



---

## Pemodelan *Data Mining* Menggunakan *Neural Network* Untuk Seleksi Mahasiswa Penerima Beasiswa Bantuan

Yogi Syarif Hidayat<sup>1\*</sup>, Ahmad Gunawan Herdipriyansyah<sup>2</sup>, Iska Asri Agustin<sup>3</sup>, Jaenudin<sup>4</sup>, Desrizal<sup>5</sup>

<sup>1,2,3,4,5</sup>Informatika/Universitas Lingga Buana PGRI Sukabumi

Email\*: [yogi.syarif@gmail.com](mailto:yogi.syarif@gmail.com)

\*) *Corresponding Author*

---

### ABSTRACT

*Scholarship is an important instrument in the course of a learning activity at a higher education institution, many of the criteria for the scholarship, among others, based on educational achievement, student activity, economic factors and the discipline of the students. Accuracy of the determination of the scholarship recipients are used for many reasons, primarily to the need for universities to provide convenience and availability of scholarship funds allocated to each student, and the opportunity to help economically disadvantaged students to remain able to continue their education at the college. For handling the problems above, the authors used data mining models based on neural network. Neural network model will be used for eligibility of students to receive scholarships. This model was chosen because of the process of learning and neural network classification algorithm is simple and fast. In general, the neural network algorithm model has a high degree of accuracy.*

**Keywords:** *Scholarship; Data Mining; Neural Network; Accuracy; Algorithm.*

### ABSTRAK

*Beasiswa merupakan instrument penting dalam proses berlangsungnya suatu kegiatan belajar pada suatu institusi perguruan tinggi, banyak kriteria untuk mendapatkan beasiswa, antara lain berdasarkan prestasi pendidikan, keaktifan mahasiswa, faktor ekonomi maupun kedisiplinan dari mahasiswa tersebut. Akurasi hasil penentuan penerima beasiswa digunakan untuk banyak alasan, utamanya adalah untuk kebutuhan bagi perguruan tinggi untuk memberikan kemudahan dan ketersediaan alokasi dana beasiswa pada masing-masing mahasiswa, dan kesempatan untuk membantu ekonomi mahasiswa yang kurang mampu untuk tetap bisa melanjutkan pendidikannya di perguruan tinggi tersebut. Untuk mengatasi permasalahan diatas, maka penulis menggunakan model data mining berbasis neural network. Model neural network digunakan untuk penentuan kelayakan seleksi penerimaan beasiswa bantuan untuk mahasiswa. Model ini dipilih karena proses learning dan klasifikasi pada algoritma neural network sederhana dan cepat. Secara umum, model algoritma neural network mempunyai tingkat akurasi yang tinggi.*

**Keywords:** *Beasiswa; Data Mining; Neural Network; Akurasi; Algoritma.*

---

## A. PENDAHULUAN

### 1. Latar Belakang

Beasiswa merupakan *instrument* penting dalam proses berlangsungnya suatu kegiatan belajar pada suatu institusi perguruan tinggi, banyak kriteria untuk mendapatkan beasiswa, antara lain berdasarkan prestasi pendidikan, keaktifan mahasiswa, faktor ekonomi maupun kedisiplinan dari mahasiswa tersebut. Pemberian beasiswa mampu memberikan pengaruh yang baik terhadap proses pembelajaran mahasiswa di institusi perguruan tinggi. Penentuan penerimaan beasiswa adalah penting karena hasil penentuan penerimaan beasiswa dapat memberikan kemudahan terhadap proses pendistribusian anggaran dari institusi perguruan tinggi dan kegiatan belajar mengajar. Tugas ini sangat rumit dan sangat sulit, banyak penelitian telah mengangkat isu penentuan calon mahasiswa penerima beasiswa. Akurasi rendah yang dihasilkan oleh model dapat meningkatkan biaya pembagian beasiswa dan dapat membuat proses pendistribusian dana beasiswa dengan aktifitas yang tidak sesuai dengan kriteria-kriteria penerima beasiswa. Penentuan penerimaan beasiswa adalah kepentingan besar untuk diterapkan di institusi perguruan tinggi dalam menjalankan kegiatan akademik perguruan tinggi tersebut.

Akurasi hasil penentuan penerima beasiswa digunakan untuk banyak alasan, utamanya adalah untuk kebutuhan bagi perguruan tinggi untuk memberikan kemudahan dan ketersediaan alokasi dana beasiswa pada masing-masing mahasiswa, dan kesempatan untuk membantu ekonomi mahasiswa yang kurang mampu untuk tetap bisa melanjutkan pendidikannya di perguruan tinggi tersebut. Jelasnya, bagaimana bisa secara akurat memperkirakan calon mahasiswa penerima beasiswa sehingga memiliki implikasi yang mendalam dan signifikan bagi perguruan tinggi dan masing-masing calon mahasiswa penerima beasiswa.

Penerimaan beasiswa adalah tahapan proses kegiatan akademik dari perguruan tinggi yang rutin diselenggarakan setiap tahun pelajaran yang baru yang harus diberikan kepada mahasiswa calon penerima beasiswa yang memenuhi kriteria-kriteria yang telah ditentukan oleh perguruan tinggi tersebut sebelumnya. Tujuan pemberian beasiswa adalah merupakan kegiatan rutin yang harus dipenuhi oleh perguruan tinggi setiap tahun ajaran baru dan sebagai motivasi terhadap mahasiswa yang berprestasi untuk lebih meningkatkan lagi prestasinya dengan diberikan beasiswa, serta untuk membantu mahasiswa yang kurang mampu dalam segi finansial sehingga mereka tidak akan terputus ditengah jalan dalam menempuh ilmu dibangku perguruan tinggi dan khususnya untuk menyediakan sarana yang obyektif dan terpercaya bagi perguruan tinggi, mahasiswa dan orang tua mahasiswa lainnya dalam memonitor kegiatan belajar mengajar yang aktif sesuai dengan kurikulumnya.

