

# **PREDIKSI TINGKAT KELULUSAN UJIAN NASIONAL MENGUNAKAN ALGORITMA ARTIFICIAL NEURAL NETWORK**

Derry Prima

A11.2010.05711

Universitas Dian Nuswantoro

Jl. Nakula I No. 5 – 11 Semarang 50131

email : [prima\\_derry@yahoo.com](mailto:prima_derry@yahoo.com)

## **ABSTRACT**

The value of national exam determining graduation grade XII, a level of graduation of students at the school were the size of the school quality, many students are depressed in the face of national exam, can even make grade XII into stress when it will face the national exam. The need to prepare a more mature and early so that students are calmer and ready to face the national exam. In order for the National Exam Prep it early and effectively it is necessary to know the potential-potential failure of national examination that will probably happen to a student with a graduation rate of predicting a national examination subjects based on conditions the value of local subjects.

This research aims to predict the rate of graduation of the UN in a report card based on the subjects and the value of UN related subjects, so that preventive action can be performed on the student. Process calculation of predictions will be using MATLAB software and using the Neural Network algorithm and algorithm Backpropagation.

Keywords : neural network backpropagation, prediction, passing the national exam level

## **PENDAHULUAN**

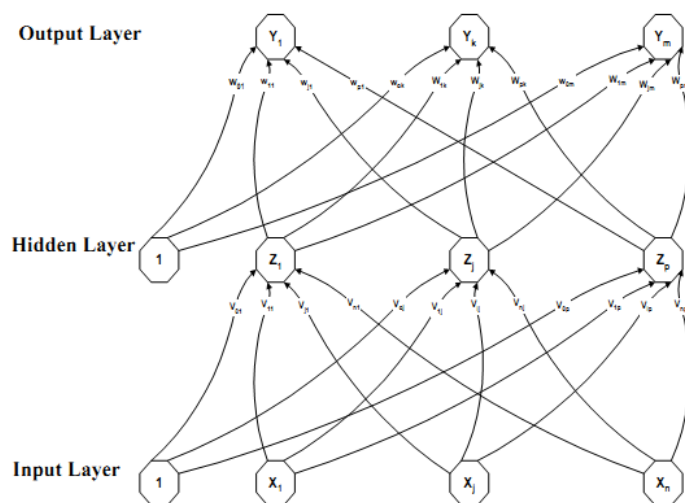
Tingkat kepentingan UN adalah untuk mengetahui kemampuan lulusan pendidikan dari berbagai jenjang dalam bidang kajian tertentu sebagai indikator keberhasilan sistem pendidikan. Nilai ujian nasional menentukan kelulusan siswa kelas XII, tingkat kelulusan siswa di sebuah sekolah dijadikan ukuran mutu sekolah tersebut. Agar persisapan ujian nasional lebih dini dan efektif maka perlu mengetahui potensi – potensi kegagalan ujian nasional yang akan mungkin terjadi pada diri seorang siswa dengan memprediksi tingkat kelulusan ujian nasiona suatu mata pelajaran berdasarkan kondisi nilai mata pelajaran lokal.

- Algoritma Backpropagation

Jaringan Syaraf Tiruan (JST) merupakan salah satu sistem pemrosesan informasi atau data yang didisain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Salah satu metode yang digunakan dalam JST adalah Backpropagation. Backpropagation adalah algoritma pembelajaran untuk memperkecil tingkat error dengan cara menyesuaikan bobotnya berdasarkan perbedaan output dan target yang diinginkan.

Backpropagation juga merupakan sebuah metode sistematis untuk pelatihan multilayer JST karena Backpropagation memiliki tiga layer dalam proses pelatihannya, yaitu input layer, hidden layer dan output layer, dimana backpropagation ini merupakan perkembangan dari single layer network (Jaringan Layer Tunggal) yang memiliki dua layer, yaitu input layer dan output layer. Dengan adanya hidden layer pada backpropagation dapat menyebabkan tingkat error pada backpropagation lebih kecil dibanding tingkat error pada single layer network. karena hidden layer pada backpropagation berfungsi sebagai tempat untuk mengupdate dan menyesuaikan bobot, sehingga didapatkan nilai bobot yang baru yang bisa diarahkan mendekati dengan target output yang diinginkan.

