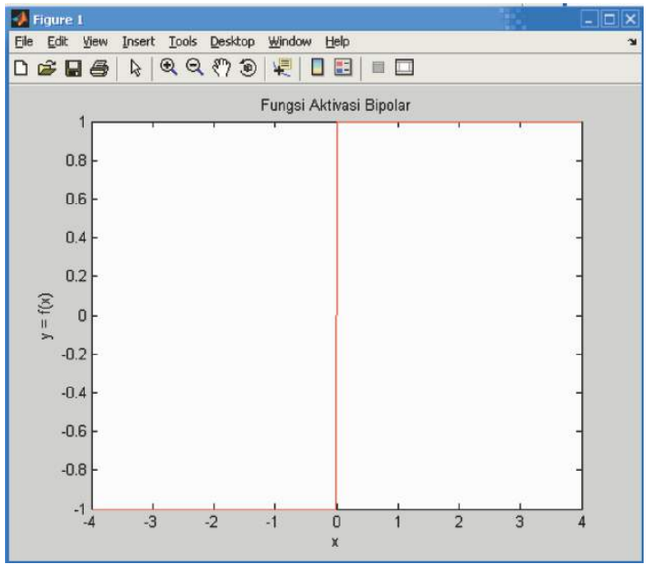
   Rumus : $y=\begin{cases}-1, & \text{jika } x<0\\ 1, & \text{jika } x\geq 0\end{cases}$

   Syntax : `y=hardlims(x)`

   Catatan : fungsi bipolar akan mengkonv
   variabel bernilai kontinu ke outp
   1 dengan treshold 0

   Contoh :
   ```
   x=-4:0.01:4;
   y=hardlims(x);
   plot(x,y,'r-')
   title('Fungsi Aktivasi B
   ylabel('y = f(x)')
   xlabel('x')
   ```

Silahkan ditulis dalam script m-file dan beri n
tersusun sebagai `bipolar.m` kemudian
tersebut, maka akan muncul tampilan berikut :



Gambar 1.8 Fungsi Aktivasi Bipolar

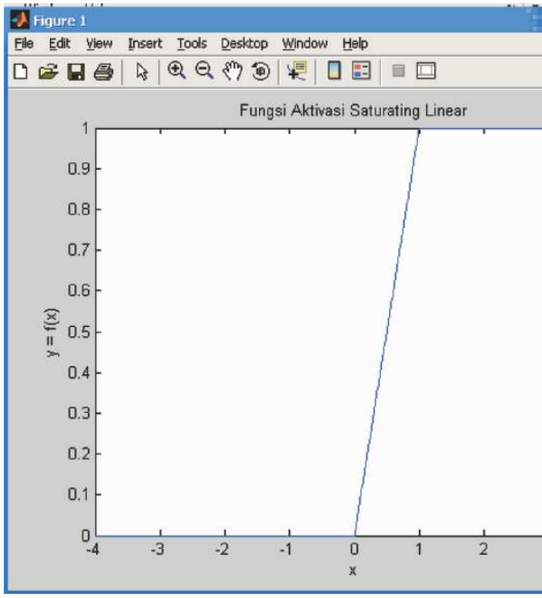
3. Fungsi Saturating linear

Rumus : $$y = \begin{cases} 0 & x \le -0.5 \\ x + 0.5 & -0.5 \le x \le 0.5 \\ 1 & x \ 0.5 \end{cases}$$

Syntax : `y=satlin(x)`

Contoh :
```
x=-4:0.01:4;
y=satlin(x);
plot(x,y,'b-')
title('Fungsi Aktivasi Saturating
Linear')
ylabel('y = f(x)')
xlabel('x')
```

Silahkan ditulis dalam script m-file dan beri nama fungsi yang tersusun sebagai `saturatinglin.m` kemudian jalankan fungsi tersebut, maka akan muncul tampilan berikut :



Gambar 1.9 Fungsi Aktivasi Saturatin

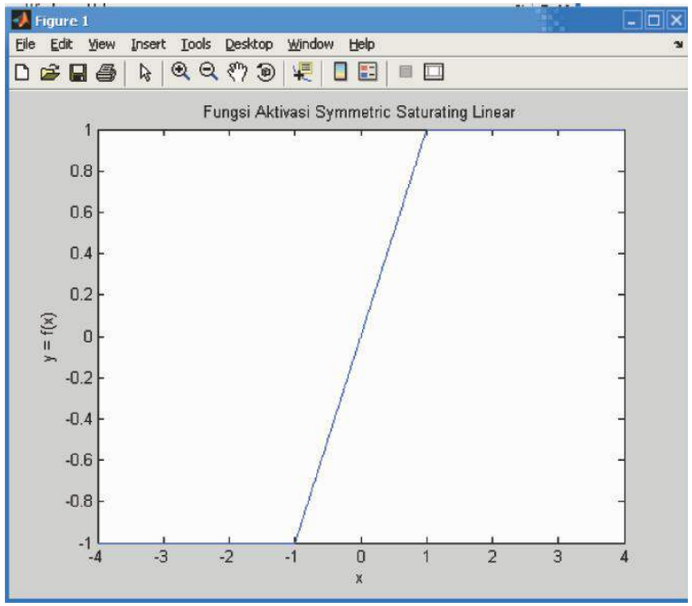
4. Fungsi Symmetric saturating linear

Rumus : $$y = \begin{cases} -1 & x \le -1 \\ x & -1 \le x \le 1 \\ 1 & x \ge 1 \end{cases}$$