Berdasarkan latar belakang yang telah diuraikan, dapat dirumuskan bahwa:

- a. Proses penerimaan beasiswa yang baik dapat memberikan tingkat pembelajaran yang lebih efektif.
- b. Akurasi hasil penerima beasiswa digunakan untuk membuat proses pembelajaran lebih baik lagi.
- c. Penerimaan beasiswa adalah tahapan proses kegiatan rutin perguruan tinggi yang harus diberikan kepada calon mahasiswa penerima beasiswa yang aktif.

## **B. METODE**

### **1. Neural Network**

*Neural network* (Han, 2006) adalah satu set unit *input/output* yang terhubung dimana tiap relasinya memiliki bobot. *Neural network* dimaksudkan untuk mensimulasikan perilaku sistem biologi susunan syaraf manusia (alpaydin,2010).

### **2. Algoritma Klasifikasi Data Mining**

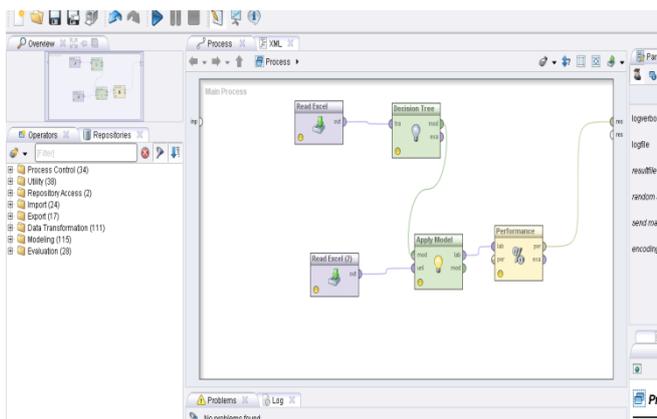
Klasifikasi (Han, 2006) adalah proses penemuan model (atau fungsi) yg menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui.

Klasifikasi data terdiri dari 2 langkah proses. Pertama adalah *learning (fase training)*, dimana algoritma klasifikasi dibuat untuk menganalisa *data training* lalu direpresentasikan dalam bentuk *rule* klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari *rule* klasifikasi. Proses klasifikasi didasarkan pada empat komponen (Gorunescu, 2011):

- a. Kelas  
Variabel dependen yang berupa kategorikal yang merepresentasikan 'label' yg terdapat pada objek. Contohnya: resiko penyakit jantung, resiko kredit, *customer loyalty*, jenis gempa.
- b. Predictor  
Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.
- c. Training Dataset  
Satu set data yang berisi nilai dari kedua komponen diatas yang digunakan untuk menentukan kelas yang cocok berdasarkan *predictor*.
- d. Testing Dataset  
Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

### 3. Rapid Miner 5

*Rapid Miner* adalah aplikasi *data mining* yang tidak perlu dipertanyakan lagi dan berbasis sistem *open-source* dunia yang terkemuka dan ternama. Tersedia sebagai aplikasi yang berdiri sendiri untuk analisis data dan sebagai mesin data mining untuk integrasi ke dalam produk sendiri.



Gambar 1. *Rapid Miner 5* interface

### 4. Evaluasi Model

Evaluasi model merupakan tahapan yang dilakukan untuk memperoleh informasi dari hasil-hasil klasifikasi terhadap algoritma yang digunakan. Ukuran yang digunakan pada *Rapid Miner classifier* adalah: *confusion matrix* yang memberikan informasi untuk rasio keberhasilan klasifikasi. *Confusion matrix* merupakan alat pengukuran berbentuk matrix yang digunakan untuk mendapatkan jumlah ketepatan klasifikasi *dataset* terhadap nilai atribut registrasi dan tidak registrasi pada algoritma yang dipakai.