Arsitektur Backpropagation



Gambar 2.6 Arsitektur Backpropagation

dimana :

$X$  = Masukan (input).

$V_{ij}$  = Bobot pada lapisan tersembunyi.

$W_{ij}$  = Bobot pada lapisan keluaran.

$V_{oj}$  = Bias pada lapisan tersembunyi

$W_{oj}$  = Bias pada lapisan tersembunyi dan lapisan keluaran.

$J = 1, 2, 3, \dots, n$ .

$n$  = Jumlah unit pengolah pada lapisan tersembunyi.

$k$  = Jumlah unit pengolah pada lapisan keluaran.

$Y$  = Keluaran hasil.

Arsitektur algoritma backpropagation terdiri dari tiga layer, yaitu input layer, hidden layer dan output layer. Pada input layer tidak terjadi proses komputasi, namun terjadi pengiriman sinyal input  $X$  ke hidden layer. Pada hidden dan output layer terjadi proses komputasi terhadap bobot dan bias dan dihitung pula besarnya output dari hidden dan output layer tersebut berdasarkan fungsi aktivasi tertentu. Dalam algoritma backpropagation ini digunakan fungsi aktivasi sigmoid biner, karena output yang diharapkan bernilai antara 0 sampai 1.

Beberapa faktor yang mempengaruhi keberhasilan algoritma backpropagation, antara lain:

- Laju pembelajaran

Laju pembelajaran merupakan parameter jaringan dalam mengendalikan proses penyesuaian bobot. Nilai laju pembelajaran yang optimal bergantung pada kasus yang dihadapi. Laju pembelajaran yang terlalu kecil menyebabkan konvergensi jaringan menjadi lebih lambat, sedang laju pembelajaran yang terlalu besar dapat menyebabkan ketidakstabilan pada jaringan.

- Momentum

Momentum digunakan untuk mempercepat pelatihan jaringan. Metode momentum melibatkan penyesuaian bobot ditambah dengan faktor tertentu dari penyesuaian sebelumnya.

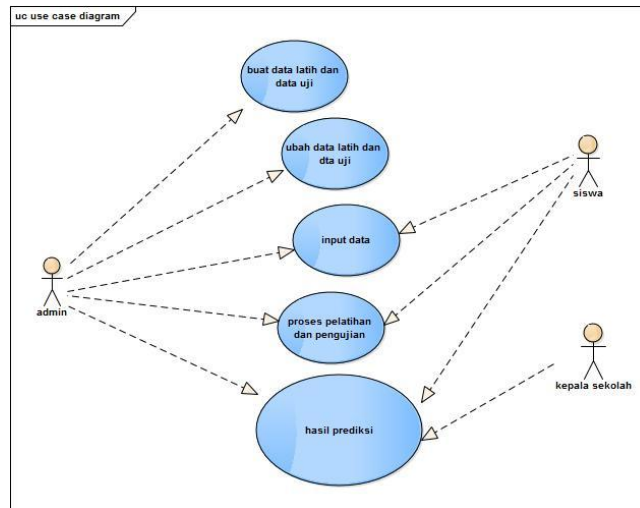
#### METODE PENELITIAN

- Data Nilai Ujian Nasional

NO	Nama	Mapel	Nilai UN
1	Aditya Nurul Susanti	B.Indonesia	8.40
		B.Inggris	6.80
		Matematika	2.00
		Fisika	8.00
		Kimia	7.40
		Biologi	8.00
2	Anggrie Januarizky	B.Indonesia	6.80
		B.Inggris	7.80
		Matematika	4.00
		Fisika	8.25
		Kimia	7.80
		Biologi	9.00
3	Bella Pitaloka Oktavia	B.Indonesia	8.20
		B.Inggris	6.40
		Matematika	5.75
		Fisika	7.50
		Kimia	7.40
		Biologi	8.20

