

**PERBANDINGAN MODEL REGRESI LOGISTIK DAN MODEL  
RADIAL BASIS FUNCTION NEURAL NETWORK UNTUK  
PENGKLASIFIKASIAN PEUBAH RESPON BINER**

**SKRIPSI**

**Sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains  
dalam bidang Statistika**

oleh :

**NUR FITRIYAH ARIF**

**0910953039-95**



**PROGRAM STUDI STATISTIKA  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2013**

**LEMBAR PENGESAHAN SKRIPSI**

**PERBANDINGAN MODEL REGRESI LOGISTIK DAN MODEL  
RADIAL BASIS FUNCTION NEURAL NETWORK UNTUK  
PENGKLASIFIKASIAN PEUBAH RESPON BINER**

oleh :

**Nur Fitriyah Arif**  
**0910953039-95**

**Setelah dipertahankan di depan Majelis Penguji  
pada tanggal 11 Desember 2013  
dan dinyatakan memenuhi syarat untuk memperoleh gelar  
Sarjana Sains dalam bidang Statistika**

**Pembimbing I**

**Pembimbing II**

**Samingun Handoyo, S.Si., M.Cs**  
**NIP. 197304151998021002**

**Prof. Dr. Ir. Henny Pramoedyo, MS**  
**NIP. 195707051981031009**

**Mengetahui,**  
**Ketua Jurusan Matematika**  
**Fakultas MIPA Universitas Brawijaya**

**Dr. Abdul Rouf Alghofari, M.Sc**  
**NIP. 196709071992031001**

## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Nur Fitriyah Arif  
NIM : 0910953039  
Jurusan : Statistika  
Penulisan Skripsi berjudul : **PERBANDINGAN MODEL  
REGRESI LOGISTIK DAN MODEL *RADIAL BASIS FUNCTION*  
*NEURAL NETWORK* UNTUK PENGKLASIFIKASIAN PEUBAH  
RESPON BINER**

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya saya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di dalam daftar pustaka dalam Skripsi ini.
2. Apabila di kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala risiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 11 Desember 2013  
Yang menyatakan,

(Nur Fitriyah Arif)  
NIM. 0910953039

# PERBANDINGAN MODEL REGRESI LOGISTIK DAN MODEL RADIAL BASIS FUNCTION NEURAL NETWORK UNTUK PENGKLASIFIKASIAN PEUBAH RESPON BINER

## ABSTRAK

Regresi logistik merupakan salah satu metode regresi di mana peubah responnya mempunyai sifat kualitatif. Pada regresi logistik tidak diperlukan asumsi linieritas, normalitas sisaan, autokorelasi, dan homokedastisitas, tetapi tidak boleh ada multikolinieritas antar peubah bebas. *Radial Basis Function Neural Network* (RBFNN) merupakan salah satu bentuk jaringan multilayer *perceptron* (MLP) yang dapat digunakan untuk pengklasifikasian obyek. Desain dari RBFNN adalah model *neural network* yang mentransformasi input secara non linier dengan menggunakan fungsi aktivasi *Gaussian* pada lapisan tersembunyi sebelum diproses secara linier pada lapisan *output*. Tujuan penelitian ini adalah membandingkan model regresi logistik dengan RBFNN untuk pengklasifikasian peubah respon biner ditinjau dari nilai APER dan SSE. Pada penelitian ini digunakan 3 jenis data yang berbeda. Pada masing-masing data diaplikasikan metode regresi logistik dan RBFNN. Nilai APER untuk model regresi logistik adalah 14,6%; 22,9%; dan 35,5% berturut-turut untuk data 1, data 2, dan data 3 sedangkan pada model RBFNN adalah 9,7%; 20,8%; dan 31,1% berturut-turut untuk ketiga jenis data tersebut. Adapun nilai SSE untuk model regresi logistik adalah 7,434; 7,538; dan 9,995 sedangkan untuk model RBFNN adalah 7,226; 7,303; dan 9,173 berturut-turut untuk data 1, data 2, dan data 3, sehingga berdasarkan hasil tersebut dapat disimpulkan bahwa model RBFNN dapat mengklasifikasi peubah respon biner lebih baik daripada model regresi logistik.

*Kata Kunci : Regresi Logistik, RBFNN, Klasifikasi, Respon Biner.*

# COMPARISON OF LOGISTIC REGRESSION MODEL AND RADIAL BASIS FUNCTION NEURAL NETWORK MODEL FOR CLASSIFICATION OF BINARY RESPONSE VARIABLE

## ABSTRACT

Logistic regression is a regression method where the response variable has a qualitative. Logistic regression do not need to be linearity assumed, normality, autocorrelate, and homokedastisitas, but there can be no multicollinearity among independent variables. Radial Basis Function Neural Network (RBFNN) is one form of network of multilayer perceptron (MLP) that can be used for object classification. Design of RBFNN is a model that transforms the nonlinear input using Gaussian activation function in the hidden layer before it is processed linearly in the output layer. The purpose of this study was to compare logistic regression model with RBFNN for classification of binary response variables based of APER and SSE value. This research used 3 cases of different data. On each of the data applied logistic regression method and RBFNN. The value of APER for logistic regression model 14,6%; 22,9%; and 35,5% consecutive to data 1, data 2, and data 3 whereas the RBFNN models are 9,7%; 20,8%; and 31,1%. SSE values in logistic regression model are 7,434; 7,538; and 9,995 while for RBFNN models are 7,226; 7,303; and 9,173 consecutive for data 1, data 2, and data 3, so from this results can be concluded that the RBFNN models are better in classifying binary response variable than the logistic regression model.

*Keywords: Logistic Regression, RBFNN, Classification, Binary Response.*



## KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Skripsi dengan judul **PERBANDINGAN MODEL REGRESI LOGISTIK DAN MODEL RADIAL BASIS FUNCTION NEURAL NETWORK UNTUK PENGKLASIFIKASIAN PEUBAH RESPON BINER.**

Selama penyusunan Skripsi ini, penulis mendapatkan banyak bantuan, dukungan, semangat, bimbingan serta motivasi dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada :

1. Bapak Samingun Handoyo, S.Si., M.Cs selaku Dosen Pembimbing I dan Bapak Prof. Dr. Ir. Henny Pramodyo, MS selaku Dosen Pembimbing II atas waktu, bimbingan dan saran dalam penyusunan Skripsi.
2. Ibu Dr. Suci Astutik, S.Si., M.Si selaku Dosen Penguji atas waktu, bimbingan, saran dan masukan yang telah diberikan.
3. Kedua orang tua yang selalu memberikan doa, semangat dan motivasi.
4. Teman-teman Prodi Statistika 2009 atas kebersamaan, bantuan, dukungan dan semangat selama ini.
5. Staf Jurusan Matematika atas bantuan yang telah diberikan.