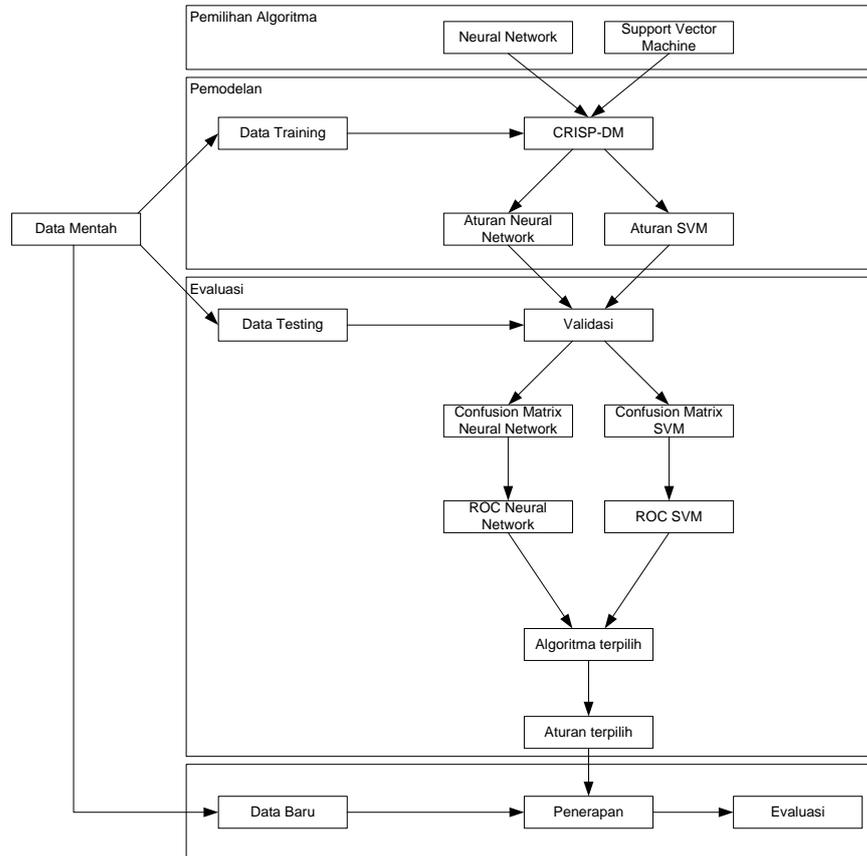
Beberapa kegiatan yang dilakukan dengan menggunakan data hasil klasifikasi dalam *confusion matrix* diantaranya: menghitung nilai rata-rata keberhasilan klasifikasi (*overall success rate*) dan penghitungan persentase kelas positif (*true positive & false positive*) yang diperoleh dalam klasifikasi yang disebut dengan *lift chart*. Besarnya akurasi dan *error rate* dapat dicari dengan persamaan sebagai berikut:

$$\text{Akurasi} = \frac{\text{Banyaknya prediksi yang benar}}{\text{Total Banyaknya prediksi}}$$

$$\text{Error Rate} = \frac{\text{Banyaknya prediksi yang salah}}{\text{Total Banyaknya prediksi}}$$

### 5. Kerangka Pemikiran

Gambar di bawah ini adalah kerangka pemikiran dari penelitian ini, dimulai dari *problem* (permasalahan) analisa data mahasiswa kurang mampu yang tidak akurat kemudian dibuat *approach* (model) yaitu algoritma *neural network* untuk memecahkan permasalahan. Untuk mengembangkan aplikasi (*development*) berdasarkan model yang dibuat, digunakan *rapid miner*. Tahap berikutnya yaitu *implementation* (implementasi), pada tahap ini objek implementasi dilakukan di Universitas Linggabuana PGRI Sukabumi, teknik *sampling* menggunakan *random sample*. Kerangka pemikiran dalam penulisan ini yaitu:



Gambar 2. kerangka pemikiran

## C. HASIL DAN PEMBAHASAN

### 1. Hasil

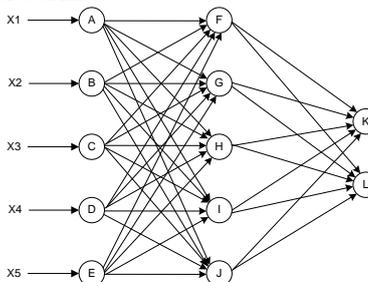
Untuk pemilihan penerima beasiswa, didapat data dari Universitas Linggabuana PGRI Sukabumi sebanyak 409 data mahasiswa, direduksi menjadi 37 data yang terdiri dari 7 atribut. Dimana 6 atribut *predictor* dan 1 atribut hasil. Penelitian ini bertujuan untuk menentukan akurasi kelayakan pemberian beasiswa yang dibandingkan dengan menggunakan metode *Neural Network*. Atribut-atribut yang menjadi parameter terlihat pada tabel 1.

Tabel 1. Atribut dan nilai kategori

No	Atribut	Nilai
1	Kelas	TI.1
		TI.3
		TI.5
		TI.7
		MI.1
		MI.3
		MI.5
		TK.1
		TK.3

		TK.5
		KA.1
		KA.3
		KA.5
2	Status orang tua	Orang tua Yatim
3	Jumlah tanggungan orang tua	<= 3 orang > 3 orang
4	Pekerjaan orang tua	Buruh Wiraswasta Petani Swasta
5	Penghasilan orang tua	< 2.000.000 > 2.000.000
6	Indeks prestasi	2.5 – 3.0 >= 3.0
7	Remark	Terima Tidak

Berdasarkan data *training*, diambil satu *record* secara acak untuk menghitung nilai *output* dengan langkah-langkah sebagai berikut:



Gambar 3. Multi Layer Perceptron

a. Inisiasi bobot awal secara acak

Untuk masukan pada tiap simpul di *input layer* dan *hidden layer*, diberi nilai random antara -1.0 s/d 1.0, begitu pula untuk simpul bias yang terhubung dengan *hidden layer* dan *output layer*.