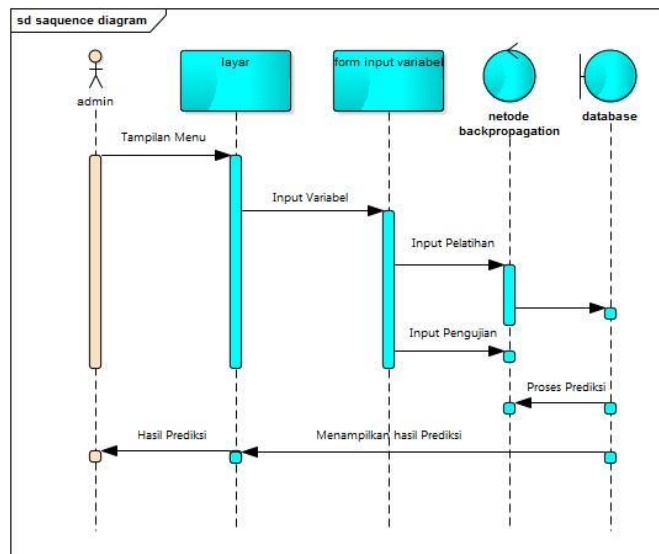
- Usecase Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.



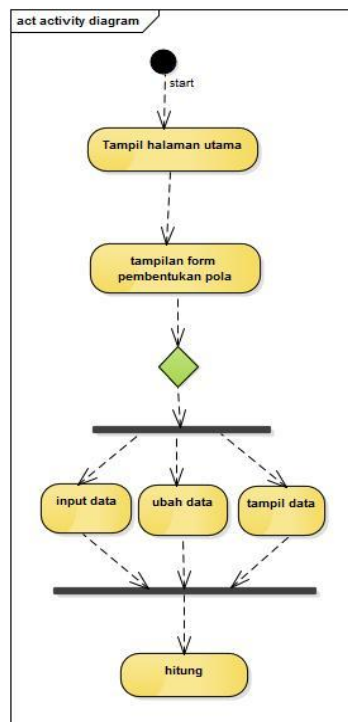
- Sequence Diagram

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa saja yang terjadi secara internal dan output apa yang dihasilkan



- Activity Diagram

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak atau tepat, tetapi lebih menggambarkan proses – proses dan jalur – jalur aktivitas dari level atas secara umum.

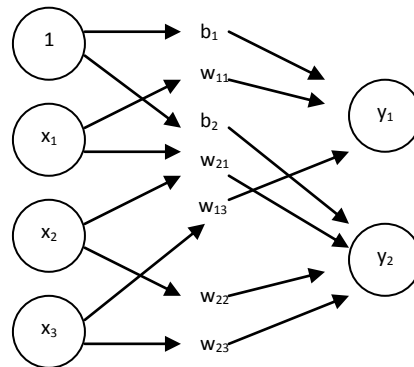


## RANCANGAN SISTEM DAN IMPLEMENTASI

MATLAB menyediakan toolbox untuk menyelesaikan masalah jaringan syaraf tiruan (JST). Spesifikasi perceptron jaringan pada MATLAB adalah sebagai berikut:

- Masukan point dan target berbentuk bebas
- Treshold yang dipakai adalah nol
- Fungsi aktivasi (*hardlim*) memiliki output biner
- Bobot diubah berdasarkan error yang terbentuk dari selisih antara target yang diinginkan dan keluaran jaringan.

### ***Pembentukan Jaringan***



Misalkan bobot jaringan ke target y<sub>1</sub> adalah [-1 0 2], bobot ke target y<sub>2</sub> adalah [0 1 0] dan bobot bias adalah [1 2]. Statement yang digunakan untuk membangun jaringan adalah

```
net.IW{1,1}=[-1 0 2;0 1 0]
net.b{1}=[1;2]
```

Untuk membentuk backpropagation untuk mengenali fungsi logika "AND" dengan 2 variabel x<sub>1</sub> dan x<sub>2</sub>, digunakan statement `net=newp(PR,S,TF,LF)` dengan  
PR=matriks ordo R x 2 menyatakan nilai maksimum dan minimum tiap unit masukan  
S=jumlah neuron dari target  
TF=fungsi aktivasi biner  
LF=fungsi pelatihan. (default: learnp)  
Sehingga:

```
%lanjutkan
net=newp([0 1 ; 0 1], 1)
```

dengan [0 1] menyatakan variabel x<sub>1</sub> dan kolom yang kedua merupakan variabel x<sub>2</sub>