Syntax : `y=satlins(x)`

Contoh :
```
x=-4:0.01:4;
y=satlins(x);
plot(x,y,'b-')
title('Fungsi Aktivasi S
Saturating Linear')
ylabel('y = f(x)')
xlabel('x')
```

Silahkan ditulis dalam script m-file dan beri n
tersusun sebagai symmetricsatlin.m kem
fungsi tersebut, maka akan muncul tampilan b



Gambar 1.10 Fungsi Aktivasi Symmetric Saturating Linear

5. Fungsi Linear (Identitas)

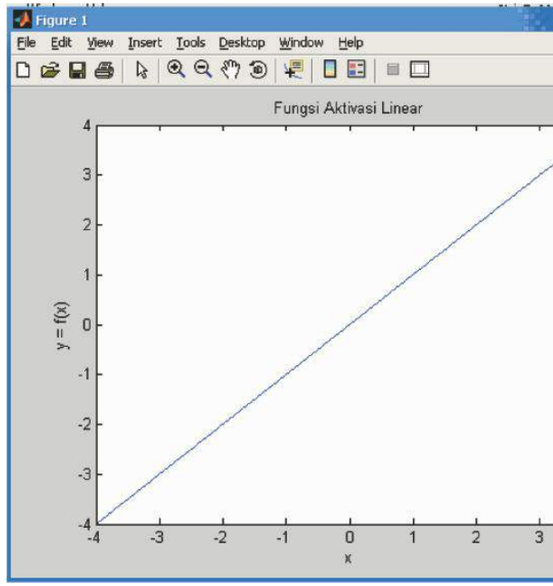
   Rumus : y = x

   Syntax : `y=purelin(x)`

   Catatan : Fungsi identitas akan menghasilkan output yang sama dengan inputnya. Fungsi ini sering dipakai jika diinginkan output jaringan berupa sebarang bilangan riil.

   Contoh :
```
x=-4:0.01:4;
y=purelin(x);
plot(x,y,'b-')
title('Fungsi Aktivasi Linear')
ylabel('y = f(x)')
xlabel('x')
```

Silahkan ditulis dalam script m-file dan beri nama fungsi yang tersusun sebagai linear.m kemudian jalankan fungsi tersebut, maka akan muncul tampilan berikut :



Gambar 1.11 Fungsi Aktivasi Line

6. Fungsi Sigmoid biner (logistic sigmoid)

   Rumus : $y=\frac{1}{1+e^{-\sigma x}}$ dengan $y'=\sigma y[1-$

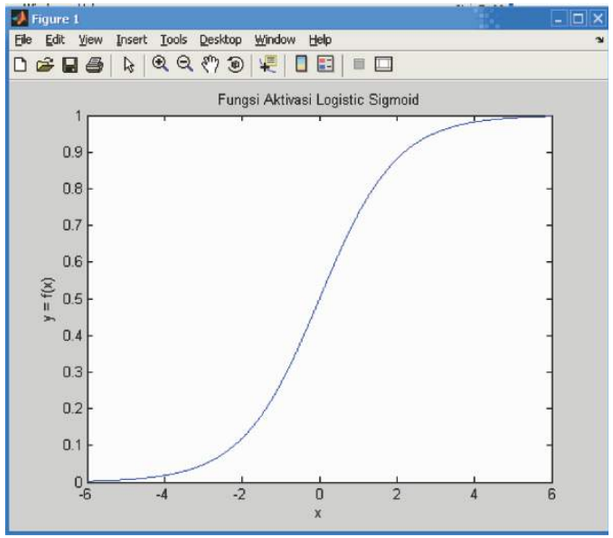
   Dalam hal ini adalah slope p
   nilainya dapat diubah-ubah.

   Syntax : `y=logsig(x)`

   Catatan : Fungsi ini bernilai antara 0 samp
   untuk jaringan dengan nilai outp
   1. Fungsi aktivasi ini umum di
   pelatihan jaringan syaraf
   menggunakan algoritma *backpr*
   hubungan yang sederhana ant
   pada suatu titik dan nilai der
   tersebut dapat mereduksi perh
   proses pembelajaran. Fungsi ini
   sebagai fungsi naik yang me
   asymptotic dan *smoothness*.

   Contoh :
```
x=-6:0.01:6;
y=logsig(x);
plot(x,y,'b-')
title('Fungsi Aktivasi I
Sigmoid')
```

Silahkan ditulis dalam script m-file dan beri nama fungsi yang tersusun sebagai logsigmoid.m kemudian jalankan fungsi tersebut, maka akan muncul tampilan berikut :



Gambar 1.12 Fungsi Aktivasi Logistic Sigmoid

1. Fungsi Sigmoid bipolar

Rumus : $y=\frac{1-e^{-\sigma x}}{1+e^{-\sigma x}}$ dengan $y'=\frac{\sigma}{2}y[1+y][1-y]$

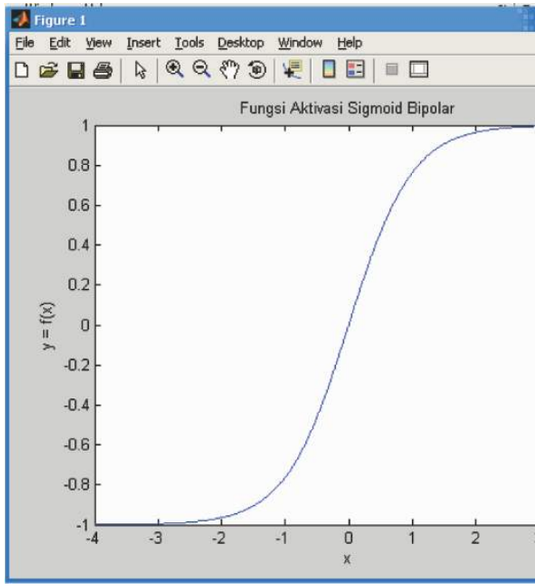
Syntax : `y=tansig(x)`

Catatan : Fungsi sigmoid bipolar mirip dengan fungsi logistic sigmoid dalam hal kekontinuan dan kemonotonan. Fungsi ini bernilai antara -1 sampai 1, digunakan untuk jaringan dengan nilai output antara -1 dan 1. Sering digunakan untuk jaringan yang dilatih dengan metode backpropagation.