Diketahui untuk nilai:

$$X_1 = 0, X_2 = 1, X_3 = 1, X_4 = 1, X_5 = 1$$

$$\begin{array}{lllllll}
 W_{AF} = 0.2 & W_{BF} = 0.4 & W_{CF} = -0.5 & W_{DF} = 0 & W_{EF} = 0 & W_{FK} = -0.3 & W_{HL} = 0 \\
 W_{AG} = -0.3 & W_{BG} = 0.1 & W_{CG} = 0.2 & W_{DG} = 0 & W_{EG} = 0 & W_{FL} = 0 & W_{IK} = 0 \\
 W_{AH} = 0 & W_{BH} = 0 & W_{CH} = 0 & W_{DH} = 0 & W_{EH} = 0 & W_{GK} = -0.2 & W_{IL} = 0 \\
 W_{AI} = 0 & W_{BI} = 0 & W_{CI} = 0 & W_{DI} = 0 & W_{EI} = 0 & W_{GL} = 0 & W_{JK} = 0 \\
 W_{AJ} = 0 & W_{BJ} = 0 & W_{CJ} = 0 & W_{DJ} = 0 & W_{EJ} = 0 & W_{HK} = 0 & W_{JL} = 0
 \end{array}$$

b. Hitung Masukan, keluaran dan error

1) Neuron F

$$\begin{aligned}
 \text{Input}_F &= (A.W_{AF}) + (B.W_{BF}) + (C.W_{CF}) + (D.W_{DF}) + (E.W_{EF}) \\
 &= (0. 0.2) + (1. 0.4) + (1. -0.5) + (1. 0) + (1. 0) \\
 &= -0.1
 \end{aligned}$$

$$\text{Output}_F = 1/(1+e^{-0.1}) = 0.475$$

2) Neuron G

$$\begin{aligned}
 \text{Input}_G &= (A.W_{AG}) + (B.W_{BG}) + (C.W_{CG}) + (D.W_{DG}) + (E.W_{EG}) \\
 &= (0. -0.3) + (1. 0.1) + (1. 0.2) + (1. 0) + (1. 0) \\
 &= 0.3
 \end{aligned}$$

$$\text{Output}_G = 1/(1+e^{-0.3}) = 0.574$$

3) Neuron H

$$\text{Input}_H = (A.W_{AH}) + (B.W_{BH}) + (C.W_{CH}) + (D.W_{DH}) + (E.W_{EH})$$

$$= (0.0) + (1.0) + (1.0) + (1.0) + (1.0)$$

$$= 0$$

Output<sub>H</sub> =  $1/(1+e^0) = 0.5$

4) Neuron I

$$\text{Input}_I = (A.W_{AI}) + (B.W_{BI}) + (C.W_{CI}) + (D.W_{DI}) + (E.W_{EI})$$

$$= (0.0) + (1.0) + (1.0) + (1.0) + (1.0)$$

$$= 0$$

Output<sub>I</sub> =  $1/(1+e^0) = 0.5$

5) Neuron J

$$\text{Input}_J = (A.W_{AJ}) + (B.W_{BJ}) + (C.W_{CJ}) + (D.W_{DJ}) + (E.W_{EJ})$$

$$= (0.0) + (1.0) + (1.0) + (1.0) + (1.0)$$

$$= 0$$

Output<sub>J</sub> =  $1/(1+e^0) = 0.5$

6) Neuron K

$$\text{Input}_K = (O_F.W_{FK}) + (O_G.W_{GK}) + (O_H.W_{HK}) + (O_I.W_{IK}) + (O_J.W_{JK})$$

$$= (0.475 \cdot -0.3) + (0.574 \cdot -0.2) + (0.5 \cdot 0) + (0.5 \cdot 0) + (0.5 \cdot 0)$$

$$= -0.257$$

Output<sub>K</sub> =  $1/(1+e^{-(0.257)}) = 0.436$

7) Neuron L

$$\text{Input}_L = (O_F.W_{FL}) + (O_G.W_{GL}) + (O_H.W_{HL}) + (O_I.W_{IL}) + (O_J.W_{JL})$$

$$= (0.475 \cdot 0) + (0.574 \cdot 0) + (0.5 \cdot 0) + (0.5 \cdot 0) + (0.5 \cdot 0)$$

$$= 0$$

Output<sub>L</sub> =  $1/(1+e^0) = 0.5$

8) Hitung nilai *error output* dan *hidden layer*

$$\text{Error}_L = \text{output}_L \cdot (1 - \text{output}_L) \cdot (\text{output}_{\text{target}} - \text{output}_L)$$

$$= 0.5 \cdot (1 - 0.5) \cdot (1 - 0.5)$$

$$= 0.125$$

$$\text{Error}_K = \text{output}_K \cdot (1 - \text{output}_K) \cdot (\text{output}_{\text{target}} - \text{output}_K)$$

$$= 0.436 \cdot (1 - 0.436) \cdot (1 - 0.436)$$

$$= 0.138$$

$$\text{Error}_J = \text{output}_J \cdot (1 - \text{output}_J) \cdot ((\text{error}_L \cdot W_{JL}) + (\text{error}_K \cdot W_{JK}))$$

$$= 0.5 \cdot (1 - 0.5) \cdot ((0.125 \cdot 0) + (0.138 \cdot 0))$$

$$= 0$$

$$\text{Error}_I = \text{output}_I \cdot (1 - \text{output}_I) \cdot ((\text{error}_L \cdot W_{IL}) + (\text{error}_K \cdot W_{IK}))$$

$$= 0.5 \cdot (1 - 0.5) \cdot ((0.125 \cdot 0) + (0.138 \cdot 0))$$

$$= 0$$

$$\text{Error}_H = \text{output}_H \cdot (1 - \text{output}_H) \cdot ((\text{error}_L \cdot W_{HL}) + (\text{error}_K \cdot W_{HK}))$$