### ***Pembentukan Vektor Masukan dan Target***

Setelah membentuk backpropagation, berikutnya didefinisikan pola masukan dan target yang akan dikenali. Masing-masing masukan dan keluaran berupa vektor kolom. Perhatikan bahwa jumlah unit tiap pola masukan serta range nilainya harus sesuai dengan spesifikasi perceptron yang dibuat.

Pembentukan backpropagation dengan masukan dan targetnya untuk mengenali pola fungsi logika "AND" dengan bobot awal  $w = [-1 \ 1]$  dan bias=[1] dilakukan prosedur di atas dan berikut:

```
%Mulai baru
net=newp([0 1;0 1],1);
```

```
net.IW{1,1}=[-1 1]
net.b{1}=[1]
p = [ [1 ; 1] [1;0] [0;1] [0;0]];
t = [1 0 0 0];
```

### ***Menghitung Keluaran Backpropagation***

Setelah pola masukan diberikan, kita dapat menghitung keluaran jaringan. Perintah `sim` dalam MATLAB dipakai untuk menghitung keluaran Backpropagation. Formatnya adalah: `[Y,Pf,Af,E,perf]=sim(net,P,Pi,Ai,T)`, dengan parameter masukan

`Net` : nama jaringan dalam jaringan `newp`

`P` : vektor masukan jaringan

`Pi` : Kondisi delay masukan (default : nol)

`Ai` : Kondisi delay layar (default : nol)

`T` : vektor target jaringan (default : nol)

Dan parameter hasil

`Y` : keluaran jaringan

`Pf` : Kondisi akhir delay masukan

`Af` : Kondisi akhir delay layar

`E` : Error jaringan ( $T - Y$ )

`Perf.` : Unjuk kerja jaringan (performance)

Untuk menghitung keluaran jaringan, dapat digunakan statemen sederhana: `y=sim(net,p)`.

Misalkan ingin menghitung keluaran Backpropagation contoh terakhir:

```
%lanjutkan
[a,Pf,Af,e,perf]=sim(net,p,[ ],[ ],t)
```

didapatkan hasil yang menunjukkan performansinya adalah 0,75 yang berarti hanya pola pertama saja yang dikenali dengan benar.

### ***Modifikasi bobot dan bias***

Setelah menghitung keluaran jaringan, langkah berikutnya adalah mengubah bobot berdasarkan selisih antara keluaran jaringan dan target yang diinginkan. Untuk mengubah bobot digunakan perintah `learnp` yang formatnya:

`dW=learnp(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)` (ketik `help learnp` untuk keterangan lebih lanjut)

Misalkan perhitungan bobot setelah perintah `sim` pada contoh di atas dijalankan berturut-turut pada pola `[1;1]`, `[1;0]`, `[0;1]`, `[0;0]`.



```

%mulai baru

p = [ [1 ; 1] [1;0] [0;1] [0;0]];
t = [1 0 0 0];
net=newp([0 1;0 1],1);
bobot=[-1,1];
net.IW{1,1}=bobot;
bias=1;
net.b{1}=bias;
p1=p(:,1); p2=p(:,2); p3=p(:,3); p4=p(:,4);

t1=t(1); t2=t(2); t3=t(3); t4=t(4);

%perubahan bobot untuk pola p1

disp('pola p1 :')
a1=sim(net,p1)
e1=t1-a1
dW=learnp(bobot,p1,[ ],[ ],[ ],[ ],e1,[ ],[ ],[ ],[ ],[ ])
bobot=bobot+dW

```

Perintah learnp akan menyimpan besarnya perubahan bobot pada variabel dW. Bobot baru diperoleh dengan menambahkan dW ke vektor bobot. Karena keluaran jaringan = target, maka tidak dilakukan perubahan bobot sehingga bobot baru = bobot lama.