Penulis menyadari bahwa penulisan Skripsi ini masih kurang sempurna. Oleh sebab itu, saran dan kritik yang membangun sangat penulis harapkan untuk penulisan yang lebih baik. Semoga hasil penyusunan skripsi ini dapat memberikan manfaat bagi pembaca.

Malang, Desember 2013

Penulis

# DAFTAR ISI

Halaman

<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PENGESAHAN</b> .....	ii
<b>HALAMAN PERNYATAAN</b> .....	iii
<b>ABSTRAK</b> .....	iv
<b>ABSTRACT</b> .....	v
<b>KATA PENGANTAR</b> .....	vi
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR TABEL</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR LAMPIRAN</b> .....	xi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Analisis Regresi.....	5
2.2 Analisis Regresi Logistik.....	9
2.2.1 Multikolinieritas.....	11
2.2.2 Pendugaan Parameter.....	11
2.2.3 Pengujian Terhadap Pendugaan Parameter.....	15
2.2.4 Uji Kesesuaian Model.....	17
2.3 Jaringan Syaraf Tiruan.....	17
2.3.1 Sejarah.....	18
2.3.2 Konsep Dasar Pemodelan JST.....	21
2.3.3 Arsitektur JST.....	21
2.4 Metode Pelatihan JST.....	22
2.4.1 Proses Pembelajaran JST.....	22
2.4.2 Fungsi Aktivasi JST.....	23
2.5 RBFNN.....	25
2.5.1 Fungsi Basis Radial.....	26

2.5.2	Optimalisasi Vektor Pembobot.....	27
2.5.3	Metode Pelatihan RBFNN.....	28
2.5.4	Metode Pelatihan RBFNN di Maltab.....	30
2.6	Prosedur Klasifikasi.....	32
2.7	Perbandingan Algoritma Klasifikasi.....	33

### **BAB III METODOLOGI**

3.1	Sumber Data.....	35
3.2	Metode Penelitian.....	35

### **BAB IV HASIL DAN PEMBAHASAN**

4.1	Analisis Regresi Logistik Biner.....	41
4.1.1	Asumsi Multikolinieritas.....	41
4.1.2	Pengujian secara Serentak.....	42
4.1.3	Pengujian secara Individu/Parsial.....	43
4.1.4	Pembentukan Model untuk Variabel Signifikan.....	45
4.1.5	Pengujian Kelayakan Model.....	46
4.1.6	Ketepatan Klasifikasi.....	47
4.2	Jaringan Syaraf Tiruan (ANN).....	49
4.2.1	Model RBFNN.....	49
4.2.2	Grafik Target dan <i>Output</i> .....	51
4.2.3	Pengukuran Ketepatan Klasifikasi.....	53
4.3	Pemilihan Model Terbaik.....	55

### **BAB V KESIMPULAN DAN SARAN**

5.1	Kesimpulan.....	57
5.2	Saran.....	57

<b>DAFTAR PUSTAKA</b> .....	59
-----------------------------	----

<b>LAMPIRAN</b> .....	63
-----------------------	----



## DAFTAR TABEL

	Halaman
Tabel 2.1 Tabel Ketepatan Klasifikasi Johnson dan Wichern (1992).....	32 35
Tabel 3.1 Sumber Data.....	41
Tabel 4.1 Hasil Korelasi Rank Spearman pada Data 1.....	41
Tabel 4.2 Hasil Korelasi Rank Spearman pada Data 2.....	42
Tabel 4.3 Hasil Korelasi Rank Spearman pada Data 3.....	42
Tabel 4.4 Hasil Uji Wald pada Data 1.....	43
Tabel 4.5 Hasil Uji Wald pada Data 2.....	44
Tabel 4.6 Hasil Uji Wald pada Data 3.....	44
Tabel 4.7 Taksiran Model Signifikan pada Data 1.....	45
Tabel 4.8 Taksiran Model Signifikan pada Data 2.....	46
Tabel 4.9 Taksiran Model Signifikan pada Data 3.....	46
Tabel 4.10 Pengujian Hosmer-Lemeshow pada Data.....	46
Tabel 4.11 Tabel Ketepatan Klasifikasi Data <i>Training</i> pada Data 1.....	47
Tabel 4.12 Tabel Ketepatan Klasifikasi Data <i>Testing</i> pada Data 1.....	48
Tabel 4.13 Tabel Ketepatan Klasifikasi Data <i>Training</i> pada Data 2.....	48
Tabel 4.14 Tabel Ketepatan Klasifikasi Data <i>Testing</i> pada Data 2.....	48
Tabel 4.15 Tabel Ketepatan Klasifikasi Data <i>Training</i> pada Data 3.....	49
Tabel 4.16 Tabel Ketepatan Klasifikasi Data <i>Testing</i> pada Data 3.....	49
Tabel 4.17 Daftar Jumlah Neuron Lapisan Tersembunyi dan Nilai SSE.....	50
Tabel 4.18 Ketepatan Klasifikasi Data 1 <i>Training</i> dan <i>Testing</i> .....	54
Tabel 4.19 Ketepatan Klasifikasi Data 2 <i>Training</i> dan <i>Testing</i> .....	54
Tabel 4.20 Ketepatan Klasifikasi Data 3 <i>Training</i> dan <i>Testing</i> .....	55
Tabel 4.21 Kesalahan Klasifikasi (APER) dan SSE <i>Testing</i> .....	55

## DAFTAR GAMBAR

Halaman

Gambar 2.1	Struktur Dasar JST dan Struktur Sederhana Sebuah Neuron.....	20
Gambar 2.2	Model Tiruan Sebuah Neuron.....	21
Gambar 2.3	Jaringan Syaraf dengan Banyak Lapisan.....	22
Gambar 2.4	Fungsi Unit Step.....	23
Gambar 2.5	Fungsi <i>Sign</i> .....	24
Gambar 2.6	Fungsi <i>Sigmoid</i> .....	24
Gambar 2.7	Fungsi <i>Sigmoid Unipolar</i> .....	24
Gambar 2.8	Fungsi <i>Sigmoid Bipolar</i> .....	24
Gambar 2.9	Arsitektur RBFNN.....	26
Gambar 2.10	Arsitektur RBFNN di Matlab.....	30
Gambar 3.1	Diagram Alir Penelitian.....	38
Gambar 3.2	Diagram Alir Regresi Logistik.....	39
Gambar 3.3	Diagram Alir RBFNN.....	40
Gambar 4.1	Grafik Hasil Pengujian Data 1 <i>Training</i> .....	51
Gambar 4.2	Grafik Hasil Pengujian Data 1 <i>Testing</i> .....	51
Gambar 4.3	Grafik Hasil Pengujian Data 2 <i>Training</i> .....	52
Gambar 4.4	Grafik Hasil Pengujian Data 2 <i>Testing</i> .....	52
Gambar 4.5	Grafik Hasil Pengujian Data 3 <i>Training</i> .....	53
Gambar 4.6	Grafik Hasil Pengujian Data 3 <i>Testing</i> .....	53