Contoh :

```
x=-4:0.01:4;
y=tansig(x);
plot(x,y,'b-')
title('Fungsi Aktivasi Sigmoid
Bipolar')
ylabel('y = f(x)')
xlabel('x')
```

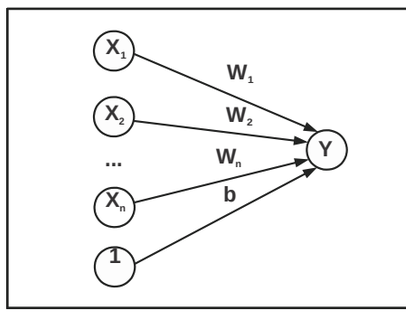
tersebut, maka akan muncul tampilan berikut :



Gambar 1.13 Fungsi Aktivasi Sigmoid

Bias dan Threshold

Dalam suatu jaringan syaraf tiruan seringk sebuah unit masukan (input) yang nilainya selal (satu). Unit yang demikian itu disebut sebagai bi 1.14). Bias dapat dipandang sebagai sebuah inp sama dengan 1. Dalam model linear nilai ini da sebagai slope (gradien) garis. Bias berfungsi u nilai threshold menjadi = 0 (bukan = S threshold berperan sebagai penimbang dalam dari sebuah unit tertentu.



Jika melibatkan bias, maka keluaran unit penjumlah adalah :

$$net = b + \sum_i x_i w_i$$

Sehingga fungsi aktivasi *threshold* menjadi :

$$f(net) = \begin{cases} 1, \text{ jika net} \geq \theta \\ -1, \text{ jika net} < \theta \end{cases}$$

Jika sebuah jaringan tidak menggunakan bias tetapi menggunakan nilai threshold tertentu dan tetap, maka fungsi aktivasi thresholdnya adalah :

$$f(net) = \begin{cases} 1, \text{ jika net} \geq \theta \\ -1, \text{ jika net} < \theta \end{cases}$$

dengan

$$net = \sum_i x_i w_i$$

Latihan

1. Jelaskan kegunaan fungsi aktivasi dari sebuah jaringan syaraf tiruan
2. Jelaskan pengertian dan kegunaan dari bias
3. Dari fungsi aktivasi logistic sigmoid pada sub bab 1.1, tunjukkan bahwa derivatif dari y = f(x) adalah : dy/dx = **ó y (1-y)**
   Berapa nilai turunannya di titik asal ?
4. Tunjukkan bahwa derivatif dari fungís tansig y = f(x) adalah : dy/dx = ½ **ó (1-y^2)**
   Berapa nilai turunannya di titik asal ?

2.3 Penutup

Fungsi aktivasi merupakan sebuah fungsi transfer yang berguna untuk menyesuaikan output yang dihasilkan oleh sistem jaringan syaraf sehingga sesuai dengan target. Dengan demikian jaringan menjadi lebih powerfull dalam mengenali pola. Berbagai fungsi aktivasi yang dapat digunakan dalam pemodelan neural network menjadikan kompleksitas yang lebih besar untuk menyelesaikan berbagai permasalahan. Apabila

I.3 PERCEPTRON

3.1 Pendahuluan

Perceptron merupakan bentuk paling sede
sistem jaringType equation here.an syaraf buata
memuat layer input dan output saja. Perceptro
dipelajari karena menjadi dasar bagi pemb
neural network yang
lebih kompleks. Pada bagian ini akan dipel
perceptron dan penerapannya pada fungsi logik
yang diaplikasikan menggunakan software Matla

3.2 Penyajian

3.2.1 Pengertian Perceptron

Suatu jaringan syaraf dengan la
dinamakan dengan perceptron. Pe
mempunyai tiga lapisan neuron yaitu un
asosiator dan unit respon, membentuk m
dari suatu retina. Salah satu percep
menggunakan aktivasi biner untuk un
asosiator serta aktivasi +1, 0 atau –1 unt
Unit-unit sensorik dihubungkan ke unit a
koneksi dengan bobot-bobot tetap yang nil
–1.