$$= 0.5 \cdot (1 - 0.5) \cdot ((0.125 \cdot 0) + (0.138 \cdot 0))$$

$$= 0$$

$$\text{Error}_G = \text{output}_G \cdot (1 - \text{output}_G) \cdot ((\text{error}_L \cdot W_{GL}) + (\text{error}_K \cdot W_{GK}))$$

$$= 0.574 \cdot (1 - 0.574) \cdot ((0.125 \cdot 0) + (0.138 \cdot -0.2))$$

$$= -0.0068$$

$$\text{Error}_F = \text{output}_F \cdot (1 - \text{output}_F) \cdot ((\text{error}_L \cdot W_{FL}) + (\text{error}_K \cdot W_{FK}))$$

$$= 0.475 \cdot (1 - 0.475) \cdot ((0.125 \cdot 0) + (0.138 \cdot -0.3))$$

$$= -0.0069$$

Hasil perhitungan akhir *neural network* sebagai berikut:

Tabel 2. *Hidden layer*

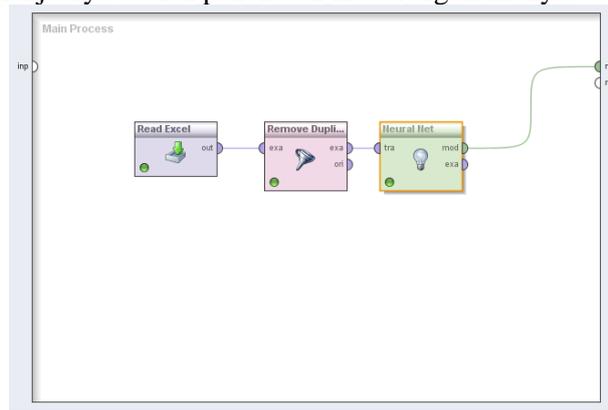
Simpul	Hidden Layer (Sigmoid)				
	1	2	3	4	5
Status	0.813	1.370	1.264	1.257	1.119
Jumlah tanggungan	-0.057	-0.242	-0.214	-0.217	-0.189
Pekerjaan orang tua	-0.153	-0.384	-0.338	-0.329	-0.297
Penghasilan	0.983	1.554	1.446	1.442	1.299
IPK	-2.419	-3.559	-3.291	-3.288	-3.012

Threshold	0.169	0.873	0.722	0.726	0.551
-----------	-------	-------	-------	-------	-------

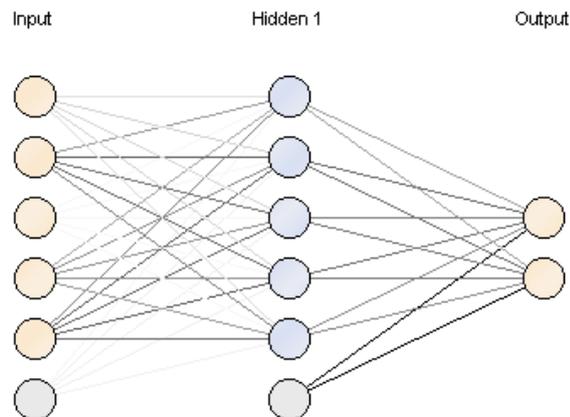
Tabel 3. *Output Layer*

Remark	Output (sigmoid)					Threshold
	1	2	3	4	5	
Terima	-1.582	-3.005	-2.623	-2.678	-2.306	4.614
Tidak	1.608	2.998	2.664	2.617	2.313	-4.615

Pembuatan model algoritma *Neural Network* diawali dengan pembacaan *file data (Read Excel)*. *Data training* disimpan dalam satu *file Excel*. Lalu data di *transform* ke bentuk *numeric*, Langkah selanjutnya adalah penentuan model algoritmanya.



Gambar 4. Desain model *neural network*



Gambar 5. *Neural network* yang dihasilkan

Gambar 5 adalah *neural network* yang dihasilkan dari pengolahan data *training* dengan metode *neural network* adalah *multilayer perceptron* yang dihasilkan dari data *training* pada Tabel III.3. Terdiri dari tiga *layer*, yaitu *Input layer* terdiri dari 6 simpul, sama dengan jumlah atribut *prediktor* ditambah satu simpul bias. Pada pembahasan ini digunakan satu *hidden layer* yang terdiri dari 6 simpul ditambah satu simpul bias. Di bagian *output layer* terdapat dua simpul yang mewakili atribut remark yaitu *terima* dan *tidak*.

Untuk setiap data pada data *training*, dihitung input untuk simpul berdasarkan nilai input dan jaringan saat itu. Bobot awal untuk *input layer*, *hidden layer*, dan bias diinisialisasi secara acak. Simpul bias terdiri dari dua, yaitu pada *input layer* yang terhubung dengan simpul-simpul pada *hidden layer*, dan pada *hidden layer* yang terhubung pada *output layer*. Setelah semua nilai awal diinisialisasi, kemudian dihitung masukan, keluaran, dan *error*. Selanjutnya membangkitkan output untuk simpul menggunakan fungsi aktivasi sigmoid. Setelah didapat nilai dari fungsi aktivasi, hitung nilai *error* antara nilai yang diprediksi dengan nilai yang sesungguhnya. Setelah nilai *error* dihitung, selanjutnya dibalik ke *layer* sebelumnya (*backpropagated*). Nilai *Error* yang dihasilkan dari langkah sebelumnya digunakan untuk

