



## **Jaringan Syaraf Tiruan dengan Teknik Backpropagation untuk Prediksi Standar Kelulusan Ujian Nasional Produktif Kompetensi di SMK (Studi Kasus di SMK Nasional Al Huda Pekanbaru)**

Rometdo Muzawi  
STMIK-Amik Riau

*rometdomuzawi@stmik-amik-riau.ac.id*

Nori Sahrin  
STIE RIAU

*norisahrin84@gmail.com*

### **Abstrak**

*Hasil Ujian Nasional Produktif Kompetensi sangat mempengaruhi total nilai Ujian Nasional. Penurunan nilai teori produktif berdampak signifikan atas jumlah kumulatif nilai ujian nasional. Salah satu teknik untuk memprediksi Standar Ujian Nasional Produktif Kompetensi kelulusan adalah dengan metode jaringan saraf tiruan yaitu metode Backpropagation. Backpropagation adalah sebuah metode sistematis untuk pelatihan multilayer jaringan saraf tiruan. Jaringan Backpropagation merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah- masalah yang rumit Metode ini menggunakan prinsip kerja otak manusia yang terdiri dari neuron sebagai pemrosesan input untuk menghasilkan output berdasarkan bobot yang ada. Dengan metode Backpropagation ini dapat di peroleh tingkat RMSE (Root Mean Square Error) yang digunakan sebagai dasar pengayaan pada Smk Nasional Al Huda Pekanbaru.*

*Kata Kunci : Jaringan Saraf Tiruan, Root Mean Square Error, Backpropagation*

### **1. Pendahuluan**

#### **1.1. Latar Belakang Masalah**

Hasil Ujian Nasional Produktif Kompetensi merupakan standarisasi dari pemerintah yang akan menunjukkan kemampuan siswa pada mata pelajaran

yang diujikan dan juga akan menentukan siswa dalam kenaikan tingkat dari SMK ke jenjang yang lebih tinggi yaitu ke bangku kuliah. Dalam proses belajar siswa akan memperoleh nilai, baik nilai harian, nilai ujian semester hingga ujian nasional produktif kompetensi yang diadakan tiap tahun untuk menentukan kelulusan siswa. Nilai tersebut akan disimpan dalam arsip baik secara komputerisasi dan manual. Pihak sekolah ingin mengetahui standar kelulusan kompetensi siswa dalam menghadapi ujian, maka diperlukan metode untuk mengolah data nilai siswa yang lama untuk memprediksi nilai siswa yang akan mengikuti ujian nasional yaitu dengan menggunakan metode Jaringan Syaraf Tiruan dengan Backpropagation.

Jaringan saraf Tiruan adalah 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 saraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. (Mistianingsih, Barong, Unmul, Kelua, & Samarinda, 2010)

Salah satu cabang dari AI (*Artificial Intelligence*) adalah apa yang dikenal dengan Jaringan Saraf Tiruan (*Artificial Neural Network*). Kecerdasan buatan (*Artificial Intelligence*) adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. (Komputer et al., 2012)

Perambatan galat mundur (*Backpropagation*) adalah sebuah metode sistematis untuk pelatihan multilayer jaringan saraf tiruan. Jaringan *Backpropagation* merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Algoritma ini memiliki dasar matematis yang kuat dan dilatih dengan menggunakan metode belajar terbimbing. Pada jaringan diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan kepada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan. Latihan ini dilakukan berulang-ulang sehingga semua pola yang dikeluarkan jaringan dapat memenuhi pola yang diinginkan. (Anwar, 2011)

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.

Jaringan Syaraf Tiruan dengan metode *Backpropagation* cocok untuk peramalan dalam mengukur tingkat korelasi prestasi siswa/siswi. Dengan menggunakan metode *Backpropagation* yang mengubah bobot dengan alur mundur dari lapisan keluaran ke lapisan masukan untuk mendapatkan keseimbangan kemampuan jaringan dalam menentukan pola yang digunakan selama masa pelatihan diharapkan dapat mengoptimalkan hasil yang diinginkan. Algoritma *Backpropagation* untuk memprediksi kinerja akademik pelajar. (Samy S. Abu Naser, 2012).

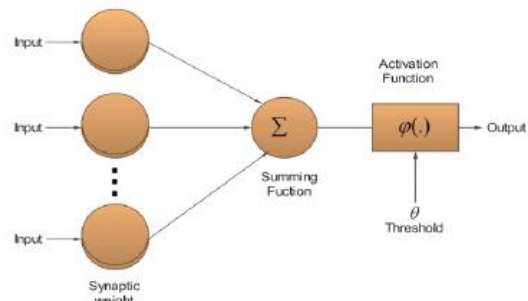
Penelitian ini bertujuan untuk menerapkan Jaringan Syaraf Tiruan dengan metode *Backpropagation* untuk memprediksi nilai Ujian Nasional produktif kompetensi di SMK Nasional Al Huda Pekanbaru, dan dapat digunakan sebagai acuan pengayaan pada mata pelajaran Teori Produktif.

## 2. Landasan Teori

### 2.1. Artificial Neural Network (ANN)

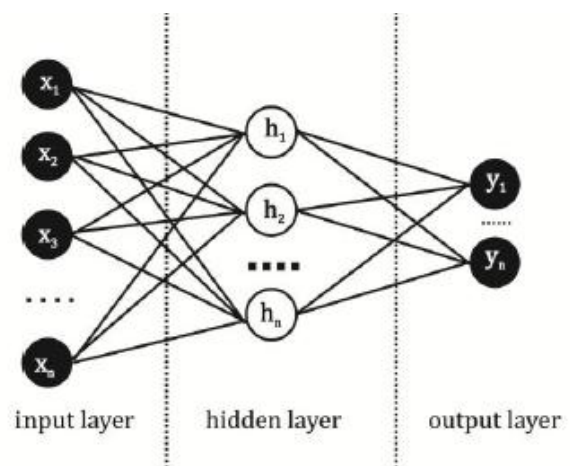
*Artificial Neural Network* (ANN) merupakan model penalaran yang didasarkan pada otak manusia. ANN terdiri dari sejumlah prosesor sangat sederhana dan saling berhubungan yang disebut neuron. Neuron yang terhubung dengan pembobotan (*weight*) melewati sinyal dari neuron satu ke neuron yang lain. (Raharjo, 2013)

ANN mampu mengenali sinyal input yang agak berbeda dari yang pernah diterima sebelumnya dan mampu bekerja meskipun beberapa neuronya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak maka neuron lain dapat dilatih untuk menggantikan fungsi neuron yang rusak tersebut. Model neuron ANN terdiri; fungsi penjumlahan (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).



Gambar 1. Model neuron

Cara kerja ANN adalah informasi yang masuk (input) akan dikirim ke neuron dengan bobot tertentu dan selanjutnya diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan akan dibandingkan dengan nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Jika input melewati suatu nilai ambang tertentu, maka neuron akan diaktifkan, jika tidak, maka neuron tidak akan diaktifkan. Neuron yang diaktifkan akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Struktur ANN terdiri dari layer input, hidden layer, dan output layer seperti gambar dibawah ini:



Gambar 2. Struktur ANN

Jaringan Saraf Tiruan (JST) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut. Untuk JST tercipta sebagai suatu generalisasi model matematika dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron. Isyarat mengalir diantara sel saraf melalui suatu sambungan penghubung, setiap sambungan penghubung memiliki bobot yang bersesuaian dan setiap sel saraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya. (Lesnussa, Latuconsina, & Persulesy, 2015)

### 2.2 Konsep Dasar Jaringan Syaraf Tiruan

Setiap informasi *input* dan *output* yang diberikan kedalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut *neuron layers*.

Menurut (Lesnussa et al., 2015) Lapisan – lapisan jaringan syaraf tiruan dibagi menjadi 3 bagian yaitu :

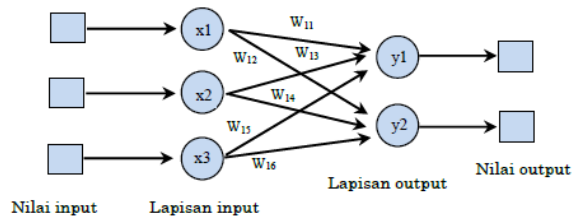
1. Lapisan *input*, unit-unit di dalam lapisan *input* disebut unit-unit *input*. Unit-unit *input* tersebut menerima pola data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan tersembunyi, unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Di mana *output*nya tidak dapat secara langsung diamati.
3. Lapisan *Output*, unit-unit di dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

### 2.3 Arsitek Jaringan Syaraf Tiruan

Menurut (Lesnussa et al., 2015) Jaringan Syaraf Tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur Jaringan Syaraf Tiruan yaitu :

#### 1. Jaringan Lapisan Tunggal (*Single Layer Network*)

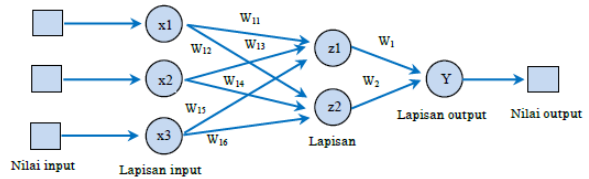
Jaringan dengan lapisan tunggal terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Setiap neuron yang terdapat di dalam lapisan *input* selalu terhubung dengan setiap neuron yang terdapat pada lapisan *output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.