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Data 1.....	63
Lampiran 2. Data 2.....	65
Lampiran 3. Data 3.....	67
Lampiran 4. <i>Output</i> SPSS Analisis Regresi Logistik Biner.....	69
Lampiran 5. <i>Output</i> MATLAB Model RBFNN.....	77
Lampiran 6. Hasil Prediksi Klasifikasi Data 1 dengan Model RegLog Biner dan RBFNN.....	79
Lampiran 7. Hasil Prediksi Klasifikasi Data 2 dengan Model RegLog Biner dan RBFNN.....	83
Lampiran 8. Hasil Prediksi Klasifikasi Data 3 dengan Model RegLog Biner dan RBFNN.....	87
Lampiran 9. <i>Syntax RBFNN</i> .....	91



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Regresi logistik yaitu metode regresi untuk mencari hubungan antara peubah respon (*dependent*) yang bersifat kategorik dengan sekumpulan peubah bebas (*independent*) yang bersifat kategorik atau kontinyu. Pada regresi logistik tidak memerlukan asumsi kenormalan baik pada sebaran galat maupun pada peubah respon, dan hubungan antara peubah respon dengan peubah bebas tidak perlu diasumsikan linier. Namun terdapat asumsi yang diberlakukan yaitu ketiadaan multikolinieritas antar peubah bebas (Hosmer dan Lemeshow, 1989).

Jaringan syaraf tiruan (JST) atau *Artificial Neural Network* (ANN) adalah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi (Siang, 2005). Jaringan syaraf tiruan (JST) telah banyak dan berhasil dilakukan oleh para ahli dibidangnya dalam berbagai aplikasi, metode pelatihan, dan model JST. Salah satunya adalah metode pelatihan hibrida. Semakin tinggi tingkat ketidaklinieran suatu sistem, seringkali sulit diselesaikan dengan metode pelatihan terbimbing dan tak terbimbing. Banyak peneliti mencoba menggabungkannya dan diperoleh hasil yang lebih baik dengan salah satu metode hibrida yaitu metode *Radial Basis Function Neural Network* (RBFNN).

RBFNN merupakan salah satu jaringan syaraf yang dikenal dalam permasalahan regresi dan klasifikasi data (Du dan Swamy, 2006). Tahun 1996 oleh Sanchez menggunakan *Neural Network* untuk aplikasi klasifikasi dan pemodelan. Pada penelitian Sanchez dikaji klasifikasi dari dua metode JST yaitu *Multi Layer Feed Forward* (MLF) dan RBFNN mengenai kemampuan pemodelan, stabilitas dan reproduktifitas jaringan berdasarkan berbagai *training* pada set data yang sama dengan yang telah ditetapkan nilai untuk kepekaan dan kekhususan. Hasil kedua metode cukup baik untuk mereproduksi ruang *input* namun, untuk jaringan RBFNN karena pendekatan lokal, studi menunjukkan beberapa *property* terkait dengan kepekaan dan kekhususan yang relevan dalam masalah-masalah praktis. Pada tahun 2003 oleh Gupta, Jin, dan Homma memperkenalkan model baru dari *Feed forward* Lapis Banyak atau *Multilayer Feed forward Neural Network* (MFNN) yaitu jaringan syaraf Fungsi Basis Radial atau *Radial*

*Basis Function Neural Network* (RBFNN). Jaringan RBFNN memiliki struktur tiga lapis dan dalam proses pelatihannya lebih cepat dibanding *Perceptron* Lapis Banyak atau *Multilayer Perceptron* (MLP).

Pada tahun 2005, Sarimveis, Doganis, dan Alexandridis memperkenalkan teknik klasifikasi dengan menggunakan metode RBFNN dan *Feed forward Neural Network* (FNN). Kedua metode ini diaplikasikan pada kasus tingkat kelembutan *tissue* dan kedua hasil *training* dari kedua metode dibandingkan. Pada penelitian Sarimveis diperoleh hasil bahwa metode RBFNN lebih akurat jika dibandingkan dengan FNN. Beberapa algoritma klasifikasi dengan prediksi yang cukup signifikan adalah Jaringan syaraf tiruan (JST), C5.0, *Classification via Regression*, dan *LogitBoost*. Jaringan syaraf tiruan (JST) terdiri dari 6 metode, yaitu *Dynamic*, *Prune*, *Exhaustive Prune*, *Quick*, *RBFN* dan *Multiple*. Pada penelitian ini ingin dibandingkan metode regresi logistik dan metode RBFNN untuk pengklasifikasian peubah respon biner (Saravanan and Malika, 2009).

## 1.2. Rumusan Masalah

Rumusan masalah penelitian ini adalah bagaimana perbandingan antara model regresi logistik dan model arsitektur RBFNN untuk pengklasifikasian peubah respon biner berdasarkan nilai APER dan SSE data *testing*?

## 1.3. Batasan Masalah

Batasan masalah penelitian ini adalah :

1. Pada regresi logistik metode pendugaan parameter yang digunakan adalah metode kemungkinan maksimum (*Maximum Likelihood Method*).
2. Fungsi aktivasi pada model RBFNN menggunakan fungsi aktivasi *Gaussian* dan proses pembentukan bobot lapisan *output* pada model RBFNN (*Radial Basis Function Neural Network*) menggunakan metode OLS (*Ordinary Least Square*).

## 1.4. Tujuan Penelitian

Tujuan penelitian ini adalah mengetahui hasil perbandingan antara model regresi logistik dan model RBFNN untuk pengklasifikasian peubah respon biner berdasarkan nilai APER dan SSE data *testing*.



### 1.5. Manfaat Penelitian

Penelitian ini dapat digunakan untuk mengetahui ketepatan pengklasifikasian dari metode regresi logistik dan metode RBFNN pada peubah respon biner sehingga dapat diketahui model yang lebih baik.

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



## BAB II TINJAUAN PUSTAKA

### 2.1 Analisis Regresi

Istilah regresi oleh Francis Galton tahun 1886, “*Family Likeness in Stature*”. Pergeseran istilah regresi yaitu suatu analisis tentang ketergantungan suatu peubah kepada peubahlain dalam rangka membuat estimasi atau prediksi dari nilai rata-rata peubahbebas dengan nilai peubah respon yang diketahui. Analisis regresi memiliki 3 kegunaan yaitu deskripsi, kendali, dan prediksi (peramalan).