memperbarui bobot relasi. Hasil perhitungan akhir *backpropagation* fungsi aktivasi untuk simpul pada *hidden layer* terdapat pada Gambar 6. berikut ini:

```

Node 1 (Sigmoid)
-----
Status: 0.813
Jumlah Tanggungan: -0.057
Pekerjaan Ortu: -0.153
Penghasilan: 0.983
IPK: -2.419
Threshold: 0.169

Node 2 (Sigmoid)
-----
Status: 1.370
Jumlah Tanggungan: -0.242
Pekerjaan Ortu: -0.384
Penghasilan: 1.554
IPK: -3.559
Threshold: 0.873

Node 3 (Sigmoid)
-----
Status: 1.264
Jumlah Tanggungan: -0.214
Pekerjaan Ortu: -0.338
Penghasilan: 1.446
IPK: -3.291
Threshold: 0.722

Node 4 (Sigmoid)
-----
Status: 1.257
Jumlah Tanggungan: -0.217
Pekerjaan Ortu: -0.329
Penghasilan: 1.442
IPK: -3.288
Threshold: 0.726

Node 5 (Sigmoid)
-----
Status: 1.119
Jumlah Tanggungan: -0.189
Pekerjaan Ortu: -0.297
Penghasilan: 1.299
IPK: -3.012
Threshold: 0.551
    
```

Gambar 6. Nilai bobot akhir untuk *hidden layer* data *training*

Sedangkan gambar 7. merupakan nilai akhir fungsi aktivasi pada *output layer remark*, yaitu atribut *remark* yang dinyatakan dengan simpul pada *output layer*. Node satu sampai dengan lima adalah nilai bias terbaru yang terdapat pada relasi antara simpul pada *hidden layer* dan simpul pada *output layer*.

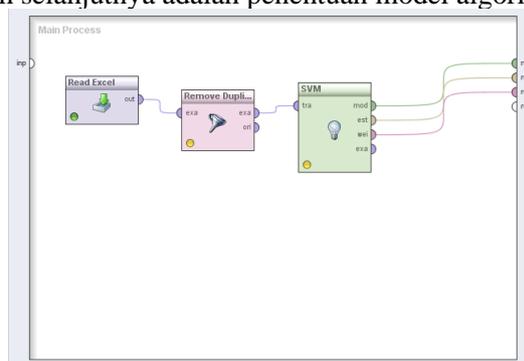
```

Class 'terima' (Sigmoid)
-----
Node 1: -1.582
Node 2: -3.005
Node 3: -2.623
Node 4: -2.678
Node 5: -2.306
Threshold: 4.614

Class 'tidak' (Sigmoid)
-----
Node 1: 1.608
Node 2: 2.998
Node 3: 2.664
Node 4: 2.617
Node 5: 2.313
Threshold: -4.615
    
```

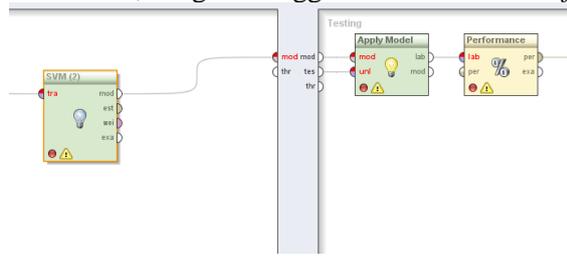
Gambar 7. Nilai bobot akhir untuk *output layer* data *training*

Pembuatan model algoritma *Support Vector Machine* (SVM) diawali dengan pembacaan file data (*Read Excel*). *Data training* disimpan dalam satu file *Excel*. Lalu data di transform ke bentuk numeric, Langkah selanjutnya adalah penentuan model algoritmanya



Gambar 8. Desain model *support vector machine*

Data tersebut kemudian divalidasi, dengan menggunakan *numerical x-falidation*

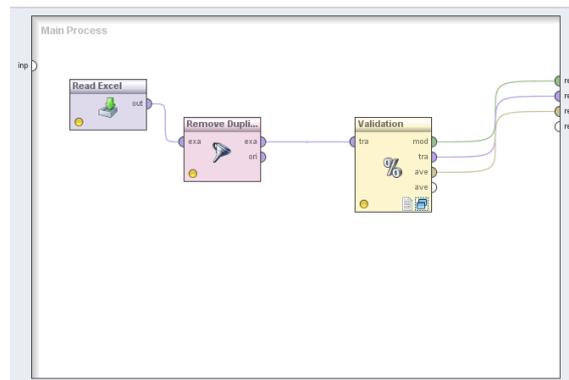


Gambar 9. Model Algoritma SVM

Dalam penulisan ini metode yang digunakan, yaitu algoritma *neural network* dan *support vector machine*, kemudian dilakukan pemilihan keduanya dan mengukur metode mana yang paling akurat. Metode klasifikasi bisa dievaluasi berdasarkan beberapa kriteria seperti tingkat akurasi, kecepatan, kehandalan, skalabilitas.

## 2. Pembahasan

Penelitian ini bertujuan untuk mengukur tingkat akurasi dua algoritma dalam memilih penerima beasiswa, kemudian menganalisa akurasi dengan membandingkan kedua metode tersebut. Adapun tahapan-tahapan dalam evaluasi dan validasi adalah sebagai berikut: Model yang telah dibentuk diuji tingkat akurasinya dengan memasukkan data uji yang berasal dari data *training*. Karena data yang didapat dalam penelitian ini setelah proses *preprocessing* hanya 37 data maka digunakan metode *cross validation* untuk menguji tingkat akurasi. Untuk nilai akurasi model untuk algoritma *neural network* dan algoritma *support vector machine* sebesar 97.50%.