Gambar 3. Lapisan tunggal

#### 2. Jaringan Banyak Lapisan (*Multilayer Net*)

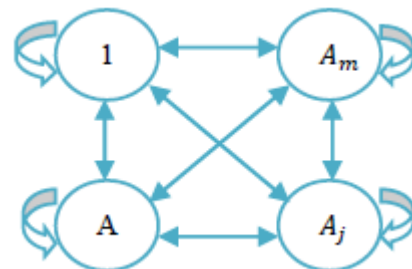
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis lapisan yakni lapisan *input*, lapisan *output*, dan lapisan tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama.



Gambar 4. Lapisan banyak

#### 3. Jaringan Lapisan Kompetitif (*Competitive Layer*)

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan jaringan ini adalah LVQ.



Gambar 5. Lapisan kompetitif

## 3. Metodologi Penelitian

### 3.1. Pendahuluan

Dalam melakukan penelitian agar hasilnya bisa maksimal, tentunya harus mengikuti kaidah-kaidah atau metode-metode yang akan digunakan .

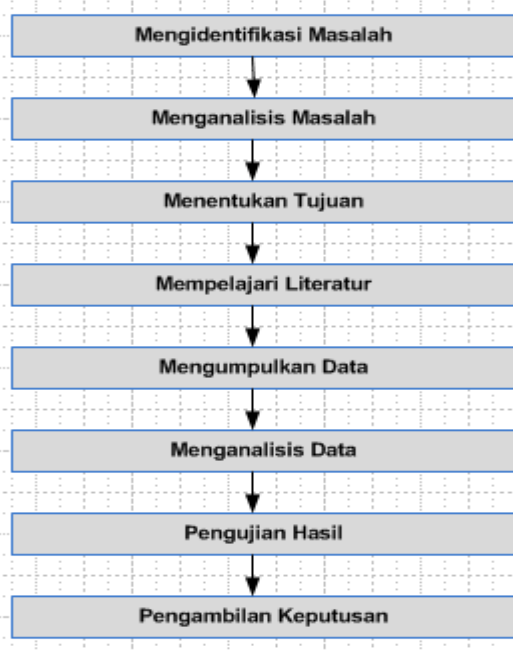
Metodologi penelitian ini memuat tentang kerangka kerja penelitian yang akan dibahas di bawah ini. Pada point ini akan dibahas metodologi penelitian

untuk mengidentifikasi permasalahan, analisa permasalahan dan pada akhirnya mencari penyelesaian masalah di dalam memprediksi kelulusan ujian nasional produktif kompetensi di SMK Nasional Al Huda Pekanbaru.

Uraian kerja akan dijelaskan di dalam kerangka penelitian yang nantinya akan menjelaskan prosedur maupun langkah-langkah yang akan dihadapi dalam memprediksi kelulusan ujian nasional produktif kompetensi di SMK Nasional Al Huda Pekanbaru.

### 3.2. Kerangka Kerja

Dalam metodologi penelitian ada urutan kerangka kerja yang harus diikuti, urutan kerangka kerja ini merupakan gambaran dari langkah-langkah yang harus dilalui agar penelitian ini bisa berjalan dengan baik. Kerangka kerja yang harus diikuti bisa dilihat pada gambar gambar 6.



Gambar 6. Kerangka kerja

## 4. Analisa Dan Perancangan

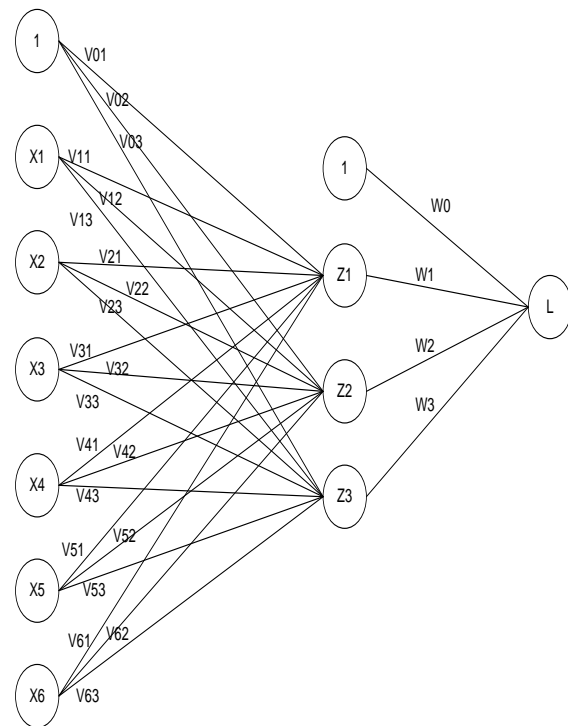
### 4.1 Gambaran Jaringan Syaraf Tiruan

Dalam membuat gambaran Jaringan Syaraf Tiruan yang harus dilakukan adalah memahami dan menentukan jumlah dari masing-masing variabel yang digunakan, diantaranya *Input layer*, *Bobot*, *Bias*, *hidden*

*layer* serta *Output layer*, hal ini berguna untuk membuat perhitungan manualnya. Pemakaian gambaran jaringan dengan berbagai macam pola akan berpengaruh terhadap kinerja jaringan dalam proses penganalisaan data dan juga untuk perengkingan jaringan untuk mengetahui pola terbaik.

Dalam penelitian ini penulis membuat gambaran Jaringan Syaraf Tiruan dengan menerapkan (*multilayer net*) dengan algoritma *Backpropagation*, gambaran tersebut berupa :

- Jumlah lapisan *input* terdiri dari 6 yaitu (X1(rekap nilai semester I ) , X2(rekap nilai semester II ) , X3(rekap nilai semester III ) , X4(rekap nilai semester IV ) , X5(rekap nilai semester V ) , X6(rekap nilai pra kompetensi )).
- Jumlah lapisan *hidden* dengan jumlah simpul yang ditentukan oleh pengguna (Z1,Z2,Z3).
- Lapisan *output* dengan 1 simpul ( L ).



Gambar 7. Gambaran jaringan syaraf tiruan penentuan ujian nasional produktif kompetensi

Keterangan :

X1 = *input layer*, *input* yang terdiri dari jumlah penentuan ujian nasional produktif kompetensi.

Z1 = *hidden layer* yang digunakan 1 lapisan yang terdiri dari 3 *neuron*.

L = lapisan *output layer* dari hasil proses analisa (dimana *output* yang akan memiliki nilai dalam range [0,1]. Apabila mendekati 0 berarti menunjukkan hasil

yang kurang baik, begitu sebaliknya jika mendekati 1 berarti menunjukkan hasil yang baik).

## 4.2 Perancangan Sistem

### 4.2.1 Perancangan Data Input-Output

Proses dalam analisa data dan sistem ini berguna untuk menentukan variabel yang menjadi patokan nantinya sebagai variabel *input*. Variabel *input* akan digunakan untuk masukan dalam mengestimasi penentuan ujian nasional produktif kompetensi pada di smk nasional al huda pekanbaru. Adapun data yang penulis gunakan berawal dari data rekap nilai kompetensi kelas xii, yang dilakukan proses penganalisaan sehingga menghasilkan data dari tahun 2014 sampai dengan tahun 2015.

**Tabel 4.2.1 Rekap data nilai smk nasional al huda pekanbaru**

NO	NAMA SISWA	S I	S II	S III	S IV	S V	PRA KOM PTS	UN KOMPTS
1	AIDIL FAJAR AKBAR	80.0	83.0	82.0	79.0	80.0	100.0	91
2	ALFA EDISON NAHAK	70.0	75.0	77.0	80.0	79.0	98.0	97
3	ANDRIAN PRANATA	73.0	76.0	75.0	75.0	76.0	85.5	93
4	AZZAER MULAHAPAN PUTRA	75.0	75.0	75.0	75.0	76.0	90.0	80
5	EBHEN HAEZER SITOANG	78.0	78.0	75.0	79.0	76.0	97.0	86
6	ERAL VALY	81.0	83.0	77.0	83.0	81.0	100.0	86
7	FEBRI RAMADHAN SAPUTRA	76.0	76.0	75.0	79.0	80.0	98.0	84
8	RIO SAFUTRA NASUTION	79.0	83.0	75.0	80.0	80.0	98.0	84
9	SANDI SUPERI	80.0	79.0	75.0	80.0	80.0	100.0	86
10	SUPRIYANTO	81.0	82.0	75.0	80.0	80.0	100.0	91
11	WISNU AGUNG WIDODO	72.0	85.0	75.0	79.0	79.0	99.5	86.0
12	DEDE ERNA SUSILAWATI	82.0	82.0	86.0	83.0	80.0	87.0	78.5
13	DELLA CHAIRUNISSA LOIPANI	91.0	89.0	88.0	88.0	85.0	94.4	82.2
14	FITRI HANDAYANI	90.0	90.0	91.0	95.0	94.0	99.4	86.0
15	INTANIA INDRIANI	80.0	82.0	83.0	85.0	80.0	91.1	80.6
16	IRMA LISA	80.0	82.0	82.0	80.0	82.0	89.8	79.9
17	KIKI SAPUTRA	81.0	86.0	85.0	83.0	81.0	97.8	86.4
18	MIA WILLIANI	87.0	83.0	88.0	90.0	87.0	91.8	83.4
19	MIRNA AGUSTINA	84.0	86.0	86.0	78.0	82.0	89.4	81.0
20	MULYANA PUTRI	78.0	78.0	76.0	78.0	80.0	90.7	81.6
21	MUTIA UTARI	82.0	80.0	83.0	78.0	80.0	89.2	79.6
22	NUR MAERUSDAHNI NGSIH	78.0	81.0	83.0	78.0	81.0	90.5	82.8