Pada analisis regresi dikenal dua macam variabel atau peubah yaitu peubah bebas X (*independent variable*) dan peubah respon Y (*dependent variable*). Peubah bebas adalah suatu peubah yang nilainya telah diketahui, sedangkan peubah respon adalah peubah yang nilainya belum diketahui dan yang akan diramalkan. Regresi dan korelasi adalah metode yang dipakai untuk mengukur hubungan antara dua peubah atau lebih. Korelasi dan regresi berhubungan sangat erat karena setiap regresi pasti didahului dengan korelasi. Korelasi dapat dilanjutkan pada analisis regresi apabila antar peubahnya memiliki hubungan kausal atau fungsional. Penetapan ada hubungan atau tidak harus didasarkan pada teori atau konsep tentang dua peubah. Analisis regresi dalam mencari hubungan peubah bebas dan peubah respon tergantung pada tipe dari peubah respon yang nilai duganya berdasarkan peubah bebas. Jika peubah respon merupakan data kontinyu maka dapat menggunakan regresi linier maupun non linier, sedangkan jika peubah respon merupakan data kategorikal maka dapat menggunakan analisis regresi logistik (Gujarati, 1991). Regresi digolongkan menjadi beberapa macam yaitu :

#### 1. Regresi Linier

Regresi linier adalah metode statistika yang digunakan untuk membentuk model atau hubungan antara satu atau lebih peubah bebas dengan sebuah peubah respon. Regresi linier ini dibedakan menjadi :

a. Regresi linier sederhana dengan model :

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad (2.1)$$

dengan

$Y$  = peubah respon

$\beta_0$  = intersep

$\beta_1$  = slope (koefisien regresi ke-1)

$X$  = peubah bebas

$\varepsilon$  = error

Pada persamaan (2.1) nilai  $\varepsilon$  sulit diketahui karena nilai  $\varepsilon$  berubah untuk setiap amatan  $Y$  namun  $\beta_0$  dan  $\beta_1$  selalu tetap. Sehingga informasi di dalam data untuk menghasilkan contoh nilai dugaan atau estimasi  $b_0$  dan  $b_1$  dapat digunakan bagi  $\beta_0$  dan  $\beta_1$  berturut-turut.

$$\hat{Y} = b_0 + b_1X \quad (2.2)$$

$\hat{Y}$  melambangkan nilai ramalan untuk suatu  $X$  tertentu jika  $b_0$  dan  $b_1$  telah ditentukan. Pendugaan terhadap nilai parameter menggunakan metode kuadrat terkecil yaitu dengan meminimumkan jumlah kuadrat galat seperti :

$$JKG = \sum_{m=1}^p \varepsilon_m^2 = \sum_{m=1}^p (Y_m - \beta_0 - \beta_1 X_m)^2 \quad (2.3)$$

Untuk mendapatkan nilai  $b_0$  dan  $b_1$  maka perlu mendiferensialkan persamaan (2.3) terhadap  $\beta_0$  dan  $\beta_1$  kemudian mensubstitusikannya dengan menyamakan hasil diferensial tersebut menjadi sama dengan nol. Sehingga didapatkan nilai duga  $b_0$  dan  $b_1$  seperti :

$$b_0 = \bar{Y} - b_1 \bar{X} \quad (2.4)$$

$$b_1 = \frac{n \sum X_m Y_m - (\sum X_m)(\sum Y_m)}{n \sum X_m^2 - (\sum X_m)^2} \quad (2.5)$$

b. Regresi linier berganda dengan model :

$$Y_i = \beta_0 + \beta_1 X_{m1} + \dots + \beta_j X_{mj} + \beta_k X_{mk} + \varepsilon_m$$

dari fungsi regresi linier sederhana dan regresi linier berganda masing-masing berbentuk garis lurus (linier sederhana) dan bidang datar (linier berganda).

## 2. Regresi non linier

Regresi non linier adalah hubungan di mana peubah bebas  $X$  dan peubah respon  $Y$  tidak linier. Tidak linier berarti bahwa laju perubahan peubah respon  $Y$  akibat laju perubahan peubah bebas  $X$  tidak konstan untuk nilai-nilai peubah bebas  $X$  tertentu. Regresi non linier dapat dibedakan menjadi dua yaitu :

a. Non linier pada peubah

$$Y = \beta_0 + \beta_1 X_1^2 + \varepsilon$$

b. Non linier pada parameter

$$Y = \beta_0 + \beta_1^2 X_1 + \varepsilon$$

Regresi non linier parameter di bagi menjadi dua yaitu :

a. Model non linier non intrinsik

Model yang berbentuk non linier pada parameter namun bentuk tersebut dapat diubah ke bentuk linier dengan melakukan transformasi yang tepat.

$$Y = ae^{\beta_0 + \beta_1 X^2} + \varepsilon \quad (2.6)$$

persamaan (2.6) diubah menjadi bentuk linier dengan cara transformasi logaritma dengan basis  $e$  sehingga menjadi :

$$e_{\log Y} = e_{\log (ae^{\beta_0 + \beta_1 X^2 + \varepsilon})}$$

$$e_{\log Y} = e_{\log a} + \beta_0 + \beta_1 X^2 + \varepsilon$$

b. Model non linier intrinsik

Jika suatu model non linier intrinsik maka model tersebut tidak dapat diubah menjadi bentuk baku.

$$Y = \frac{\beta_0}{\beta_0 - \beta_1} [e^{-\beta_0 X} - e^{\beta_1 X}] + \varepsilon$$

Model-model lebih rumit :

a. Regresi polinomial yaitu regresi dengan sebuah peubah bebas sebagai faktor dengan pangkat terurut. Bentuk-bentuk model polinom adalah sebagai berikut :

$$Y = \beta_0 + \beta_1 X + \beta_{11} X^2 + \varepsilon \text{ (model ordo-kedua)}$$

$$Y = \beta_0 + \beta_1 X + \beta_{11} X^2 + \beta_{111} X^3 + \varepsilon$$

(model ordo-ketiga)

dan seterusnya.

b. Regresi hiperbola (model transformasi resiprokal)

Pada regresi hiperbola, di mana peubah bebas  $X$  atau peubah respon  $Y$  dapat berfungsi sebagai penyebut sehingga regresi ini disebut regresi dengan fungsi pecahan.

Regresi ini mempunyai bentuk model :

$$Y = \beta_0 + \beta_1 \left(\frac{1}{X_1}\right) + \beta_2 \left(\frac{1}{X_2}\right) + \varepsilon$$

c. Regresi fungsi perpangkatan atau geometrik

Pada regresi ini mempunyai bentuk fungsi yang berbeda dengan fungsi polinomial maupun fungsi eksponensial.