Gambar 10. Desain model Validasi

Selain itu dalam penelitian ini diuji juga dengan menggunakan *confution matrix* dan kurva ROC

Tabel 4 diketahui dari 37 data, 12 diklasifikasikan ya sesuai dengan prediksi yang dilakukan dengan metode *neural network*, kemudian 1 data diprediksi ya tetapi ternyata hasilnya prediksi tidak, 24 data *remark* tidak diprediksi sesuai, dan 0 data diprediksi tidak ternyata hasil prediksinya ya.

Tabel 4 Model *Confusion Matrix Neural Network*

accuracy: 97.50% +/- 7.50% (mikro: 97.30%)			
	true terima	true tidak	class precision
pred. terima	12	1	92.31%
pred. tidak	0	24	100.00%
class recall	100.00%	96.00%	
precision: 100.00% +/- 0.00% (mikro: 100.00%) (positive class: tidak)			
	true terima	true tidak	class precision
pred. terima	12	1	92.31%
pred. tidak	0	24	100.00%
class recall	100.00%	96.00%	
recall: 96.67% +/- 10.00% (mikro: 96.00%) (positive class: tidak)			
	true terima	true tidak	class precision
pred. terima	12	1	92.31%
pred. tidak	0	24	100.00%
class recall	100.00%	96.00%	

Berdasarkan Tabel 4 tersebut menunjukkan bahwa, tingkat akurasi dengan menggunakan algoritma *neural network* adalah sebesar 97.50%.

Tabel 5 diketahui dari 37 data, 12 diklasifikasikan ya sesuai dengan prediksi yang dilakukan dengan metode svm, lalu 1 data diprediksi ya tetapi ternyata hasilnya prediksi tidak, 24 data *class tidak* diprediksi sesuai, dan 0 data diprediksi tidak ternyata hasil prediksinya ya.

Tabel 5. Model *Confusion Matrix Support Vector Machine*

accuracy: 97.50% +/- 7.50% (mikro: 97.30%)			
	true terima	true tidak	class precision
pred. terima	12	1	92.31%
pred. tidak	0	24	100.00%
class recall	100.00%	96.00%	

precision: 100.00% +/- 0.00% (mikro: 100.00%) (positive class: tidak)			
	true terima	true tidak	class precision
pred. terima	12	1	92.31%
pred. tidak	0	24	100.00%
class recall	100.00%	96.00%	

recall: 96.67% +/- 10.00% (mikro: 96.00%) (positive class: tidak)			
	true terima	true tidak	class precision
pred. terima	12	1	92.31%
pred. tidak	0	24	100.00%
class recall	100.00%	96.00%	

Dari dua Tabel *confusion matrix* diatas, selanjutnya dilakukan perhitungan nilai *accuracy*, *precision* dan *recall*. Perbandingan nilai *accuracy*, *precision* dan *recall* yang telah dihitung untuk *neural network* dan *Support Vector Machine* dapat dilihat pada Tabel 6 berikut:

Tabel 6. Nilai *accuracy*, *precision* dan *recall*

	<i>Neural Network</i>	<i>Support Vector Machine</i>
Accuracy	97.50%	97.50%
Precision	100%	100%
Recall	96.67%	96.67%

Tabel 7. Perhitungan Nilai *accuracy*, *precision* dan *recall*

		Predicted	
		Terima	Tidak
Actual	Terima	12 (a)	1 (b)
	Tidak	0 (c)	24 (d)

Dari Tabel diatas dapat dihitung nilai *accuracy*, *precision* dan *recall* sebagai berikut:

$$AC = (a+d) / (a+b+c+d)$$

$$AC = (12+24) / (12+1+0+24)$$

$$AC = 36/37 = 97,29\%$$

$$FP = b / (a+b)$$

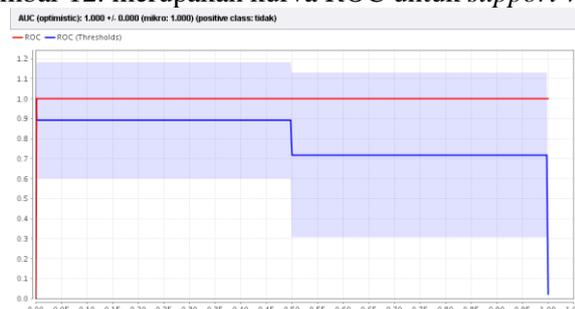
$$FP = 1 / (12+1)$$

merupakan kelas kasus *negative* yang salah yang diklasifikasikan sebagai positif, sehingga untuk meramalkan kasus positif yang di anggap benar adalah:

$$P = d / (b+d) = 24 / 25 = 96.00\%$$

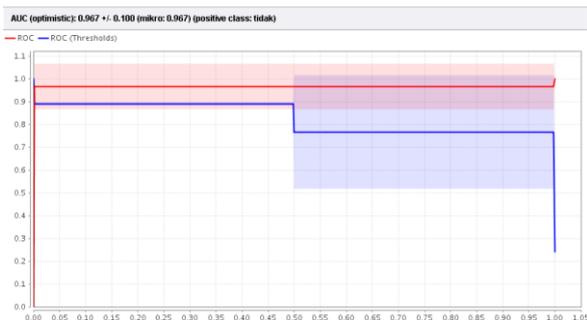
dan kasus positif yang diidentifikasi benar adalah sebesar 1 – 100%.