23	PUTRI HADAINA	78.0	79.0	80.0	83.0	79.0	86.0	81.8
24	RAHIL NOVRI ISMI	70.0	72.0	81.0	78.0	80.0	89.7	83.6
25	RENI ANGGRIANI	77.0	81.0	84.0	83.0	80.0	87.1	82.3
26	SONNIA CHERNOVITA ARIANA	92.0	90.0	89.0	85.0	93.0	94.8	87.4
27	ZURYANI	93.0	89.0	89.0	93.0	93.0	96.3	85.7
28	AKMAL SYARIF	9.0	9.0	8.7	8.70	8.5	89.9	89.9
29	ANDRE MANAHAN SIREGAR	8.5	8.6	8.6	8.50	8.5	89.0	89.0
30	ANDRY SILALAH	8.8	8.8	8.6	8.90	8.8	89.5	89.5
31	DEDI FIRDAUS PANDIANGAN	9.0	9.0	8.8	9.00	8.6	86.3	86.3
32	DENNY PRATAMA PUTRA	8.8	8.6	8.6	8.60	8.6	89.0	89.0
33	EBIN SIMAMORA	8.6	8.6	8.6	8.70	8.9	85.7	85.7
34	EDY SURIFTO HARIANJA	9.5	8.8	8.7	8.90	8.7	89.5	89.5
35	ELIOLA ZEPIANDA	8.5	8.6	8.6	8.80	8.9	89.0	89.0
36	FAJAR JULIANTO	8.8	8.9	8.8	8.60	8.5	88.0	88.0
37	MUHAMMAD ILHAM KARIM	8.5	8.5	8.9	8.60	8.9	87.0	87.0
38	MUHAMMAD RIZKY POETRA PRATAMA	9.0	9.0	9.0	8.70	8.6	88.8	88.8
39	NANDO RUMAHORBO	8.5	8.6	8.8	8.60	8.5	87.5	87.5
40	PUTRA DESWANTO	9.0	8.8	8.8	8.90	8.9	89.0	89.0
41	QUDDI RAMADHAN NASUTION	9.0	8.7	8.6	8.60	9.0	89.0	89.0
42	RIAN SAHADAT TAMBA	8.5	8.7	8.7	8.50	8.7	89.0	89.0
43	ROMI ARIFIN	8.9	8.7	8.9	8.50	8.8	89.5	89.5
44	RONALDO FRANSISCUS NAINGGOLAN	8.5	8.6	8.6	8.50	8.9	87.0	87.0
45	RUSDI ARDIANTO	9.0	8.9	8.8	8.60	8.5	89.0	89.0
46	SEPTIAN ABDI PRATAMA	9.3	9.0	8.6	8.70	8.5	89.5	89.5
47	SIMSON GANDA TUA PAKPAHAN	8.8	8.6	8.8	8.70	8.6	89.0	89.0
48	SUFRIYONO SILITONGA	8.6	8.7	8.6	8.60	8.6	89.0	89.0
49	ZAENAL FANANI	9.0	9.0	8.7	8.80	9.0	89.5	89.5
50	ROLANDO SIRINGO RINGO	9.0	8.8	9.0	8.50	8.6	98.9	98.9
51	WAN RIAN TO JUNI	8.5	8.5	8.7	8.60	8.6	98.0	98.0
52	AGNES AMANDA SITANGGANG	8.8	8.8	8.8	8.60	8.8	89.7	89.7
53	DANI RIALDI PASARIBU	8.6	8.7	8.6	8.90	8.9	89.9	89.9
54	DESVI ALFAN S	8.5	8.6	8.7	9.00	9.0	91.7	91.7
55	ELMI SOFIA REZA	9.0	9.0	8.9	9.00	8.9	96.7	96.7
56	ELON RIDEMO TRI PARTAI PANGGAB	9.0	8.9	9.0	9.00	8.9	96.9	96.9
57	HELMILIA CONTESA	8.9	8.8	8.6	8.90	9.0	92.8	92.8
58	HENGGI FERNANDITO SIBARANI	9.0	8.8	8.7	8.80	8.6	96.9	96.9
59	ILHAM PRATAMA	8.9	8.6	8.5	8.60	9.0	96.2	96.2
60	IRVANDRY GULTOM	9.0	8.8	8.5	8.60	9.0	92.8	92.8

61	JAYROZI ACHMAD	8,7 0	8,8 0	8,5 0	8,70	9,0 0	95,7	95,7
62	LUSI LAWES PANGGABEAN	8,6 0	8,6 0	8,5 0	8,90	8,5 0	89,4	89,4
63	REZA NOOR	8,6 0	8,7 0	8,6 0	8,60	8,7 0	95,1	95,1
64	RINI WATI	8,6 0	8,6 0	8,6 0	8,50	8,6 0	90,3	90,3
65	RIO PRESLEY HUTAGALUNG	8,8 0	8,8 0	8,7 0	8,60	8,5 0	94,1	94,1
66	VELLY JUNARTI	8,6 0	8,6 0	8,8 0	8,60	8,9 0	90,1	90,1
67	YAYANG ANGGARI	8,7 0	8,6 0	8,6 0	8,70	9,0 0	89,7	89,7
68	ZAKQIA WARDANI	8,8 0	8,7 0	8,6 0	8,50	8,7 0	92,8	92,8

#### 4.2.2 Normalisasi Data Real

Proses normalisasi merupakan suatu langkah kerja dalam memindahkan angka dari kolom menjadi baris dan dari bilangan bulat menjadi pecahan, hal ini dilakukan agar data tadi mudah untuk dilakukan proses perkalian bobot pada matlab karna memiliki angka pecahan atau dinormalisasikan. Untuk menentukan estimasi Ujian Nasional Produktif Kompetensi ini, dilakukan proses pelatihan dan pengujian Jaringan Syaraf Tiruan. Pada penyelesaian permasalahan ini penulis mengambil analisis data Ujian Nasional Produktif Kompetensi 2014-2015. Langkah awal dalam melakukan normalisasi adalah dengan menentukan data yang akan kita jadikan *input*, kemudian menentukan nilai paling maksimum dan nilai paling minimum dari data jumlah ujian nasional produktif kompetensi. Data maksimum dan minimum ini dijadikan sebagai nilai untuk melengkapi rumus dari normalisasi data pada setiap tahun tersebut untuk mendapatkan nilai terbaik, berikut ini rumus dari normalisasi.:

$$X_n = 0.8 * (S_n - \text{Min}) / (\text{Max} - \text{Min}) + 0.1 = N$$

Keterangan :

$X_n$  = *input layer*, *input* yang terdiri dari jumlah penentuan ujian nasional produktif kompetensi.

$S_n$  = Nilai dari yang digunakan untuk proses normalisasi.

Max = Nilai Tertinggi dari Rekap Data Siswa.

Mix = Nilai Terendah dari Rekap Data Siswa.

N = Output dari Nilai Normalisasi

Berikut ini dilakukan *Input* proses normalisasi pada table 4.2 , maka berdasarkan tabel 4.1 didapatkan data maksimumnya 100, dan data minimumnya 70.

**Tabel 4.2.2 Proses data normalisasi**

	Input	Nilai Normalisasi
X1	80.0	0.3667
X2	83.0	0.4467

X3	82.0	0.4200
X4	79.0	0.3400
X5	80.0	0.3667
X6	100.0	0.9000
T	91	0.6680

Untuk Tahun 2014-2015 diperoleh sebagai berikut :

$$X1 = 0.8 * (80 - 70) / 100 - 70 + 0.1 = 0,3667$$

$$X2 = 0.8 * (83 - 70) / 100 - 70 + 0.1 = 0,4467$$

$$X3 = 0.8 * (82 - 70) / 100 - 70 + 0.1 = 0,4200$$

$$X4 = 0.8 * (79 - 70) / 100 - 70 + 0.1 = 0,3400$$

$$X5 = 0.8 * (80 - 70) / 100 - 70 + 0.1 = 0,3667$$

$$X6 = 0.8 * (100 - 70) / 100 - 70 + 0.1 = 0,9000$$

Kemudian untuk *Target* pada pelatihan Jaringan Syaraf Tiruan. Normalisasi selanjutnya juga tetap menggunakan rumus seperti sebelumnya, berikut adalah tahapan normalisasi selanjutnya :

$$T = 0.8 * (91 - 70) / 100 - 70 + 0.1 = 0,6680$$

Proses normalisasi ini berguna untuk mendapatkan data yang dapat digunakan sebagai *input*-an pada Jaringan Syaraf Tiruan Hasil normalisasi data dapat dilihat pada tabel 4.3, dengan pola data ujian nasional produktif kompetensi pada tahun 2014-2015.