Fungsi aktivasi untuk masing-masin
merupakan fungsi step biner dengan suat
bebas tetapi tertentu. Sinyal yang dik
asosiator ke unit output merupakan sinyal
Output dari perceptron adalah $y = f(y\_in)$
aktivasinya adalah :

$$f(y\_in) = \begin{cases} 1 & \text{jika } y\_in > \theta \\ 0 & \text{jika } -\theta \leq y\_in \leq \theta \\ -1 & \text{jika } y\_in < \theta \end{cases}$$

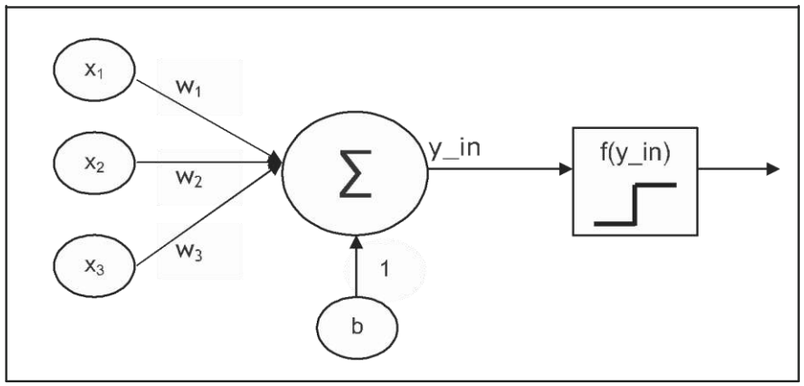
Bobot dari unit asosiator ke unit
diatur dengan hukum pembelajara
(perceptron learning rule). Untuk tiap-tia
jaringan akan menghitung respon da
Kemudian jaringan akan menentukan apa
untuk pola ini melalui pembandingan

dengan error pada output terhitung +1 dan targetnya –1. Tanda dari error menunjukkan bahwa bobot harus diubah dalam arah yang diindikasikan oleh nilai target. Akan tetapi hanya bobot pada koneksi dari unit-unit yang mengirimkan sinyal tidak nol ke unit output yang dibetulkan (hanya sinyal tersebut yang memberikan kontribusi adanya error). Jika error terjadi pada pelatihan pola input tertentu bobot akan diubah dengan rumus :

$$w_i(new) = w_i(old) + \eta t x_i$$

Dalam hal ini nilai target t adalah +1 atau –1 dan adalah tingkat pelatihan (learning rate). Jika tidak terjadi error maka bobot tidak diubah. Pelatihan akan berlangsung sampai tidak ada error. Perceptron dapat dilatih dengan prosedur least square yaitu suatu prosedur pemilihan bobot yang meminimumkan jumlah kuadrat error dimana penjumlahannya meliputi semua output dan pelatihan.

Secara sederhana perceptron dengan input x_1, x_2, dan x_3 disajikan seperti pada gambar 1.15 :



Gambar 1.15 Single layer perceptron

Pada gambar 1.15, b adalah bias dengan bobot bernilai 1 dan f(y_in) adalah output dari perceptron. Secara analogi pada statistika, seperti pada model regresi linear dimana x_1, x_2 dan x_3 merupakan variabel independen, bias identik dengan slope model sedangkan output jaringan f(y_in) identik dengan nilai prediksi dari variabel dependen y.

Algoritma pelatihan perceptron d
sebagai berikut (Kusumadewi, 2004) :

0. Inisialisasi :
   a. Bobot input variabel ke-i menuju neu
      bobot bias menuju ke neuron ke-j (b_j).
   b. Set learning rate :
   c. Set maksimum epoch (MaxEpoch)
1. Tetapkan epoch = 0
2. Selama kondisi berhenti bernilai false, la
   langkah berikut :
   a. Untuk setiap pasangan pembelajara
      k=1,2,…,n kerjakan :
      (i). Set input dengan nilai sama
      input: $x_{ki} = s_{ki}$; dengan k = 1,2,…
      (ii). Hitung respon untuk unit outpu

$$y\_in_j = b_j + \sum_{i=1}^{m} x_i w_{ij} \quad \text{dengan } j =$$

$$\begin{cases} y_j = 1, & \text{jika } y\_in_j \geq 0 \\ \quad\ 0, & \text{jika } y\_in_j < 0 \end{cases} \quad ; \text{ unt}$$

$$\begin{cases} y_j = 1, & \text{jika } y\_in_j \geq 0 \\ \quad -1, & \text{jika } y\_in_j < 0 \end{cases} \quad ; \text{ untu}$$

      (iii). Perbaiki bobot dan bias jika terja
      Jika $y_j \neq t_{kj}$ maka : $w_{ij} = w$

$$b_j = b_j +$$

      jika tidak, tidak akan terjadi perubaha
   b. Tes kondisi berhenti : jika masih te
      bobot atau jumlah kuadrat error (SS
      MaxEpoch, kondisi berhenti FALS
      sudah tidak terjadi perubahan bob
      epoch ≥ MaxEpoch, maka kondisi berhe

Untuk memberikan gambaran yang le
pemodelan perceptron, berikut ini
aplikasi jaringan perceptron menggu

1.2.2 *Membangun Fungsi Perceptron*

Software MATLAB telah menyediakan fungsi yang digunakan untuk membangun jaringan perceptron yaitu dengan perintah `newp`. Syntax penulisannya adalah :

```
net=newp(PR,N)
net=newp(PR,N,AF,LF)
```

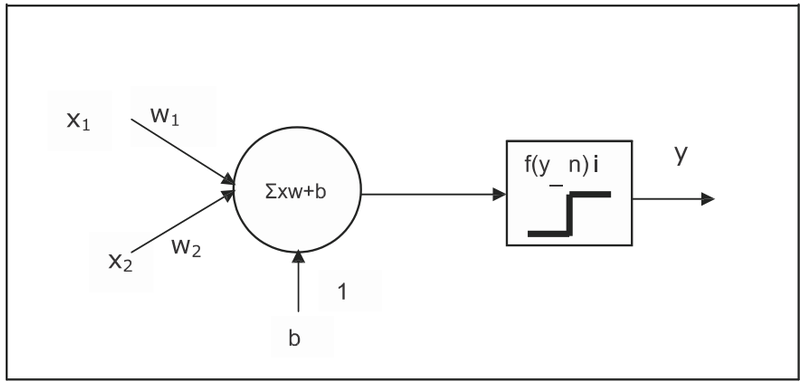
Keterangan :