Regresi ini mempunyai bentuk fungsi :

$$Y = \beta_0 + \beta_1^X + \varepsilon$$

d. Regresi eksponensial

Regresi eksponensial adalah regresi di mana peubah bebas  $X$  berfungsi sebagai pangkat atau eksponen. Bentuk fungsi regresi ini adalah :



$$Y = \beta_0 e^{\beta_1 X} + \varepsilon \text{ atau } Y = \beta_0 10^{\beta_1 X} + \varepsilon$$

Modifikasi dari bentuk di atas adalah :

$\frac{1}{Y} = \beta_0 + \beta_1 e^X + \varepsilon$ , ini disebut model umum dari model pertumbuhan.

Modifikasinya juga seperti model transformasi logaritma resiprokal berikut :

$$Y = e^{(\beta_0 + \beta_1/X)} + \varepsilon$$

e. Regresi logaritma

Regresi logaritma adalah di mana peubah respon Y berfungsi sebagai pangkat (eksponen) dan peubah bebas X mempunyai bentuk perpangkatan.

Model regresi ini adalah :

$$Y = \beta_0 + \beta_1 \ln X_1 + \beta_2 \ln X_2 + \varepsilon$$

f. Regresi fungsi geometri

Model regresi geometri hampir sama dengan model eksponensial karena dapat dikembalikan pada model linier dengan jalan melakukan pengambilan logaritma pada persamaannya. Persamaan garis geometri :

$$Y = \beta_0 X^{\beta_1} + \varepsilon$$

jika diambil logaritmanya, maka akan menjadi :

$$Y = \log \beta_0 + \beta_1 \log X + \varepsilon$$

3. Regresi *dummy*

Hubungan antara peubah respon (data kuantitatif) dengan peubah bebas (data kualitatif) di mana terdapat suatu faktor yang mempunyai dua atau lebih taraf yang berbeda pada peubah bebas bahwa masing-masing faktor mempengaruhi deterministik yang terpisah dan berbeda terhadap peubah respon. Peubah ini disebut peubah boneka (*dummy*). Umumnya peubah ini tidak berkaitan dengan tingkatan-tingkatan fisik yang mungkin ada dalam faktor itu sendiri. Model regresi *dummy*:

$$Y = \beta_0 + \beta_1 Z_1 + \varepsilon$$

dengan  $Z_1$  adalah peubah *dummy*.

(Draper dan Smith, 1992).

4. Regresi logistik

Metode regresi untuk mengetahui hubungan peubah respon yang berkategori dengan satu atau lebih peubah bebas

berkategori dan atau kontinyu. Analisis regresi logistik selengkapnya dibahas pada Sub bab 2.2.

## 2.2 Analisis Regresi Logistik

Model regresi logistik merupakan salah satu metode regresi yang digunakan untuk mengetahui hubungan peubah respon yang berkategori dengan satu atau lebih peubah bebas yang bersifat kategorial atau lebih peubah bebas yang bersifat kategorial dan kontinyu. Peubah respon yang digunakan dalam model regresi logistik, yaitu peubah respon biner (*dikotomus*) jika banyaknya kategori dalam peubah respon terdiri dari dua kategori (Hosmer dan Lemeshow, 1989). Gambaran dari peubah respon *dikotomus* (biner) adalah sebagai berikut :

0 : bila peubah respon terjadi (sukses)

1 : bila peubah respon tidak terjadi (gagal)

Dalam menentukan nilai kategori dapat digunakan sembarang angka, akan tetapi untuk memudahkan maka digunakan angka 0 dan 1.

Agresti (1990) menyatakan jika terdapat peubah respon  $Y$  dan  $\pi$  adalah peluang sukses bagi peubah acak  $Y$  maka  $P(Y=1) = \pi$  dan  $P(Y=0) = 1 - \pi$ , jika banyaknya pengambilan  $n$  adalah 1 maka  $Y$  mengikuti sebaran Bernoulli sedangkan jika  $n \geq 2$  dan saling bebas maka  $Y$  mengikuti sebaran Binomial ( $n, \pi$ ). Peluang sebaran Binomial :

$$P(Y = y) = \binom{n}{y} \pi^y (1 - \pi)^{n-y} \quad (2.7)$$

dengan

$$y = 0, 1, 2, \dots, n$$

$n$  = banyaknya pengamatan

untuk menganalisis hubungan antara peluang peubah respon  $\pi_i$  dengan  $k$  peubah bebas  $X_{1i}, X_{2i}, \dots, X_{ki}$  didapatkan model peluang linier atau *Linier Probability Model* (LPM), yaitu :

$$\pi(X_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_k X_{ki} \quad (2.8)$$

dengan

$\pi(X_i) = E(Y|X_i)$  = nilai harapan peubah respon  $Y$  dengan syarat  $X_i$

$X_{1i}, X_{2i}, \dots, X_{ki}$  = peubah-peubah bebas ke- $k$  kategori ke- $i$

$\beta_0$  = intersep (titik potong garis regresi dengan sumbu  $Y$ )

$\beta_1, \beta_2, \dots, \beta_k$  = koefisien regresi parsial untuk setiap peubah bebas

$X_{1i}, X_{2i}, \dots, X_{ki}$  secara berturut-turut

$i$  = 1, 2, 3, ...,  $n$

$k$  = banyaknya peubah bebas

Model peluang linier tersebut tidak sah digunakan karena menimbulkan permasalahan sebagai berikut :

1. Model memungkinkan menghasilkan penduga peluang respon di luar interval  $[0, 1]$ , di mana hal ini bertentangan dengan sifat peluang. Untuk mengatasi maka dilakukan transformasi model dengan menggunakan transformasi logit  $g[\pi(X_i)] = \ln\left(\frac{\pi}{1-\pi}\right)$  sehingga interval  $[-\infty, \infty]$  menjadi  $[0,1]$ .

Adapun model regresi logistik menurut Agresti (1990) adalah :

$$\pi(X_i) = \frac{\exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})} \quad (2.9)$$

fungsi logit dari model regresi logistik adalah :

$$\begin{aligned} g[\pi(X_i)] &= \ln\left[\frac{\pi(X_i)}{1-\pi(X_i)}\right] \\ &= \ln\left[\frac{\frac{\exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})}}{1 - \frac{\exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})}}\right] \\ &= \ln[\exp(\beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki})] \\ &= \beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki} \quad (2.10) \end{aligned}$$

(Hosmer dan Lemeshow, 1989).

2. Ragam  $\hat{\pi}$  ( $Var(\hat{\pi}_i)$ ) tidak konstan sehingga *Ordinary Least Square* (OLS) tidak layak digunakan untuk pendugaan parameter. Masalah tersebut dapat diatasi dengan menggunakan metode kemungkinan maksimum (*Maximum Likelihood Method*) (Agresti, 1990).