**Tabel 4.2.2 Hasil normalisasi 2014 s/d tahun 2015**

N o	NAMA SISWA	x1 SM TR I	x2 SMS TR II	x3 SMS TR III	x4 SMS TR IV	x5 SM TR V	x6 PRA. KOMP ETENS I	T UN KOMPET ENSI
1	AIDIL FAJAR AKBAR	0,36 67	0,446 7	0,420 0	0,340 0	0,36 67	0,9000	0,6680
2	ALFA EDISON NAHAK	0,10 00	0,233 3	0,286 7	0,366 7	0,34 00	0,8467	0,8067
3	ANDRIAN PRANATA	0,18 00	0,260 0	0,233 3	0,233 3	0,26 00	0,5133	0,7080
4	AZZAER MULAHAPAN PUTRA	0,23 33	0,233 3	0,233 3	0,233 3	0,26 00	0,6333	0,3667
5	EBHEN HAEZER SITOHANG	0,31 33	0,313 3	0,233 3	0,340 0	0,26 00	0,8200	0,5267
6	ERAL VALY	0,39 33	0,446 7	0,286 7	0,446 7	0,39 33	0,9000	0,5347
7	FEBRI RAMADHAN SAPUTRA	0,26 00	0,260 0	0,233 3	0,340 0	0,36 67	0,8467	0,4733
8	RIO SAFUTRA NASUTION	0,34 00	0,446 7	0,233 3	0,366 7	0,36 67	0,8467	0,4733
9	SANDI SUPERI	0,36 67	0,340 0	0,233 3	0,366 7	0,36 67	0,9000	0,5347
10	SUPRIYANTO	0,39 33	0,420 0	0,233 3	0,366 7	0,36 67	0,9000	0,6680

#### 4.2.3 Data Input dan Target

Sebelum proses pengolahan data dilakukan, perlu dilakukan proses penentuan masukan X1 sampai dengan X6 sebagai *input* serta *target* atau hasil yang diinginkan dari proses pengolahan data, setelah

penentuan input dan target maka langkah selanjutnya proses penentuan pola sehingga hasil yang didapatkan pada akhir penelitian ini sesuai dengan yang diharapkan penulis. Pada tabel 4.2.3 merupakan pola rekap nilai smk nasional al huda pekanbaru pada tahun 2014 sampai dengan 2015.

**Tabel 4.2.3 Pola rekap nilai tahun 2014 s/d tahun 2015**

	X1	X2	X3	X4	X5	X6	Target
Pola1	0.3667	0.4467	0.4200	0.3400	0.3667	0.9000	0.6680
Pola2	0.1000	0.2333	0.2867	0.3667	0.3400	0.8467	0.8067
Pola3	0.1800	0.2600	0.2333	0.2333	0.2600	0.5133	0.7080
Pola4	0.2333	0.2333	0.2333	0.2333	0.2600	0.6333	0.3667
Pola5	0.3133	0.3133	0.2333	0.3400	0.2600	0.8200	0.5267
Pola6	0.3933	0.4467	0.2867	0.4467	0.3933	0.9000	0.5347
Pola7	0.2600	0.2600	0.2333	0.3400	0.3667	0.8467	0.4733
Pola8	0.3400	0.4467	0.2333	0.3667	0.3667	0.8467	0.4733
Pola9	0.3667	0.3400	0.2333	0.3667	0.3667	0.9000	0.5347
Pola10	0.3933	0.4200	0.2333	0.3667	0.3667	0.9000	0.6680

Pada tabel 4.2.3 diatas , penulis hanya melakukan pengambilan sampel sebanyak 10 pola dimana 5 merupakan pola rekap nilai smk nasional al huda pekanbaru Untuk melakukan *Pelatihan*, Pola yang diperlukan 5 Pola, seperti yang tertera pada tabel 4.2.3.

**Tabel 4.2.3 Pola data pelatihan Tahun 2014 s/d Tahun 2015**

	X1	X2	X3	X4	X5	X6	Target
Pola1	0.3667	0.4467	0.4200	0.3400	0.3667	0.9000	0.6680
Pola2	0.1000	0.2333	0.2867	0.3667	0.3400	0.8467	0.8067
Pola3	0.1800	0.2600	0.2333	0.2333	0.2600	0.5133	0.7080
Pola4	0.2333	0.2333	0.2333	0.2333	0.2600	0.6333	0.3667
Pola5	0.3133	0.3133	0.2333	0.3400	0.2600	0.8200	0.5267

Sedangkan untuk 5 lagi merupakan pola rekap nilai smk nasional al huda pekanbaru untuk melakukan *Pengujian*, Pola yang diperlukan 5 Pola, seperti yang tertera pada tabel 4.2.3.

**Tabel 4.6 Pola Data Pengujian Tahun 2014 s/d Tahun 2015**

Pola6	0.3933	0.4467	0.2867	0.4467	0.3933	0.9000	0.5347
Pola7	0.2600	0.2600	0.2333	0.3400	0.3667	0.8467	0.4733
Pola8	0.3400	0.4467	0.2333	0.3667	0.3667	0.8467	0.4733

Pola9	0.3667	0.3400	0.2333	0.3667	0.3667	0.9000	0.5347
Pola10	0.3933	0.4200	0.2333	0.3667	0.3667	0.9000	0.6680

#### 4.2.4 Perancangan Manual Jaringan Syaraf Tiruan

Sampel data input dalam penelitian ini adalah data ujian nasional produktif kompetensi sebagai contoh dengan menggunakan 6 variabel input, yaitu X1 = 0.3667, X2 = 0.4467, X3 = 0.4200, X4 = 0.3400, X5 = 0.3667, X6 = 0.9000.

Tahap *Initialitation* pola 5-3-1 :

a. Tahapan dalam proses inisialisasi ini adalah menginisialisasi data *input*, bobot dari *input* ke pada *hidden layer* lalu ke *output layer*. Yang akan menjadi nilai input adalah X1-Xn, dengan menggunakan *learning rate default*= 0.2 yaitu :

$$\begin{aligned} X1 &= 0.3667, \\ X2 &= 0.4467, \\ X3 &= 0.4200, \\ X4 &= 0.3400, \\ X5 &= 0.3667, \\ X6 &= 0.9000. \end{aligned}$$

$$\text{Target} = 0.6680 \quad \text{Learning Rate}$$

*Default* ( $\alpha$ ) = 0.2

b. nilai bobot dari *Input Layer* ke *Hidden Layer*. Pada tabel 4.2.4 akan terlihat bobot dari *input layer* ke *hidden layer* yang penulis ambil dari Sistem Matlab 6.1.

**Tabel 4.2.4 Bobot yang diberikan dari input layer ke hidden layer**

	V1	V2	V3
x1	18.0698	-8.0070	3.6420
x2	-0.7033	14.5640	11.1656
x3	-2.4964	-20.4840	15.6504
x4	-4.4379	6.8729	19.8952
x5	42.3248	17.7320	-27.5828
x6	-2.6106	8.9444	9.8011
b	-16.6109	-10.3309	-10.6472

c. Berikan nilai bobot dari *HiddenLayer* ke *Output Layer*. Pada tabel 4.2.4 akan terlihat bobot *hidden layer* ke *output* yang penulis ambil dari Sistem Matlab 6.1.



**Tabel 4.2.4 Bobot yang diberikan hidden layer ke output layer**

	L
Z <sub>1</sub>	-0.1795
Z <sub>2</sub>	0.7873
Z <sub>3</sub>	-0.8842
b	-0.2943

Tahap 1 : Perhitungan maju

1. Perhitungan nilai *Output* dari unit tersembunyi atau *hidden layer*

Sebagai sampel data untuk penyelesaiannya dapat diambil dari  $z_{net1} - z_{net3}$ , Berdasarkan rujukan pada landasan teori maka rumus yang di gunakan adalah yaitu :  $Z_{net1} = v1 + (x1 * v11) + (x2 * v21) + (x3 * v31) + (x4 * v41) + (x5 * v51) + (x6 * v61)$ .