PR : matrik berukuran Rx2 yang berisi nilai minimum dan maximum, dengan R adalah jumlah variabel input
N : jumlah neuron
AF : fungsi aktivasi (*actifation function,* default : `hardlim`)
LF : fungsi pembelajaran (*learning function,* default : `learnp`)

Contoh :

Akan dibangun sebuah perceptron untuk operasi logika OR dengan input dan output biner, sebagai berikut :

INPUT		TARGET
P1	P2	T
0	0	0
0	1	1
1	0	1
1	1	1



Untuk menyelesaikan permasalah t
fungsi yang dituliskan dalam m-file s
Silahkan diberi nama `perceptron_or.m`

```
%Aplikasi Perceptron pada Fungsi Logika OR
P = [1 1 0 0; 1 0 1 0];
T = [1 1 1  0];
net = newp([0 1;0  1],1);
Bobot_Input_Awal = net.IW{1,1}
Bobot_Bias_Awal = net.b{1}
net = init(net);
net.adaptParam.passes = 10;
[net,Y,e] = adapt(net,P,T);
plotpv(P,T);
plotpc(net.IW{1,1},net.b{1});
title('Plot Hasil Klasifikasi Perceptron pada Fungs
xlabel('X1')
ylabel('P2')
Bobot_Input_Akhir = net.IW{1,1}
Bobot_Bias_Akhir = net.b{1}
MeanSquareError = mse(e)
```

Penjelasan dari fungsi tersebut dapat
sebagai berikut :

```
P = [0 0 1 1; 0 1 0 1];
T = [0 1 1 1];
    net = newp([0 1;0 1],1);
```

Instruksi di atas berarti memban
dengan input P yang merupakan vekt
masing mempunyai nilai minimum 0 da
dan target T. Perceptron tersebut memp
fungsi aktivasi yang digunakan `hardli`
fungsi pembelajarannya `learnp` (defa
yang dibangun didefinisikan sebagai
informasi yang dapat diperoleh dari p
dibangun tersebut disajikan pada tabel 1.1.

Tabel 1.1 Beberapa informasi dari perceptron yang dibangun dengan fungsi `newp` pada Matlab

No	Informas	Perintah
1	Ukuran nput I	`net.inputs{1}.size`
2	Range input	t p t {1} y
3	Ukuran output	`net.outputs{1}.size`
4	Ukuran lapisan	`net.layers{1}.size`
5	Fungsi aktivasi	`net.layers{1}.transferFcn`
6	Ukuran bias	`net.biases{1}.size`
7	Ukuran bobot input	`net.inputWeights{1}.size`
8	Bobot bobot i p t	t IW{ }
9	Ukuran bobot lapisan	t LW{ }
10	Bobot bias	t b{1}

Misalkan ingin diketahui bobot awal dan bias dari jaringan, maka dituliskan perintah berikut :

```
Bobot_Input_Awal = net.IW{1,1}
Bobot_Bias_Awal = net.b{1}
```

Perintah `init` digunakan jika nilai-nilai bobot, baik bobot input maupun bobot bias ingin dikembalikan sesuai dengan inisialisasi fungsinya.

```
net = init(net)
```

Setelah jaringan syaraf dibentuk, langkah selanjutnya adalah melakukan pembelajaran agar jaringan syaraf bisa beradaptasi, digunakan perintah `adapt` seperti berikut :

```
[net,Y,e] = adapt(net,P,T)
```

Dalam hal ini `net` telah berubah menjadi jaringan syaraf yang telah beradaptasi, `Y` adalah output jaringan dan `e` adalah error yang terjadi setelah beradaptasi (target – output).
Jika diinginkan jumlah adaptasi sebanyak n kali perintahnya ;

```
net.adaptParam.passes = n;
```

dengan target dan plot yang dihasilkan bel
hasil klasifikasi yang benar untuk menyel
fungsi logika OR.
Untuk menggambar hubungan antara vekt
vektor target pada perceptron, digu
`plotpv` sedangkan untuk melihat gari
komputasi, dapat digunakan perintah `plot`

```
plotpv(P,T);
plotpc(net.IW{1,1},net.b{1})
```

Silahkan Saudara coba tuliskan fungsi ter
file, beri nama file `perceptron_or.m` ke
running. Jika nilai belum mencapai 0, lak
pada perintah `net.adaptParam.pa`
menambah nilainya sampai nilai MSE menc
Untuk melakukan simulasi terhadap in
tertentu terhadap jaringan syaraf yan
melakukan pembelajaran digunakan perin
penulisannya adalah :

```
z=sim(net,p)
```

Keterangan :

- z : output hasil simulasi
- net : jaringan syaraf yang telah dilatih
- p : input data yang akan disim jaringan syaraf

Misalkan dipunyai sebuah data ba
disimulasikan untuk input yaitu [1 0],
dituliskan perintah berikut pada Ma
Window :

```
>> z = sim(net,[0; 1])
```