Perubahan bobot untuk pola p2, p3 dan p4 dilakukan dengan program serupa

```

%lanjutkan

disp('pola p2 :')
a2=sim(net,p2)
e2=t2-a2
dW=learnp(bobot,p2,[ ],[ ],[ ],[ ],e2,[ ],[ ],[ ],[ ],[ ])
bobot=bobot+dW

disp('pola p3 :')
a3=sim(net,p3)
e3=t3-a3
dW=learnp(bobot,p3,[ ],[ ],[ ],[ ],e3,[ ],[ ],[ ],[ ],[ ])
bobot=bobot+dW

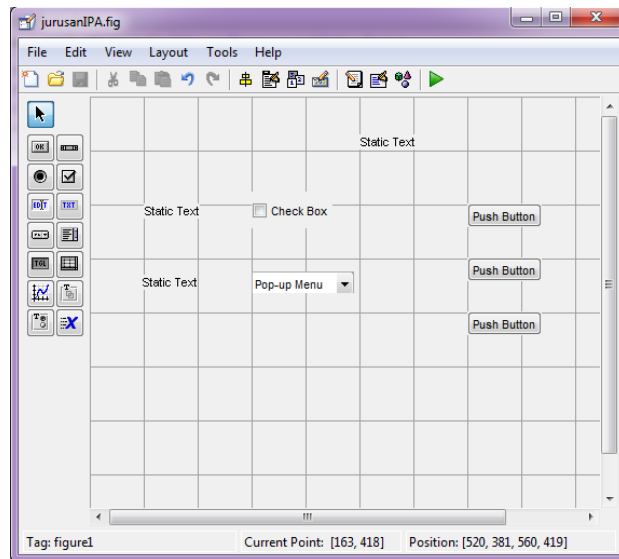
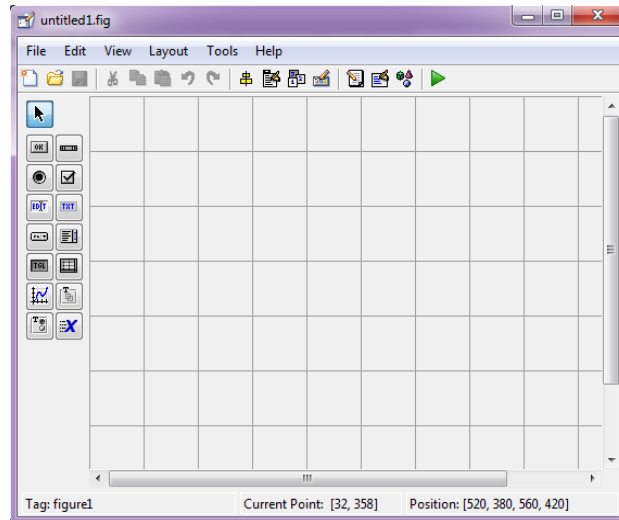
disp('pola p4 :')
a4=sim(net,p4)
e4=t4-a4

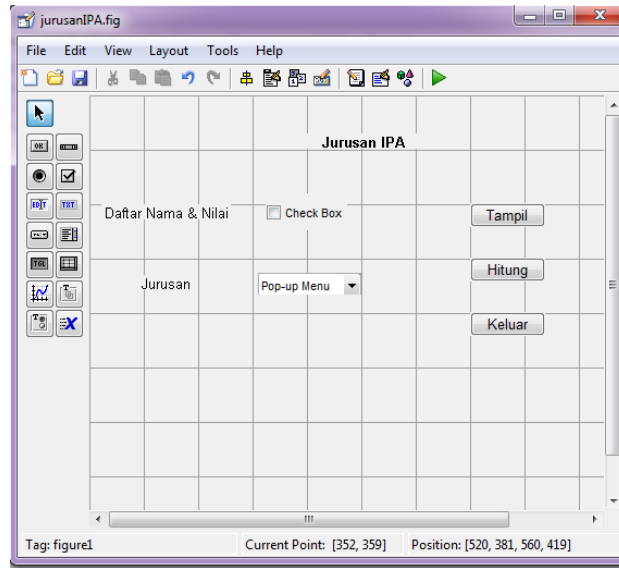
```

```
dW=learnp(bobot,p4,[ ],[ ],[ ],[ ],e4,[ ],[ ],[ ],[ ],[ ])
bobot=bobot+dW
```

Pada pola 2 dan 3,  $e = t - a = -1$  Sehingga  $dW = -p$ . Perubahan ini ditambahkan ke vektor bobot. Pada pola 4,  $e = 0$  sehingga tidak dilakukan perubahan bobot.