- $z_{net1} = -16.6190 + (18.0698 * 0.3667) + (-0.7033 * 0.4467) + (-2.4964 * 0.4200) + (-4.4379 * 0.3400) + (42.3248 * 0.3667) + (-2.6106 * 0.9000) = 0.9517$
- $Z_{net2} = v2 + (x1 * v12) + (x2 * v22) + (x3 * v32) + (x4 * v42) + (x5 * v52) + (x6 * v62)$   
 $z_{net2} = -10.3309 + (-80070 * 0.3667) + (14.5640 * 0.4467) + (-20.4840 * 0.4200) + (6.8279 * 0.3400) + (17.7320 * 0.3667) + (8.9444 * 0.9000) = 1.5237$
- $Z_{net3} = v3 + (x1 * v13) + (x3 * v23) + (x3 * v33) + (x4 * v43) + (x5 * v53) + (x6 * v63)$   
 $z_{net3} = -10.6472 + (3.6420 * 0.3667) + (11.1656 * 0.4467) + (15.6504 * 0.4200) + (19.8952 * 0.3400) + (-27.5828 * 0.3667) + (9.8011 * 0.9000) = 7.7203$

2. Kemudian dilanjutkan dengan menghitung nilai keluaran dengan menggunakan fungsi aktivasi yang dipilih, di mana fungsi aktivasi yang digunakan adalah fungsi sigmoid biner yang mempunyai persamaan, :

- $z1 = \text{sigmoid} [0.9517] = 0.7215$
- $z2 = \text{sigmoid} [1.5237] = 0.8211$
- $z3 = \text{sigmoid} [7.7203] = 0.9996$

3. Menghitung nilai output pada unit  $yk$  ( $k=1,2,3,\dots,m$ ) dengan menggunakan nilai bobot-nya, yang dijelaskan pada rumus :

$$Yink = (-0.2943 + 0.7215) * (-0.1795 + 0.8211) * (0.7873 + 0.9996) * (-0.8842$$

$$= -0.66118$$

4. Kemudian dihitung nilai Output dengan menggunakan fungsi aktivasi :

$$= \text{sigmoid} [-0.66118] = \frac{1}{1 + e^{-y_{net k}}} = 0.3405$$

Fase II perhitungan mundur

1. Untuk tiap unit keluaran ( $k=1,\dots,m$ ) menerima pola target yang bersesuaian dengan pola masukan, dan kemudian dihitung informasi kesalahan Hitung faktor  $\delta$  diunit keluaran berdasarkan kesalahan setiap unit keluaran  $yk$ , rumus yang digunakan:

$$\delta 1 = (0.6680 - 0.3405) * 0.3405 (1 - 0.3405) = 0.0735$$

2. Suku perubahan bobot  $Wjk$  dilakukan perhitungan (yang akan digunakan untuk merubah bobot  $Wjk$ ) dengan laju pelatihan *learning rate*  $\alpha=0.2$ , rumus yang digunakan:

1. Koreksi nilai bobot yang kemudian digunakan

untuk memperbaharui nilai  $W10$  :

$$\Delta W10 = 0.2 * 0.0735 * 1 = 0.0147$$

2. Koreksi nilai bobot yang kemudian digunakan

untuk memperbaharui nilai  $W11$  :

$$\Delta W11 = 0.2 * 0.0055 * 0.7215 = 0.0106$$

3. Koreksi nilai bobot yang kemudian digunakan

untuk memperbaharui nilai  $W12$  :

$$\Delta W12 = 0.2 * 0.0055 * 0.8211 = 0.0121$$

4. Koreksi nilai bobot yang kemudian digunakan

untuk memperbaharui nilai  $W13$  :

$$\Delta W13 = 0.2 * 0.0055 * 0.9996 = 0.0147$$

3. Untuk setiap ( $Zj, j=1,\dots,p$ ) dihitung delta masukan yang berasal dari neuron pada layer di atasnya:

- hidden layer =  $\delta k$  0.0735 dan  $Wkj$  -0.1795:

$$-\delta_{net 1} = 0.0735 * -0.1795 = -0.0132$$

- hidden layer =  $\delta k$  0.0735 dan  $Wkj$  0.7873 :

$$-\delta_{net 2} = 0.0735 * 0.7873 = 0.0579$$

- hidden layer =  $\delta k$  0.0735 dan  $Wkj$  -0.8842 :

$$-\delta_{net 3} = 0.0735 * -0.8842 = -0.0650$$

4. Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung informasi kesalahan, berdasarkan rujukan pada :

a) Nilai kesalahan pada *hidden layer*  $\delta 1$  yaitu :

$$\delta 1 = -0.0132 * 0.7215 * (1 - 0.7215)$$

$$\delta 1 = -0.0027$$

b) Nilai kesalahan pada *hidden layer*  $\delta 2$  yaitu :

$$\delta 2 = 0.0579 * 0.8211 * (1 - 0.8211)$$

$$\delta 2 = 0.0085$$

c) Nilai kesalahan pada *hidden layer*  $\delta 3$  yaitu :

$$\delta 3 = -0.0650 * 0.9996 * (1 - 0.9996)$$

$$\delta 3 = 0.0000$$



Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui  $v_{ij}$ , berdasarkan rujukan (proses 2.6c) pada :

$$\begin{aligned} \Delta V11 &= 0.2 * -0.0027 * 0.8733 = -0.0004634 \\ \Delta V21 &= 0.2 * -0.0027 * 0.8319 = -0.0004414 \\ \Delta V12 &= 0.2 * 0.0085 * 0.8733 = 0.0014858 \\ \Delta V22 &= 0.2 * 0.0085 * 0.8319 = 0.0014152 \\ \Delta V13 &= 0.2 * 0.0000 * 0.8733 = -0.0000050 \\ \Delta V23 &= 0.2 * 0.0000 * 0.8319 = 0.0000048 \end{aligned}$$

Proses akan tetap berlanjut hingga  $v_{53}$  sehingga bobot  $v_{53}$  didapatkan data sebagai berikut :

$$\begin{aligned} \Delta V11 &= -0.0004634 & \Delta V12 &= 0.0014858 \\ \Delta V13 &= -0.0000050 \\ \Delta V21 &= -0.0004414 & \Delta V22 &= 0.0014152 \\ \Delta V23 &= 0.0000048 \\ \Delta V31 &= -0.0001678 & \Delta V32 &= 0.0005381 \\ \Delta V33 &= -0.0000018 \\ \Delta V41 &= -0.0000405 & \Delta V42 &= 0.0001298 \\ \Delta V43 &= -0.0000004 \\ \Delta V51 &= -0.0000074 & \Delta V52 &= 0.0008859 \\ \Delta V53 &= -0.0000030 \end{aligned}$$

5. Tiap nilai bias dan bobot ( $j=0, \dots, p$ ) pada output neuron ( $Y_k, k=1, 2, 3, \dots, m$ ) diperbaharui, berdasarkan rujukan pada :

$$\begin{aligned} W1 \text{ (baru)} &= 0.8436 + 0.0106 = 0.8542 \\ W2 \text{ (baru)} &= 0.4764 + 0.0121 = 0.4885 \\ W3 \text{ (baru)} &= -0.6475 + 0.0147 = -0.6328 \end{aligned}$$

Bobot garis mengalami perubahan pada saat terhubung ke *hidden layer*, berdasarkan rujukan pada (2.7b):

$$\begin{aligned} V11 \text{ (baru)} &= 13.6473 + -0.0004634 = 13.646837 \\ V21 \text{ (baru)} &= -0.8501 + -0.0004414 = -0.850541 \end{aligned}$$

Pada tahapan Selanjutnya, proses akan tetap berjalan hingga  $v_{23}$

$$\begin{aligned} V12 \text{ (baru)} &= -5.4165 + 0.0014858 = -5.415014 \\ V22 \text{ (baru)} &= 15.7691 + 0.0014152 = 15.770515 \end{aligned}$$

$$\begin{aligned} V13 \text{ (baru)} &= 2.8488 + 0.0000050 = 2.848795 \\ V23 \text{ (baru)} &= 13.9793 + 0.00000048 = 13.979295 \end{aligned}$$

Pada bobot  $v_{63}$  maka proses akan terhenti, sehingga pada iterasi  $P=1$  pada matlab dapat ditampilkan suku perubahan bobot akhir antara input ke *hidden*, dan bobot *hidden* ke *output* seperti yang terlampir pada tabel di bawah ini :

**Tabel 4.2.4 Perubahan bobot akhir pada  $p=1$  dari input ke hidden layer**

	V1	V2	V3
x1	18.0698	-8.0070	3.6420

x2	-0.7033	14.5640	11.1656
x3	-2.4964	-20.4840	15.6504
x4	-4.4379	6.8729	19.8952
x5	42.3248	17.7320	-27.5828
x6	-2.6106	8.9444	9.8011
1	-16.6109	-10.3309	-10.6472

Sedangkan pada matlab dapat ditampilkan suku perubahan bobot akhir antara *Hidden* ke *Output Layer* seperti yang terlampir pada tabel 4.2.4 di di bawah ini.:

**Tabel 4.2.4 Perubahan bobot akhir pada  $p=1$  dari hidden ke output layer**

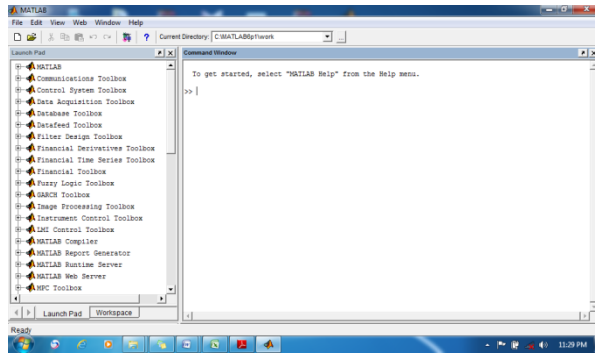
	L
Z <sub>1</sub>	-0.1795
Z <sub>2</sub>	0.7873
Z <sub>3</sub>	-0.8842
1	-0.2943

## 5. Implementasi Dan Pengujian

### 5.1 Hasil Pengujian

Pada pembahasan point ini, penulis masih menggunakan proses perhitungan manual, saat ini penulis akan bereksperimen pada penerapan sistem metode *backpropagation* dengan *software* Matlab 6.1 yang digunakan pada penelitian ini.