Menurut Hosmer dan Lemeshow (1989), terdapat hal-hal yang perlu diperhatikan dalam analisis regresi logistik apabila peubah respon yang digunakan adalah *dikotomis* atau biner, yaitu:

1. Rata-rata peluang bersyarat pada model berada dalam kisaran 0 dan 1, dalam analisis regresi logistik dinyatakan dengan  $\pi(X_i)$ .
2. Peubah respon dalam regresi logistik tidak berdistribusi normal, akan tetapi berdistribusi binomial.
3. Prinsip-prinsip yang mendasari analisis regresi linier misalnya tidak adanya multikolinieritas antar peubah bebas, juga mendasari penggunaan analisis regresi logistik.

### 2.2.1 Multikolinieritas

Hosmer dan Lemeshow (1989) menjelaskan bahwa pada model regresi logistik juga sensitif dengan adanya kolinieritas, seperti pada regresi linier. Dalam model regresi, kolinieritas mengindikasikan bahwa terdapat hubungan ketergantungan atau korelasi tinggi antara dua peubah bebas. Menurut Gujarati (1991), adanya multikolinieritas berarti adanya hubungan linier yang tinggi di antara peubah-peubah bebas dalam model.

Apabila dalam suatu model regresi terdapat korelasi antara dua atau lebih peubah bebas, maka akan terdapat hubungan linier yang tinggi di antara peubah bebas dalam model. Penduga yang dihasilkan akan bersifat bias dan memiliki ragam yang mendekati tak terhingga (Sembiring, 1995). Adapun jika peubah bebas yang digunakan tidak diketahui distribusinya maka uji korelasi *rank Spearman* dapat digunakan karena uji ini tidak memiliki asumsi apapun. Berikut hipotesis uji korelasi *rank Spearman* :

$$H_0 : \rho_{x_j, x_j'} = 0 \quad \text{lawan} \quad H_1 : \rho_{x_j, x_j'} \neq 0$$

dengan statistik uji :

$$r_s = 1 - \left( \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} \right) \quad (2.11)$$

Jika  $H_0$  benar, maka  $\frac{r_s \sqrt{n-2}}{\sqrt{1-r_s^2}} \sim t_{\alpha/2(n-2)}$

dengan

$d_i$  = selisih antara peubah bebas  $X_{1i}$  dan  $X_{2i}$

$n$  = banyaknya pasangan data

jika korelasi antara kedua peubah bebas tersebut ada pada kisaran -0.7 dan 0.7 maka kolinieritas di antara dua peubah tersebut dapat dianggap tidak ada masalah (Lind, Robert, dan William, 2000). McClave dan Sincieh (2003) menjelaskan cara yang paling mudah untuk mengatasi masalah multikolinieritas adalah dengan menghilangkan salah satu peubah bebas yang dianggap memiliki korelasi tinggi.

### 2.2.2 Pendugaan Parameter Model Regresi Logistik

Parameter merupakan karakteristik dari suatu populasi. Nilai parameter secara eksak dapat diketahui melalui penelitian yang mengamati keseluruhan anggota populasi, di mana kegiatan ini dikenal dengan nama sensus. Sensus jarang dilakukan oleh seorang peneliti karena terdapat keterbatasan waktu, biaya, dan tenaga serta tidak dapat



dilakukan untuk penelitian yang sifatnya destruktif, maka dari itu perlu dilakukan suatu pendugaan parameter. Tujuan dari pendugaan parameter adalah mendapatkan nilai dugaan terbaik bagi parameter yang diamati (Sumodiningrat, 1994). Hosmer dan Lemeshow (1989) menjelaskan bahwa metode kemungkinan maksimum (*Maximum Likelihood Method*) mempunyai banyak kelebihan dibandingkan dengan metode yang lain yaitu dapat digunakan untuk model non linier seperti pada model logistik, sehingga metode ini lebih sering digunakan untuk menduga koefisien regresi logistik. Model logistik *dikotomus* (peubah respon biner), antar pengamatan diasumsikan saling bebas dan nilai peluang peubah responnya tidak linier terhadap  $\beta$ , untuk  $\mathbf{X}_i = (X_{1i}, X_{2i}, \dots, X_{ki})$ . Fungsi sebaran yang akan dimaksimumkan yaitu :

$$f(y_i; \pi_i) = \pi(\mathbf{X}_i)^{y_i} [1 - \pi(\mathbf{X}_i)]^{1-y_i} \quad (2.12)$$

Setiap pasang pengamatan  $(\mathbf{X}_i, \mathbf{Y}_i)$  bersifat bebas, maka fungsi kemungkinan (*likelihood function*) merupakan perkalian dari masing-masing fungsi peluang untuk  $\mathbf{X}_{1i}, \mathbf{X}_{2i}, \dots, \mathbf{X}_{ki}$  sehingga didapatkan :

$$\begin{aligned} L(\boldsymbol{\beta}) &= \prod_{i=1}^n \pi(\mathbf{X}_i)^{y_i} [1 - \pi(\mathbf{X}_i)]^{1-y_i} \\ &= \prod_{i=1}^n [1 - \pi(\mathbf{X}_i)] \left[ \frac{\pi(\mathbf{X}_i)}{1 - \pi(\mathbf{X}_i)} \right]^{y_i} \\ &= \prod_{i=1}^n \left[ \frac{1}{1 + \exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})} \right] [\exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})]^{y_i} \end{aligned} \quad (2.13)$$

Persamaan (2.13) perlu diubah ke dalam bentuk *log likelihood* sebagai berikut :

$$\text{Ln } L(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta})$$

$$\begin{aligned} \ell(\boldsymbol{\beta}) &= \ln \prod_{i=1}^n \left[ \frac{1}{1 + \exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})} \right] [\exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})]^{y_i} \\ &= \ln \left[ \frac{1}{1 + \exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})} \right] + \ln [\exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})]^{y_i} \\ &= n \left[ \ln 1 - \ln [1 + \exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})] \right] + \\ &\quad \sum_{i=1}^n y_i \left[ \exp \left( \beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji} \right) \right]^{y_i} \\ &= \sum_{i=1}^n y_i \ln [\exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})]^{y_i} - \\ &\quad \left[ n \ln \left[ 1 + \exp \left( \beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji} \right) \right] \right] \\ &= \sum_{i=1}^n y_i (\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji}) - n \ln [1 + \exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \beta_j X_{ji})] \end{aligned}$$



Agar nilai persamaan  $l(\beta)$  mencapai maksimum, turunan parsial terhadap  $\beta_j$  harus sama dengan nol dan  $\hat{\pi}(X_i)$  sebagai penduga bagi  $\pi(X_i)$ , maka :

$$\frac{\partial l(\hat{\beta})}{\partial(\hat{\beta}_j)} = \sum_{j=0}^k \sum_{i=0}^n y_i X_{ji} - n \sum_{i=1}^n X_{ji} \left[ \frac{\exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \hat{\beta}_j X_{ji})}{1 + \exp(\beta_0 + \sum_{j=0}^k \sum_{i=0}^n \hat{\beta}_j X_{ji})} \right]$$

$$= \sum_{j=0}^k \sum_{i=0}^n y_i X_{ji} - n \sum_{i=1}^n X_{ji} \hat{\pi}(X_i) = 0$$