Hasil perhitungan divisualisasikan dengan kurva ROC. Gambar 11. adalah kurva ROC untuk *neural network* dan gambar 12. merupakan kurva ROC untuk *support vector machine*.



Gambar 11. Kurva ROC *neural network*

Dari Gambar 11. terdapat grafik ROC dengan nilai AUC (Area Under Curve) sebesar 1.000 dimana diagnosa hasilnya *Excellent classification*. Kurva ROC pada gambar III.10 diatas mengekspresikan *confusion matrix* Garis X adalah *false positives* dan garis Y *true positives* sebesar 1.000.



Gambar 12. Kurva ROC *support vector machine*

Dari Gambar 12. terdapat grafik ROC dengan nilai AUC (Area Under Curve) sebesar 0.967 dimana diagnosa hasilnya *Excellent classification*. Kurva ROC pada gambar 12. diatas mengekspresikan *confusion matrix* Garis X adalah *false positives* dan garis Y *true positives* sebesar 0.967.

Perbandingan hasil perhitungan nilai AUC untuk *neural network* dan *support vector machine* dapat dilihat pada Tabel 8.

Tabel 8. Perbandingan Nilai AUC

	<i>Neural Network</i>	<i>Support Vector Machine</i>
AUC	1.000	0.967

Model yang dihasilkan dari data *training* dan data uji validasi dengan menggunakan *neural network* di uji menggunakan *cross validation*, terlihat nilai *accuracy*, *precision*, dan *recall* pada Tabel 9, memiliki nilai *accuracy*, *precision* dan *recall* yang tinggi.

Tabel 9. Nilai *Accuracy* dan AUC

	<i>Neural Network</i>	<i>SVM</i>
Accuracy	97.50%	97.50 %
AUC	1.000	0.967

Dari tabel diatas terlihat bahwa nilai *accuracy* dan AUC dari *neural network* dan *support vector machine* termasuk klasifikasi yang sangat baik. Untuk klasifikasi data *mining*, nilai AUC dapat dibagi menjadi beberapa kelompok.

- 0.90-1.00 = klasifikasi sangat baik
- 0.80-0.89 = klasifikasi baik
- 0.70-0.79 = klasifikasi cukup
- 0.60-0.69 = klasifikasi buruk
- 0.50-0.59 = klasifikasi salah

Berdasarkan hasil diatas dan Tabel 9 maka dapat disimpulkan bahwa metode *neural network* termasuk klasifikasi sangat baik karena memiliki nilai AUC antara 0.90-1.00.

#### D. KESIMPULAN

Dari uraian yang telah dijelaskan sebelumnya, dapat disimpulkan hal-hal sebagai berikut. Dengan menggunakan klasifikasi *data mining*, algoritma yang berbasis jaringan syaraf tiruan (*artificial neural network*), yaitu: algoritma *Multilayer Perceptron* (MLP) dapat diterapkan untuk menentukan beasiswa di Universitas Linggabuana PGRI Sukabumi. Penerapan algoritma *Multilayer perceptron* (MLP) menghasilkan keputusan yang akurat dibandingkan dengan *Support vector Machine* (SVM) dalam pemilihan penerima beasiswa bagi para mahasiswa.

## E. DAFTAR PUSTAKA

- [1] Bramer, Max, *Principles of Data Mining*, Springer, London, 2007.
- [2] Gorunescu, *Data Mining: Concepts, Models, and Techniques*, Verlag Berlin Heidelberg: Springer, 2011.
- [3] Han, Jiawei and Kamber, M. *Data Mining: Concepts and Techniques*, University of Illinois at Urbana-Champaign, 2007.
- [4] Jamhur, H. (2020). Pemodelan Prediksi Predikat Kelulusan Mahasiswa Menggunakan Fuzzy C-Means Berbasis Particle Swarm Optimization. *TeknoIS : Jurnal Ilmiah Teknologi Informasi dan Sains*, 10(1), 13-24. doi:<https://doi.org/10.36350/jbs.v10i1.79>
- [5] Kusriani dan Luthfi, E.T, *Algoritma Data Mining*, Andi Publishing, Yogyakarta, 2008.
- [6] Larose, D. T., *Discovering Knowledge in Data*, New Jersey: John Wiley & Sons, Inc, 2005.
- [7] Larose, D.T., *Data Mining Methods and Models - 2007*: John Wiley & Sons, Inc, 2007.
- [8] Larose, D.T and Markov, Zdravkov, *Uncovering Patterns in Web Content, Structure and Usage*: John Wiley & Sons, Inc, 2007.
- [9] Maimon, Oded and Rokach, Lior, *Data Mining and Knowledge Discovery Handbook*, Springer, York, 2005.
- [10] Marimin, *Teknik dan Aplikasi Pengambilan Keputusan Kriteria Majemuk*, PT. Grasindo, Jakarta, 2008.
- [11] Myatt, Glenn J *Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining*, John Wiley & Sons, Inc., New Jersey, 2007.
- [12] Santoso, Budi, *Data Mining Terapan dengan Matlab*, PT. Graha Ilmu, Surabaya, 2007.
- [13] Vercellis, *The Business Intelligence*, Politecnico di milano, Italy, 2009.
- [14] Wu, Xindong & Kumar, Vipin, *The Top Ten Algorithms in Data Mining*, CRC Press, Boca Raton, 2009.