Hasil Rancangan GUI





## KESIMPULAN DAN SARAN

### Kesimpulan

1. Sistem prediksi tingkat kelulusan ujian nasional dapat menghasilkan informasi yang dibutuhkan oleh pihak sekolah secara cepat, relevan, akurat, lengkap dan tepat.
2. Sistem prediksi ini merupakan suatu sistem berbasis komputer yang menitikberatkan pada perencanaan sistem yang dikhususkan bagi sekolah sehingga dapat memberi semua kebutuhan informasi untuk mendukung tingkat kelulusan ujian nasional pada SMA Mardisiswa.
3. Sistem baru yang dirancang ini diharapkan dapat membantu pihak sekolah untuk lebih memperoleh informasi tentang tingkat kelulusan ujian nasional yang akan berlangsung.

### Saran

Adapun saran yang diberikan penulis dalam penelitian ini adalah :

1. Sistem prediksi ini hanya menggunakan beberapa sampel data nilai siswa yang ada, untuk pengembangan selanjutnya sebaiknya dapat melakukan inputan data prediksi sesuai dengan spesifikasi perubahan yang ada.

2. Untuk peningkatan pengembangan selanjutnya sebaiknya dapat menambahkan data uji coba dengan variabel yang lebih banyak dan lebih spesifik dengan variasi inputan yang lebih beragam agar dapat lebih meningkatkan keakuratan kinerja output dari metode yang digunakan.

3. Untuk pengembangan kinerja penggunaan metode Artificial Neural Network ini kiranya dapat melakukan pengujian dengan kasus yang berbeda dengan menggunakan data dan objek yang berbeda atau dengan menggunakan metode lain agar untuk masalah prediksi tingkat kelulusan ujian nasional ini dapat menghasilkan prosentase tingkat kelulusan yang lebih baik.

## DAFTAR PUSTAKA

[1] Davies, and Paul Beynon, "Database Systems Third Edition", Palgrave Macmillan, New York, 2004.

[2] Pramudiono, I. Pengantar Data Mining : Menambang Permata Pengetahuan di Gunung Data, 2007.

[3] Santosa, Budi, "Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis", Graha Ilmu, Yogyakarta, 2007.

[4] Han, J. and Kamber, M, "Data Mining Concepts and Techniques Second Edition". Morgan Kaufman, San Francisco, 2006.

[5] Pressman, Roger S, "Software Engineering: A Practitioner's Approach." The McGraw-Hill Companies, Inc., New York, 1997.

[6] Pärt-Enander, E. dan Sjöberg A., The Matlab 5 Handbook, Addison-Wesley, 1999.

[7] Situs <http://oktoferiana.blogspot.com/2013/10/dampak-ujian-nasional-terhadap.html> diakses pada 26 april 2014.

[8] Ashish, D. Land-use classification of aerial images using artificial neural network, M.S. Thesis, Artificial Intelligence, University of Georgia, Athens, U.S.A, 2002.

[9] Badrul Anwar, "Penerapan Algoritma Jaringan syaraf tiruan backpropagation dalam memprediksi tingkat suku bunga bank, 2011".

[10] Zurada, J.M, Introduction To Artificial Neural Systems, Boston: PWS Publishing Company, 1992.

[11] Haykin, S, Neural Networks: A Comprehensive Foundation, NY, Macmillan, 1994.

[12] DARPA Neural Network Study, AFCEA International Press, halaman 60, 1988.

[13] <http://sharetechnolive.blogspot.com/2013/05/pengertian-estimasi.html> diakses pada 27 april 2014.

[14] <http://ikkholis27.wordpress.com/2011/07/07/feed-forward-dan-backpropagation/> diakses pada 28 april 2014.