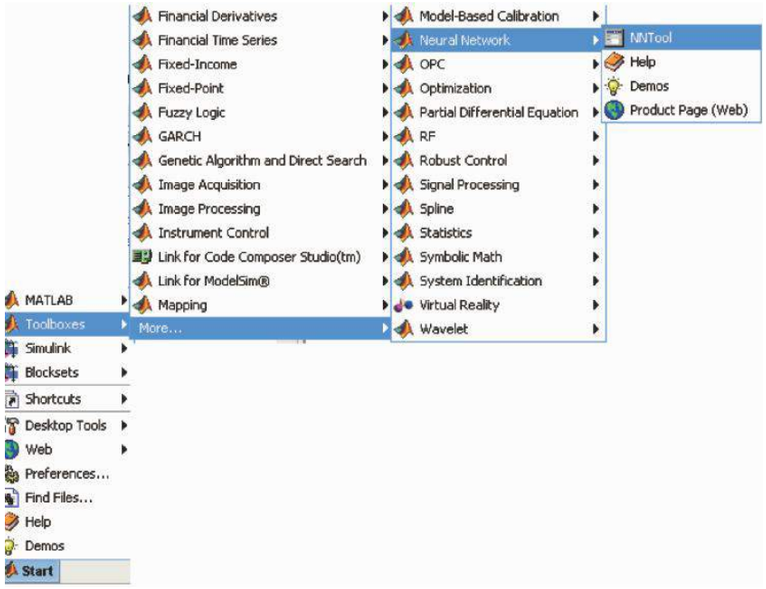
Hasilnya adalah 1 yang berarti benar, ka
pada fungsi logika akan menghasilkan n
dapat dikatakan bahwa sekarang jaring
mengenali pola dengan tepat setelah se

Silakan saudara coba membuat fungsi analog pada m-file untuk operasi AND, kemudian beri nama `perceptron_and.m`

MEMBANGUN PERCEPTRON dengan MATLAB TOOLBOX

Selain dengan menggunakan fungsi newp yang telah tersedia, Matlab juga menyediakan Toolbox untuk membangun jaringan perceptron. Urut-urutan langkah yang harus dilakukan adalah sebagai berikut :

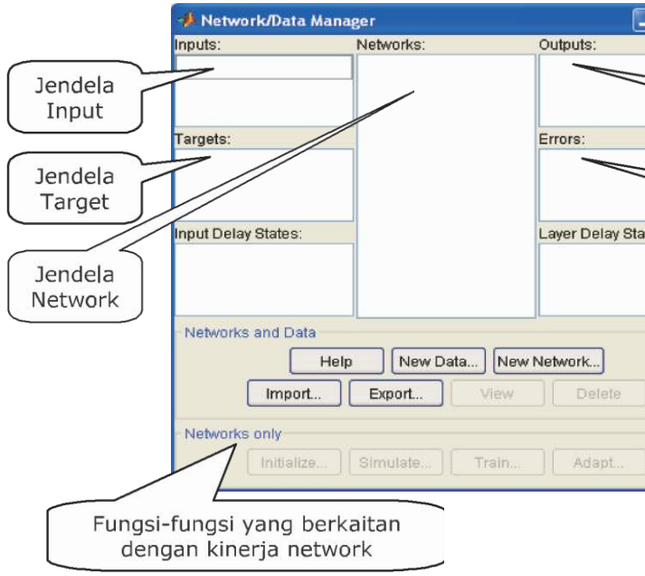
- Mengaktifkan Toolbox Neural Network (nntool)
  Dari Command Window klik Start - Toolboxes - Neural Network - NNTool seperti gambar berikut :



Atau untuk meringkas cukup tuliskan perintah `nntool` pada jendela command :

```
>> nntool
```

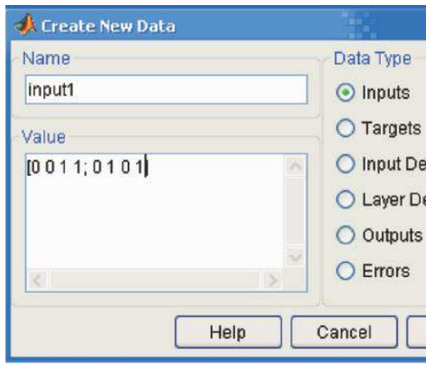
sehingga muncul dialog sebagai berikut:



Gambar Toolbox Neural Netwo

Berikut ini adalah contoh jaringan syaraf p
operasi OR dengan input dan output
langkah yang perlu dilakukan adalah sebag

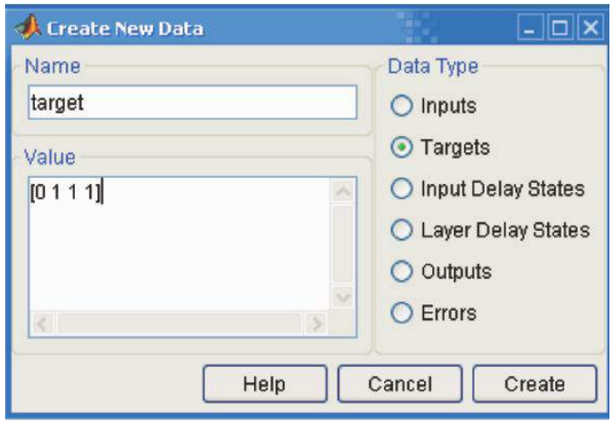
- Menginputkan data
  Dari Network/Data Manager klik jen
  toolbox kemudian klik New Data...
  muncul tampilan berikut :



Masukkan nama, nilai dan tipe data
kebutuhan. Dalam hal ini pada kota
input1, sedangkan pada kotak **Value** is
1] dan pada kotak **Data Type** klik