Matlab 6.1 adalah *software* yang dirancang dengan menggunakan visual *interface* yang dapat memudahkan user dalam mengisi pola inputan sampai proses menemukan output. Matlab dirancang khusus memiliki *commandwindow* untuk mengisikan *script* yang kita perlukan dan juga *commandhistory* sehingga apabila data *script* input ingin kita tampilkan kembali dapat menggunakan fasilitas ini, berikut bentuk tampilan dari Matlab 6.1 :



Gambar 8. Tampilan jendela awal matlab 6.1

## 5.2 Pengujian Sistem

Pada bagian ini akan dijelaskan hasil yang diperoleh dari implementasi sistem terhadap *software* Matlab 6.1.

### 5.1.1 Pelatihan Data Pola 6-3-1

Untuk melatih data pola 6-3-1 akan diolah menggunakan Matlab 6.1 dengan input sebanyak 6, nilai input ini didapat dari tabel 4.4, terdiri dari 3 hidden dan 1 output dengan tahapan sebagai berikut :

1. Memasukkan data ke sistem Matlab 6.1 (input) / (p)  
Pada tahapan ini dilakukan proses input data pada pola yang sudah ditetapkan sebelumnya dengan menggunakan pola 6-3-1, dengan menggunakan command window

```
>> p=[
0.3667 0.1000 0.1800 0.2333 0.3133;
0.4467 0.2333 0.2600 0.2333 0.3133;
0.4200 0.2867 0.2333 0.2333 0.2333;
0.3400 0.3667 0.2333 0.2333 0.3400;
0.3667 0.3400 0.2600 0.2600 0.2600;
0.9000 0.8467 0.5133 0.6333 0.8200;
]
```

2. Memasukkan Nilai target ke sistem Matlab 6.1

Pada tahapan ini proses menginput data pada pola yang sudah ditargetkan, dengan menggunakan command window

```
>> t=[0.6680 0.8067 0.7080 0.3667 0.5267].
```

3. Membentuk pola Jaringan Syaraf Tiruan metode *Backpropagation* Pada tahapan ini, Pola jaringan akan dibentuk menjadi pola 6-3-1 dengan menggunakan *command window* Matlab 6.1 :

```
net=newff(minmax(p),[3,1],{'logsig','purelin'},'traingd
x');
```

Keterangan :

- logsig* merupakan fungsi sigmoid binner
- purelin* merupakan fungsi linear

c. *traingdx* merupakan cara membangun Jaringan Syaraf dengan metode Pembelajaran Algoritma *Backpropagation*

4. Melihat bobot awal dan bobot bias pertama dari *input layer* ke *hidden layer* dengan *command window* pada matlab:

```
>>net.IW{1,1}
```

```
>>net.b{1}
```

Keterangan :

a. *net.IW* digunakan sebagai variabel untuk menyimpan bobot dari unit masukan ke unit tersembunyi. Karena dalam *Backpropagation*, unit masukan hanya terhubung dengan layar tersembunyi paling bawah, maka bobotnya disimpan dalam *net.IW{1,1}*.

b. *net.b{1}* merupakan bobot bias awal lapisan input ke lapisan tersembunyi pertama.

Setelah tahapan tadi terbentuk maka didapatkan nilai bobot awal dari *input layer* ke *hidden layer* dan bobot bias *input layer* ke *hidden layer* dapat dilihat pada tabel 5.1.1

Tabel 5.1.1 Bobot Awal *Input Layer* ke *Hidden Layer* Pola 6-3-1

	V1	V2	V3
x1	18.0698	-8.0070	3.6420
x2	-0.7033	14.5640	11.1656
x3	-2.4964	-20.4840	15.6504
x4	-4.4379	6.8729	19.8952
x5	42.3248	17.7320	-27.5828
x6	-2.6106	8.9444	9.8011
1	-16.6109	-10.3309	-10.6472

5. Selanjutnya melihat nilai bobot dan bobot bias dari *hidden layer* ke *output layer* dengan *command window*, dengan script matlab:

```
>>net.LW{2,1}
```

```
>>net.b{2}
```

Keterangan :

a. *net.LW* digunakan untuk menyimpan bobot dari layar tersembunyi paling bawah ke layar tersembunyi di atasnya.

b. *net.b{2}* merupakan bobot bias awal lapisan tersembunyi pertama ke lapisan tersembunyi kedua.

Setelah tahapan di atas dilanjutkan maka didapat nilai hidden layer ke output layer dan bobot bias hidden layer ke output layer dapat dilihat pada tabel 5.1.1

**Tabel 5.1.1 Bobot hidden layer ke output layer pola**

6-3-1	
	L
Z1	-0.1795
Z2	0.7873
Z3	-0.8842
1	-0.2943

6. Menetapkan parameter yang akan digunakan

Sebelum melakukan proses pengujian data untuk proses pelatihan maka perlu menetapkan parameter yang akan digunakan sebagai acuan dalam melaksanakan proses train nantinya, berikut tahapan dari parameter yang digunakan :

```
>>net.trainParam.epoch=1000;
```

Parameter ini memiliki fungsi sebagai penentu pada epochs seberapa proses pelatihan atau pengujian berhenti, sehingga epochs yang didapat berada dalam keadaan maksimal dari proses pengujian dan pelatihan pada sistem Matlab 6.1.

```
>>net.trainParam.goal=0.1;
```

Parameter ini berfungsi memberikan batasan terhadap nilai MSE proses iterasi dapat berhenti. Kondisi iterasi pengujian berhenti jika nilai MSE < batas yang ditentukan dalam net.trainParam.goal atau jumlah epochs yang telah ditentukan dalam net.trainParam.epoch, sehingga jika MSE bernilai > dari batas yang ditentukan maka iterasi belum bisa berhenti.

```
>>net.trainParam.Lr=0.2;
```

Parameter ini bertujuan agar dapat mengoptimalkan dari laju pembelajaran dan pemahaman sistem. ( $\alpha$  = learning rate). Biasanya digunakan dengan Range Default =0.1, 0.2 ke-n. Pada kasus ini digunakan learning rate Default = 0.2. Ketentuan dalam pemakaian learning rate adalah jika nilai ( $\alpha$ ) besar maka proses semakin cepat, namun sebaliknya jika ( $\alpha$ ) terlalu besar maka akan berpengaruh terhadap kinerja algoritma, menjadi tidak stabil.

```
>>net.trainParam.show=100;
```

Untuk melihat perubahan bentuk MSE maka dilakukan train dengan (default : setiap 25 epoch), jika pada epochs ke 25 belum berhenti maka proses akan tetap berlanjut ke epoch 26 sampai matlab menemukan epochs terbaik.

6. Melihat nilai output

Hasil output yang dikeluarkan oleh Jaringan dapat disimulasikan menggunakan perintah sebagai berikut :

```
>>[y,Pf,Af,e,perf] = sim(net,p,[ ],[ ],t)
```

Keterangan :

Y : keluaran jaringan

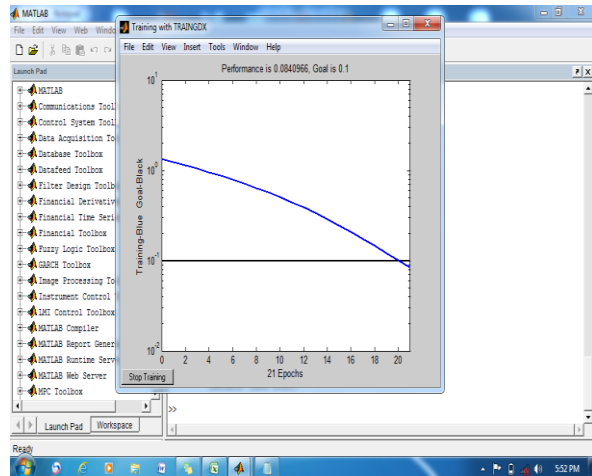
Pf : kondisi akhir masukan

Af : kondisi akhir layar

E : error jaringan = T-Y

Perf : kinerja jaringan

Setelah dilakukan perulangan maka ditemukan error minimum pada epochs ke 19 seperti pada gambar 5.1.1.



**Gambar 9. Hasil pelatihan pola 6-3-1**

Gambar 5.1.1 di atas menunjukkan hasil pelatihan dari pola 6-3-1 yang mencapai goal atau nilai akhir pada epoch atau iterasi ke 21 dan error 0.0840966.