Agresti (1990) menyatakan bahwa pendugaan parameter dilakukan dengan menggunakan persamaan :

$$\frac{\partial l(\hat{\beta})}{\partial(\hat{\beta}_j)} = \sum_{j=0}^k \sum_{i=0}^n y_i X_{ji} - n \sum_{i=1}^n X_{ji} \hat{\pi}(X_i) = 0 \quad (2.14)$$

dengan

$$y_i = \text{nilai kategori ke-}i \text{ pada peubah respon,}$$

$$X_{ji} = \text{nilai kategori ke-}i \text{ pada peubah } X \text{ ke-}j,$$

$$j = 0, 1, \dots, k$$

dalam menyelesaikan persamaan (2.14) tidaklah mudah, karena  $\beta_j$  yang akan diduga bersifat non linier, sehingga diperlukan metode iterasi di mana dalam perhitungannya dibantu komputer. Adapun metode iterasi yang digunakan adalah iterasi *Newton Raphson*. Metode ini biasa digunakan untuk menyelesaikan persamaan non linier seperti menentukan nilai maksimum suatu fungsi. Menurut Agresti (1990) metode *Newton Raphson* menentukan  $\hat{\beta}$  dengan memaksimalkan  $l(\beta)$ . Pandang  $u'$  sebagai matriks turunan pertama  $l(\beta)$  terhadap  $\beta_0$  dan  $\beta_j$  yaitu :

$$u' = \left( \frac{\partial l(\beta)}{\partial \beta_0}, \frac{\partial l(\beta)}{\partial \beta_1}, \dots, \frac{\partial l(\beta)}{\partial \beta_k} \right)$$

Pandang  $H$  (matriks Hessian) adalah matriks yang berisi unsur-unsur turunan kedua  $l(\beta)$  terhadap  $\beta_a$  dan  $\beta_b$  ( $a, b = 0, 1, \dots, k$ ) yaitu :

$$H_{ab} = \frac{\partial^2 l(\beta)}{\partial \beta_a \partial \beta_b} = - \sum_{i=1}^n X_{ia} X_{ib} n_i \pi(X_i) (1 - \pi(X_i))$$

Menyatakan  $u^{(t)}$  dan  $H^{(t)}$  sebagai  $u$  dan  $H$  yang terevaluasi pada  $\beta^{(t)}$  yaitu penduga  $\hat{\beta}$  pada iterasi ke- $t$  ( $t = 0, 1, 2, \dots$ ). Nilai  $\beta^{(t)}$  pada setiap iterasi mengikuti bentuk deret Taylor orde kedua yaitu :

$$L(\beta) \approx L(\beta^{(t)}) + u^{(t)}(\beta - \beta^{(t)}) + \frac{1}{2}(\beta - \beta^{(t)})' H^{(t)}(\beta - \beta^{(t)})$$

dengan menyelesaikan  $\frac{\partial l(\hat{\beta})}{\partial(\hat{\beta})} \approx u^{(t)} - H^{(t)}(\hat{\beta} - \beta^{(t)}) = 0$ , maka  $\hat{\beta}$  pada iterasi *Newton Raphson* adalah :

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - [\mathbf{H}^{(t)}]^{-1} \mathbf{u}^{(t)} \quad (2.15)$$

dengan nilai  $\mathbf{H}^{(t)}$  dan  $\mathbf{u}^{(t)}$  berupa matriks :

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left\{ -\mathbf{X}' \text{diag}[n_i \pi(\mathbf{X}_i^{(t)})(1 - \pi(\mathbf{X}_i)) \mathbf{X}]^{-1} \mathbf{X}' (y_i - n_i \pi(\mathbf{X}_i^{(t)})) \right\}$$

Kekonvergenan  $\boldsymbol{\beta}^{(t)}$  yang mendekati  $\hat{\boldsymbol{\beta}}$  terpenuhi jika pada setiap- $j$

$$\left| \boldsymbol{\beta}_j^{(t+1)} - \hat{\boldsymbol{\beta}}_j^{(t)} \right| \leq c \left| \boldsymbol{\beta}_j^{(t)} - \hat{\boldsymbol{\beta}}_j^{(t-1)} \right|^2 \text{ untuk } c > 0.$$

(Agresti, 1990).

Langkah-langkah iterasi *Newton Raphson* adalah :

1. Menentukan nilai awal dugaan parameter yaitu  $\boldsymbol{\beta}^{(t)}$ . Menurut Gerald dan Wheatly (1996), nilai  $\boldsymbol{\beta}^{(t)}$  berdekatan dengan selisih antara nilai penduga dan dugaan yang diharapkan, hal ini disebabkan agar proses iterasi tidak terlalu banyak. Sehingga untuk mempermudah perhitungan digunakan 0 sebagai nilai dugaan awal iterasi.

2. Membentuk keseluruhan data dalam bentuk matriks.

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1n} \\ 1 & X_{21} & X_{22} & \dots & X_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & X_{k1} & X_{k2} & \dots & X_{kn} \end{bmatrix} \text{ dan } \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_p \end{bmatrix}$$

3. Menentukan model regresi logistik pada persamaan (2.9) yaitu :

$$\pi_i^{(t)} = \frac{\exp(x_i \boldsymbol{\beta}^{(t)})}{1 + \exp(x_i \boldsymbol{\beta}^{(t)})}; x_i = [1 \quad X_{1i} \quad X_{2i} \quad \dots \quad X_{kn}]$$

4. Menentukan matriks  $\mathbf{V}$

$$\mathbf{v}^{-1} = \begin{bmatrix} \pi(\mathbf{X}_i^{(t)})(1 - \pi(\mathbf{X}_i)) & 0 & \dots & 0 \\ 0 & \pi(\mathbf{X}_2^{(t)})(1 - \pi(\mathbf{X}_2)) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \pi(\mathbf{X}_n^{(t)})(1 - \pi(\mathbf{X}_n)) \end{bmatrix}$$

5. Mensubstitusi hasil dari langkah pada persamaan (2.15) dengan  $\mathbf{H}^{(t)} = \mathbf{X}' \mathbf{V}^{(t)} \mathbf{X}$  (2.16)

dan

$$\mathbf{u}^{(t)} = \mathbf{X}' (y_i - n_i \pi(\mathbf{X}_i^{(t)})) \quad (2.17)$$

$$\pi(\mathbf{X}_i^{(t)}) = \begin{bmatrix} \pi(\mathbf{X}_1^{(t)}) \\ \pi(\mathbf{X}_2^{(t)}) \\ \dots \\ \pi(\mathbf{X}_n^{(t)}) \end{bmatrix}$$

6. Diperoleh  $\boldsymbol{\beta}_j^{(t+1)}$

7. Menentukan selisih  $(\beta_j^{(t+1)} - \hat{\beta}_j^{(t)})$  jika hasil selisihnya memenuhi kondisi konvergen yaitu bernilai  $(1 \times 10^{-7}) = 0,0000001$  maka  $\hat{\beta}_j^{(t+1)}$  merupakan hasil pendugaannya).