- Menentukan target

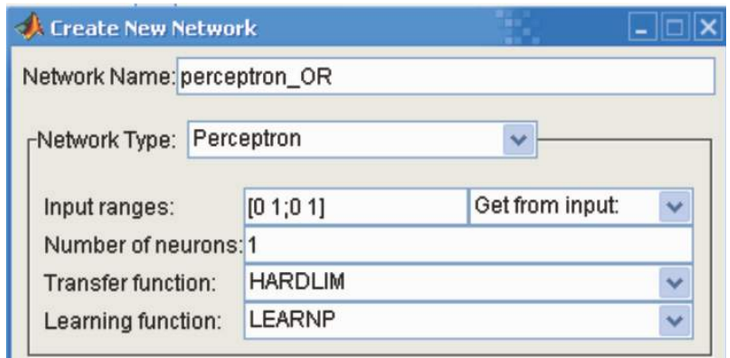
Lakukan hal yang sama untuk menuliskan target, yaitu dengan klik jendela target pada toolbox kemudian klik New Data... sehingga muncul tampilan berikut :



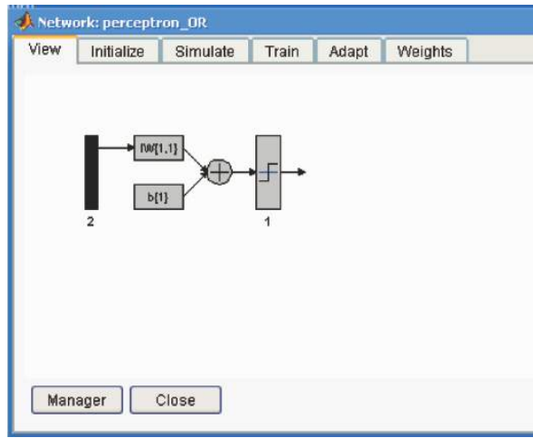
Pada kotak **Name**, isikan **target**, sedangkan pada kotak **Value** isikan [0 1 1 1] dan pada kotak **Data Type** klik pilihan **Target** kemudian klik Create .

- Membuat jaringan

Untuk membuat jaringan baru, klik jendela networks pada toolbox kemudian klik **New Network** sehingga muncul tampilan berikut. Jaringan yang dibentuk akan diberi nama perceptron_OR sehingga pada **Network Name** tuliskan perceptron_OR, pada **Network Type** pilih Perceptron, **Input Range** pilih **Get from Input**. Sedangkan **Number of Neurons** (jumlah neuron) diisi 1, fungsi aktifasi HARDLIM, dan fungsi pembelajaran LEARNP. Kemudian klik **Create**.

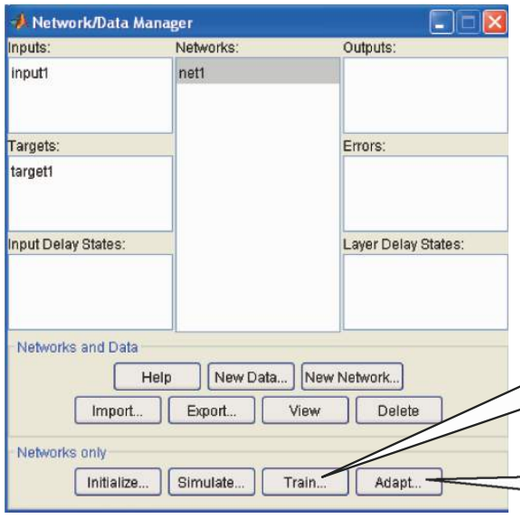


Untuk melihat gambar arsitektur perceptron yang telah dibentuk, dari pada menu **Network/Data M** **perceptron_OR** kemudian klik **View** tampilan berikut :

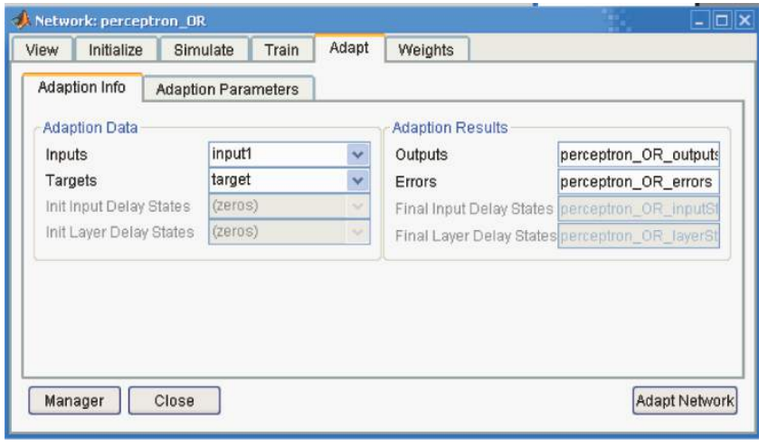


- Proses pembelajaran jaringan

Setelah jaringan syaraf dibentuk, lang adalah melakukan pembelajaran agar dapat beradaptasi. Pertama kali kl jaringan (perceptron_OR) untuk meng pada **Networks only**, kemudian pilih p dilakukan. Klik **Adapt** untuk proses ada

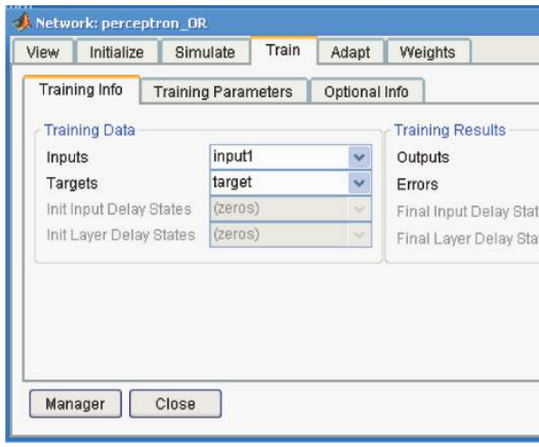


Pilih **Adaption Info** kemudian pada **Adaption Data**, isikan **input1** pada pilihan **Inputs** dan isikan target pada pilihan **Targets**.