**5.1.2 Pelatihan Data Model 6-5-1**

Kemudian dilanjutkan dengan pola yang ke dua yaitu Untuk menguji data pola 6-5-1 akan diolah menggunakan Matlab 6.1 dengan input sebanyak 5, nilai input ini didapat dari tabel 4.1, terdiri dari 5 hidden dan 1 output dengan tahapan sebagai berikut :

1. Memasukkan data ke sistem Matlab 6.1 (input) / (p) Pada tahapan ini dilakukan proses input data pada pola yang sudah ditetapkan sebelumnya dengan

menggunakan pola 5 pola data, dengan 3 hidden dan 1 output, menggunakan fasilitas command window seperti sikrip dibawah ini :

```
>>p=[
0.3667 0.1000 0.1800 0.2333 0.3133;
0.4467 0.2333 0.2600 0.2333 0.3133;
0.4200 0.2867 0.2333 0.2333 0.2333;
0.3400 0.3667 0.2333 0.2333 0.3400;
0.3667 0.3400 0.2600 0.2600 0.2600;
0.9000 0.8467 0.5133 0.6333 0.8200;]
```

## 2. Memasukkan Nilai target ke sistem Matlab 6.1

Setelah proses menginput data pada pola sudah selesai, sekarang kita inputkan pula data yang sudah ditargetkan, dengan menggunakan command window :

```
>>t=[0.6680 0.8067 0.7080 0.3667 0.5267].
```

3. Membentuk pola Jaringan Syaraf Tiruan metode Backpropagation Pada tahapan ini, Pola jaringan akan dibentuk menjadi pola 5 input layer, 5 hidden layer dan 1 output layer dengan menggunakan command window Matlab 6.1 :

```
net=newff(minmax(p),[5,1],{'logsig','purelin'},'traingd
x');
```

Keterangan :

- logsig merupakan fungsi sigmoid biner
- purelin merupakan fungsi linear
- traingdx merupakan cara membangun Jaringan Syaraf dengan metode Pembelajaran Algoritma Backpropagation.

4. Lakukan proses menampilkan bobot awal dan bobot bias pertama dari input layer ke hidden layer dengan command window pada Matlab 6.1 seperti pada syntax dibawah ini :

```
>>net.IW{1,1}
>>net.b{1}
```

Keterangan :

- net.IW digunakan sebagai variabel untuk menyimpan bobot dari unit masukan ke unit tersembunyi. Karena dalam Backpropagation, unit masukan hanya terhubung dengan layer tersembunyi paling bawah, maka bobotnya disimpan dalam net.IW{1,1}.
- net.b{1} merupakan bobot bias awal lapisan input ke lapisan tersembunyi pertama.

Setelah tahapan tadi terbentuk maka didapatlah nilai bobot awal dari input layer ke hidden layer dan bobot bias input layer ke hidden layer dapat dilihat pada tabel 5.1.2

**Tabel 5.1.2 Bobot awal input layer ke hidden layer dengan pola 6-5-1**

	V1	V2	V3	V4	V5
X1	16.2661	-10.8673	3.4968	-0.5672	13.4612
X2	11.8369	-2.1990	-19.6945	16.2541	-2.3774
X3	5.9587	16.8563	19.7207	13.7693	-15.9089
X4	-6.8123	35.1899	27.279	-7.2591	27.0739
X5	-39.9331	-14.8648	25.6188	-49.5743	-31.0507
X6	-7.4079	-8.3987	2.3428	-6.3578	-7.1459
1	6.3635	-0.356	-18.4285	10.4821	13.1825

5. Menampilkan nilai bobot dan bobot bias dari hidden layer ke output layer dengan perintah matlab:

```
>>net.LW{2,1}
```

```
>>net.b{2}
```

Keterangan :

a. net.LW digunakan untuk menyimpan bobot dari layer tersembunyi paling bawah ke layer tersembunyi di atasnya.

b. net.b{2} merupakan bobot bias awal lapisan tersembunyi pertama ke lapisan tersembunyi kedua.

Setelah tahapan di atas dilanjutkan maka didapat nilai hidden layer ke output layer dan bobot bias hidden layer ke output layer dapat dilihat pada tabel 5.1.2

**Tabel 5.1.2 Bobot hidden layer ke output layer pola 6-5-1**

	L
K1	-0.9695
K2	0.4936
K3	-0.1098
K4	0.8636
K5	-0.068
1	-0.1627

6. Menetapkan parameter yang akan digunakan Sebelum melakukan pengujian data untuk proses pelatihan, maka perlu ditetapkan terlebih dahulu parameter-parameter yang dibutuhkan dalam proses pelatihan. Adapun parameter-parameter yang dibutuhkan adalah sebagai berikut :

```
>>net.trainParam.epoch=1000;
```

Parameter ini memiliki fungsi sebagai penentu pada epochs beberapa proses pelatihan atau pengujian berhenti, sehingga epochs yang didapat berada dalam keadaan maksimal dari proses pengujian dan pelatihan pada sistem Matlab 6.1.

```
>>net.trainParam.goal=0.1;
```

Parameter ini digunakan untuk menentukan batas nilai MSE agar iterasi dihentikan. Iterasi akan berhenti jika  $MSE < \text{batas}$  yang ditentukan dalam `net.trainParam.goal` atau jumlah epochs yang telah ditentukan dalam `net.trainParam.epoch`, sehingga jika MSE bernilai  $>$  dari batas yang ditentukan maka iterasi belum bisa berhenti.

```
>>net.trainParam.Lr=0.2;
```

Parameter ini digunakan untuk menentukan laju pemahaman ( $\alpha = \text{learning rate}$ ). Biasanya digunakan dengan Range Default =0.1, 0.2 ke-n, namun pada kasus ini digunakan Default = 0.2. Jika semakin besar nilai  $\alpha$ , maka semakin cepat pula proses pelatihan. Akan tetapi, jika nilai  $\alpha$  terlalu besar, maka algoritma menjadi tidak stabil mencapai titik minimum lokal.

```
>>net.trainParam.show=100;
```

Parameter ini digunakan untuk menampilkan frekuensi perubahan MSE (default : setiap 25 epoch), jika pada epochs ke 25 belum berhenti maka proses akan tetap berlanjut ke epoch 26 sampai matlab menemukan epochs terbaik.

#### 7. Melihat keluaran yang dihasilkan

Untuk melihat hasil output yang dikeluarkan oleh Jaringan dapat disimulasikan menggunakan perintah sebagai berikut :

```
>>[y,Pf,Af,e,perf] = sim(net,p,[ ],[ ],t)
```

Keterangan :

Y : keluaran jaringan

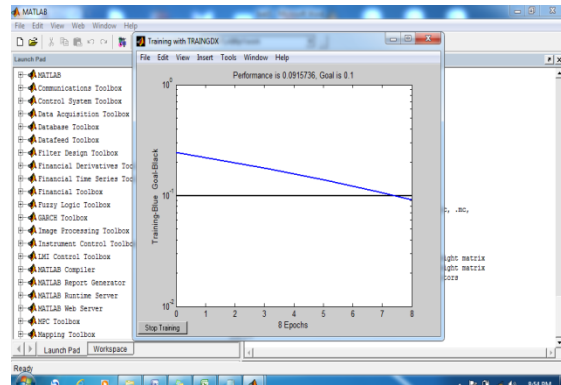
Pf : kondisi akhir masukan

Af : kondisi akhir layar

E : error jaringan = T-Y

Perf : kinerja jaringan

Setelah dilakukan perulangan maka ditemukan error minimum pada epochs ke 31 seperti pada gambar 10.



Gambar 10. Hasil pelatihan pola 6-5-1

Gambar 10. di atas menunjukkan hasil pelatihan dari pola 6-5-1 yang mencapai goal atau nilai akhir pada epoch atau iterasi ke 8 dan error 0.0915736.

#### 5.1.3 Pelatihan Data Model 6-6-1

Kemudian dilanjutkan dengan pola yang ke tiga yaitu untuk menguji data pola 6-6-1 akan diolah menggunakan Matlab 6.1 dengan input sebanyak 5, nilai input ini didapat dari tabel 4.1, terdiri dari 6 hidden dan 1 output dengan tahapan sebagai berikut :

##### 1. Memasukkan data ke sistem Matlab 6.1 (input) / (p)

Pada tahapan ini dilakukan proses input data pada pola yang sudah ditetapkan sebelumnya dengan menggunakan pola 5 pola data, dengan menggunakan command window >>p=[

```
0.3667 0.1000 0.1800 0.2333 0.3133;
0.4467 0.2333 0.2600 0.2333 0.3133;
0.4200 0.2867 0.2333 0.2333 0.2333;
0.3400 0.3667 0.2333 0.2333 0.3400;
0.3667 0.3400 0.2600 0.2600 0.2600;
0.9000 0.8467 0.5133 0.6333 0.8200;
]
```

##### 2. Memasukkan Nilai target ke sistem Matlab 6.1

Pada tahapan ini proses menginput data pada pola yang sudah ditargetkan, dengan menggunakan command window

```
>> t=[0.6680 0.8067 0.7080 0.3667 0.5267].
```

1. Membentuk pola Jaringan Syaraf Tiruan metode Backpropagation Pada tahapan ini, Pola jaringan akan dibentuk menjadi pola 5 input layer, 6 hidden layer dan 1 output layer dengan menggunakan command window Matlab 6.1 :

```
net=newff(minmax(p),[6,1],{'logsig','purelin'},'traingdx');
```

Keterangan :

a. `logsig` merupakan fungsi sigmoid binner

b. `purelin` merupakan fungsi linear

c. `traingdx` merupakan cara membangun Jaringan Syaraf dengan metode Pembelajaran Algoritma Backpropagation

1. Menampilkan bobot awal dan bobot bias pertama dari input layer ke hidden layer dengan command window pada matlab:

```
>>net.IW{1,1}
```

```
>>net.b{1}
```

Keterangan :

- net.IW digunakan sebagai variabel untuk menyimpan bobot dari unit masukan ke unit tersembunyi. Karena dalam B
- Backpropagation, unit masukan hanya terhubung dengan layer tersembunyi paling bawah, maka bobotnya disimpan dalam net.IW{1,1}.
- net.b{1} merupakan bobot bias awal lapisan input ke lapisan tersembunyi pertama.