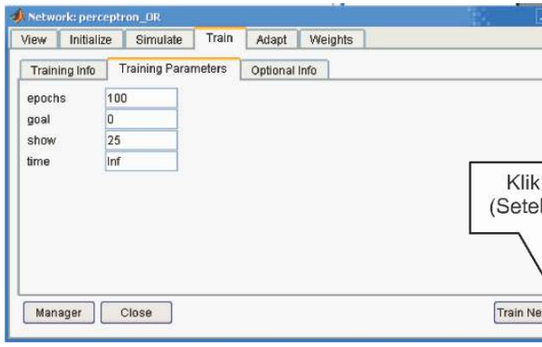


Pilih **Adaption Parameters** kemudian pada kotak passes isikan jumlah adaptasi yang diinginkan. Default untuk ini bernilai 1. Aetelah selesai mengisikan, klik **Adapt Network** untuk proses adaptasi.

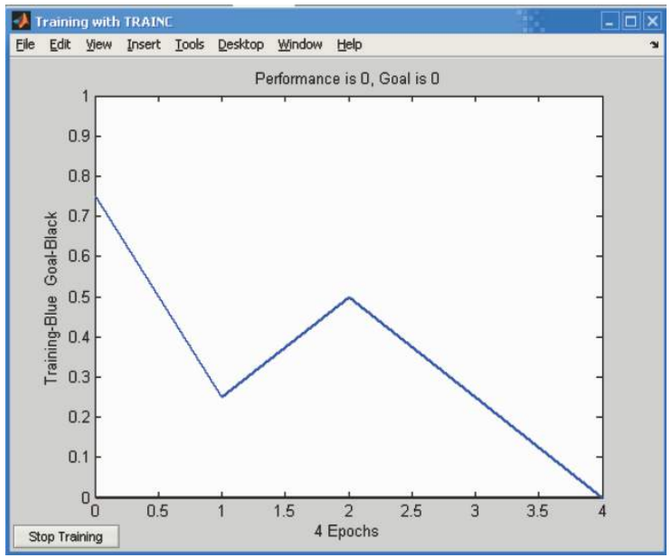
Untuk melakukan proses pelatihan jaringan, pada **Network/Data Manager** klik **Train**. Kemudian klik Training Info, selanjutnya tentukan dan tulis nama untuk data input sekaligus outputnya. Pada box **Training Data** tuliskan input1 pada pilihan **Inputs** dan isikan target pada pilihan **Targets**.



Klik **Training Parameters** sehingga m
berikut. Isikan parameter-parameter y
seperti banyaknya epoch dan nilai
(**goal**). Goal adalah unjuk kinerja yan
tuliskan 0, sedangkan **show** adalah it
ditampilkan. Pilih nilai secukupnya
iterasi yang ditampilkan tidak terlalu p
terlalu pendek, cukup untuk menunjuk
berjalan. Untuk 100 epoch, nilai show
Selanjutnya untuk **time** isikan **inf** yang
membatasi waktu iterasi (detik) yang
Klik Train Network untuk mengekseku



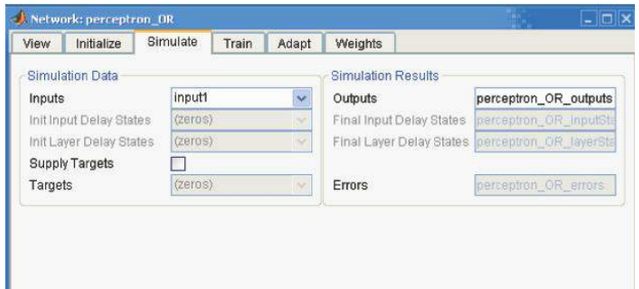
berikut ini adalah hasil proses training jaringan. Amatilah hasilnya!



Gambar di atas menunjukkan bahwa pada epoch pertama unjuk kerja bernilai 0.75, yang berarti hanya satu pola yang dikenali dengan benar. Pada epoch kedua unjuk kerja bernilai 0.25, yang berarti ada tiga pola yang dikenali dengan benar. Demikian seterusnya sehingga iterasi selesai pada epoch ke-4, dimana semua pola telah dikenali dengan benar (Performance = 0).

- Proses simulasi jaringan

Langkah selanjutnya adalah proses simulasi jaringan. Untuk melakukan simulasi input data baru tertentu terhadap jaringan syaraf yang telah selesai melakukan pembelajaran, klik pada tab **Simulate** sehingga muncul tampilan berikut.



Isikan input yang akan disimula
mengisikan **input1** pada pilihan **Input**
Simulation Data. Pada kotak **Simulat**
nama output yang akan dihasilkan
output_sim1 pada pilihan **Outputs**
Simulate Network.

Melihat nilai bobot-bobot

Untuk melihat bobot jaringan, pili
kemudian pilih nilai bobot yang ak
pilihan **Select the weight or bias to view**