Setelah tahapan tadi terbentuk maka didapatlah nilai bobot awal dari input layer ke hidden layer dan bobot bias input layer ke hidden layer dapat dilihat pada tabel 5.1.3

**Tabel 5.1.3 Bobot awal input layer ke hiddenlayerpola 6-6-1**

	V1	V2	V3	V4	V5	V6
X1	14.6547	-9.3314	4.2398	-0.8255	15.5516	9.0952
X2	-1.7713	-20.8853	15.9398	-4.0696	5.7335	12.6610
X3	19.6173	11.8101	-18.3512	-7.9320	24.7231	20.6664
X4	-5.8404	27.3147	-35.0748	-17.3219	24.8834	-34.0044
X5	-29.3857	-25.7857	-29.883	15.2773	-22.6309	-26.1241
X6	-10.8839	5.9073	-1.5026	17.5375	-0.9321	-1.9470
1	5.6514	3.3965	19.7788	-8.5711	-11.1045	10.3598

2. Menampilkan nilai bobot dan bobot bias dari hidden layer ke output layer dengan perintah matlab:

```
>>net.LW{2,1}
```

```
>>net.b{2}
```

Keterangan :

- net.LW digunakan untuk menyimpan bobot dari layer tersembunyi paling bawah ke layer tersembunyi di atasnya.
- net.b{2} merupakan bobot bias awal lapisan tersembunyi pertama ke lapisan tersembunyi kedua.

Setelah tahapan di atas dilanjutkan maka didapat nilai hidden layer ke output layer dan bobot bias hidden layer ke output layer dapat dilihat pada tabel 5.1.3

**Tabel 5.1.3 Bobot hidden layer ke output layer pola 6-6-1**

	L
K1	0.6924
K2	0.0503
K3	-0.5947
K4	0.3443
K5	0.6762

K6	-0.9607
1	0.3626

2. Menetapkan parameter yang akan digunakan

Sebelum melakukan pengujian data untuk proses pelatihan, maka perlu ditetapkan terlebih dahulu parameter-parameter yang dibutuhkan dalam proses pelatihan. Adapun parameter-parameter yang dibutuhkan adalah sebagai berikut :

```
>>net.trainParam.epoch=1000;
```

Parameter ini memiliki fungsi sebagai penentu pada epochs beberapa proses pelatihan atau pengujian berhenti, sehingga epochs yang didapat berada dalam keadaan maksimal dari proses pengujian dan pelatihan pada sistem Matlab 6.1.

```
>>net.trainParam.goal=0.1;
```

Parameter ini digunakan untuk menentukan batas nilai MSE agar iterasi dihentikan. Iterasi akan berhenti jika  $MSE < \text{batas}$  yang ditentukan dalam net.trainParam.goal atau jumlah epochs yang telah ditentukan dalam net.trainParam.epoch, sehingga jika MSE bernilai  $>$  dari batas yang ditentukan maka iterasi belum bisa berhenti.

```
>>net.trainParam.Lr=0.2;
```

Parameter ini digunakan untuk menentukan laju pemahaman ( $\alpha = \text{learning rate}$ ). Biasanya digunakan dengan Range Default =0.1, 0.2 ke-n, namun pada kasus ini digunakan Default = 0.2. Jika semakin besar nilai  $\alpha$ , maka semakin cepat pula proses pelatihan. Akan tetapi, jika nilai  $\alpha$  terlalu besar, maka algoritma menjadi tidak stabil mencapai titik minimum lokal.

```
>>net.trainParam.show=100;
```

Parameter ini digunakan untuk menampilkan frekuensi perubahan MSE (default : setiap 25 epoch), jika pada epochs ke 25 belum berhenti maka proses akan tetap berlanjut ke epoch 26 sampai matlab menemukan epochs terbaik.

3. Melihat keluaran yang dihasilkan

Untuk melihat hasil output yang dikeluarkan oleh Jaringan dapat disimulasikan menggunakan perintah sebagai berikut :

```
>>[y,Pf,Af,e,perf] = sim(net,p,[ ],[ ],t)
```

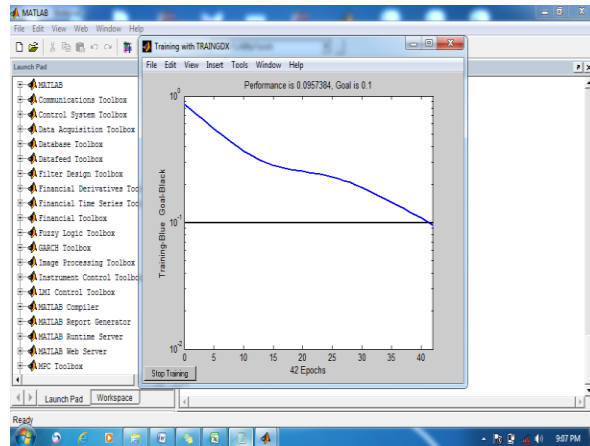
Keterangan :

Y : keluaran jaringan

Pf : kondisi akhir masukan

Af : kondisi akhir layar  
E : error jaringan = T-Y  
Perf : kinerja jaringan

Setelah dilakukan perulangan maka ditemukan error minimum pada epochs ke 47 seperti pada gambar 5.1.3



Gambar 11. Hasil pelatihan pola 6-6-1

Gambar 9. di atas menunjukkan hasil pelatihan dari pola 6-6-1 yang mencapai goal atau nilai akhir pada epoch atau iterasi ke 42 dan error 0.0957384.

## 4. Penutup

### a. Kesimpulan

Dari hasil pembahasan dan pengujian dengan menggunakan Software Matlab, maka dapat ditarik kesimpulan sebagai berikut:

- 1) Jaringan yang digunakan terdiri dari tiga buah lapisan yaitu lapisan input, lapisan tersembunyi dan lapisan keluaran.
- 2) Algoritma *Backpropagation* dapat melakukan proses prediksi, akan tetapi baik atau tidaknya nilai yang dihasilkan sangat dipengaruhi oleh penentuan parameter besarnya *Learning Rate* dan jumlah neuron pada *Hidden Layer*.
- 3) Semakin besar jumlah unit dalam hidden layer maka hasil prediksi akan semakin mendekati nilai yang ditargetkan.
- 4) Terdapat beberapa faktor yang mempengaruhi tingkat kebenaran prediksi pada Jaringan Syaraf Tiruan *Backpropagation* yaitu *learning rate*, target error, jumlah data dan nilai

bobot yang diberikan secara random pada tiap-tiap neuron.

### b. Saran

Berdasarkan pengamatan yang dilakukan agar penelitian yang akan dilakukan selanjutnya lebih baik lagi maka penulis menyarankan sebagai berikut:

- 1) Berdasarkan pada kesimpulan, disarankan agar diadakan penelitian lebih lanjut.
- 2) Prediksi Standar Kelulusan Ujian Nasional Produktif Kompetensi Di Smk Nasional Al Huda Pekanbaru dilakukan dengan metode yang lain misalnya menggunakan *fuzzy* atau yang lainnya.
- 3) Perlu dikembangkan penelitian yang lebih mendalam dan variasi *algoritma* pelatihan supaya mendapatkan hasil yang lebih optimal dengan waktu pelatihan yang lebih singkat lagi.
- 4) Lapisan tersembunyi lebih dari satu dan parameter yang digunakan di set lebih tepat lagi agar jaringan dapat mengenali pola semua pola data input. Agar dapat memprediksi dengan lebih akurat.

## 5. Referensi

- Anwar, B. (2011). Penerapan algoritma jaringan syaraf tiruan backpropagation dalam memprediksi tingkat suku bunga bank. *SAINTIKOM*, 10(2), 111–123.
- Komputer, J., Volume, E. I., Teknik, F., Universitas, K., Indonesia, K., Dipati, J., & No, U. (2012). BACKPROPAGATION DAN LEARNING VECTOR QUANTIZATION Program Studi Teknik Informatika Jurnal Komputer dan Informatika ( KOMPUTA ). *Komputer Dan Informatika (KOMPUTA)*, 1, 45–51.
- Lesnussa, Y. A., Latuconsina, S., & Persulesy, E. R. (2015). Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA ( Studi kasus : Prediksi Prestasi Siswa SMAN 4 Ambon ). *Matematika Integratif*, 11(2), 149–160.
- Mistianingsih, M. F. A., Barong, J., Unmul, K., Kelua, G., & Samarinda, S. (2010). Jumlah Pengangguran di Provinsi Kalimantan Timur Dengan Menggunakan Algoritma Pembelajaran Backpropagation. *Informatika Mulawarman*, 5(1), 50–54.
- Raharjo, J. S. D. (2013). MODEL ARTIFICIAL NEURAL NETWORK BERBASIS PARTICLE SWARM OPTIMIZATION UNTUK PREDIKSI LAJU INFLASI Joko S . Dwi Raharjo PENDAHULUAN Inflasi tidak dapat dihindarkan oleh setiap negara dan menjadi indikator tinggi akan Kesimpulan beberapa penelitian tentang p. *Sistem Komputer*, 3(1), 10–21.