Bila kondisi konvergen belum terpenuhi maka iterasi dilanjutkan terus dengan cara menjadikan nilai  $\hat{\beta}_j^{(t+1)}$  sebagai  $\beta_j^{(t)}$  untuk iterasi selanjutnya, diulangi mulai langkah ke-3 sampai ke-7 hingga diperoleh nilai  $\beta$  yang konvergen.

### 2.2.3 Pengujian terhadap Penduga Parameter

Menurut Hosmer dan Lemeshow (1989), uji signifikan merupakan suatu uji apakah parameter yang diduga dengan suatu peubah tertentu dalam suatu model lebih baik dibandingkan dengan model tanpa peubah tersebut. Dengan kata lain, tujuan pengujian hipotesis secara statistika untuk menentukan apakah peubah-peubah bebas dalam model mempunyai hubungan yang nyata dengan peubah respons.

Pembandingan nilai-nilai pengamatan dengan nilai-nilai prediksi pada dua model dapat digunakan untuk mengetahui seberapa penting peubah berada pada model, apakah dengan penambahan peubah di dalam model akan dapat menjelaskan suatu peubah respons dibandingkan tanpa peubah bebas tersebut. Model-model yang dimaksud yaitu model tanpa peubah tersebut dan model yang memuat peubah itu, jika nilai prediksi model yang melibatkan peubah memang baik dan lebih akurat dibandingkan nilai prediksi model tanpa peubah tersebut, maka dapat disimpulkan bahwa peubah tersebut memang berarti atau berpengaruh.

Pengujian terhadap parameter dapat dilakukan secara simultan dan parsial.

1. Pengujian  $\hat{\beta}_j$  secara simultan atau serentak bertujuan untuk membandingkan nilai pengamatan respons dengan nilai dugaan respons untuk kedua model. Model yang pertama adalah model penuh dan model yang kedua adalah model reduksi dengan hipotesis sebagai berikut :

$$H_0 : \beta_1 = \beta_2 = \dots = \beta_j = 0 \text{ lawan}$$

$$H_1 : \text{paling sedikit ada satu } \beta_j \neq 0, j = 1, 2, \dots, k$$

Uji yang digunakan adalah uji nisbah kemungkinan (*Likelihood Ratio Test*) yaitu :

$$G = -2 \ln \left[ \frac{L_0(\boldsymbol{\beta})}{L_p(\boldsymbol{\beta})} \right] = -2 \left[ \ln L_0(\boldsymbol{\beta}) - \ln L_p(\boldsymbol{\beta}) \right] \quad (2.19)$$

$L_0(\boldsymbol{\beta})$  yaitu fungsi kemungkinan dari himpunan parameter di bawah  $H_0$  atau  $L_0(\boldsymbol{\beta}) = L(\beta_0, \theta)$  dan  $L_p(\boldsymbol{\beta}) = L(\beta_0, \beta_1, \beta_2, \dots, \beta_p, \theta)$ .  
Penjabaran dari  $L_0(\boldsymbol{\beta})$  adalah :

$$L_0(\boldsymbol{\beta}) = L(\beta_0, \theta)$$

$$= \prod_{i=1}^n \binom{n_i}{y_i} \frac{\left[ \prod_{l=0}^{y_i-1} \left( \frac{\exp(\beta_0)}{1 + \exp(\beta_0)} + \theta \right) \right] \left[ \prod_{l=0}^{n_i-y_i-1} \left( \frac{\exp(\beta_0)}{1 + \exp(\beta_0)} + \theta \right) \right]}{\prod_{l=0}^{n_i-1} (1 + \theta)}$$

$$\begin{aligned} \ln L_0(\boldsymbol{\beta}) &= \sum_i \sum_{l=0}^{y_i-1} \ln \left( \frac{\exp(\beta_0)}{1 + \exp(\beta_0)} + \theta \right) \\ &+ \sum_i \sum_{l=0}^{n_i-y_i-1} \ln \left( 1 - \frac{\exp(\beta_0)}{1 + \exp(\beta_0)} + \theta \right) - \sum_i \sum_{l=0}^{n_i-1} \ln(1 + \theta) \\ &= \sum_i \sum_{l=0}^{y_i-1} \ln(\theta) + \sum_i \sum_{l=0}^{n_i-y_i-1} \ln(\theta) - \sum_i \sum_{l=0}^{n_i-1} \ln(\theta) \end{aligned}$$

Penjabaran dari  $L_p(\boldsymbol{\beta})$  adalah :

$$L_p(\boldsymbol{\beta}) = L(\beta_0, \beta_1, \beta_2, \dots, \beta_p, \theta)$$

$$= \prod_{i=1}^n \binom{n_i}{y_i} \frac{\left[ \prod_{l=0}^{y_i-1} \left( \frac{\exp(X_j, \boldsymbol{\beta})}{1 + \theta \exp(X_j, \boldsymbol{\beta})} + \theta \right) \right] \left[ \prod_{l=0}^{n_i-y_i-1} \left( \frac{\exp(X_j, \boldsymbol{\beta})}{1 + \theta \exp(X_j, \boldsymbol{\beta})} + \theta \right) \right]}{\prod_{l=0}^{n_i-1} (1 + \theta)}$$

$$\begin{aligned} \ln L_p(\boldsymbol{\beta}) &= \sum_i \sum_{l=0}^{y_i-1} \ln \left( \frac{\exp(X_j, \boldsymbol{\beta})}{1 + \theta \exp(X_j, \boldsymbol{\beta})} + \theta \right) \\ &+ \sum_i \sum_{l=0}^{n_i-y_i-1} \ln \left( 1 - \frac{\exp(X_j, \boldsymbol{\beta})}{1 + \theta \exp(X_j, \boldsymbol{\beta})} + \theta \right) - \sum_i \sum_{l=0}^{n_i-1} \ln(1 + \theta) \\ &= \sum_i \sum_{l=0}^{y_i-1} \ln(1 - \theta) \exp(X_j, \boldsymbol{\beta}) + \sum_i \sum_{l=0}^{y_i-1} \ln(\theta) \\ &- \sum_i \sum_{l=0}^{n_i-y_i-1} \ln(1 - \theta) \exp(X_j, \boldsymbol{\beta}) + \sum_i \sum_{l=0}^{n_i-y_i-1} \ln(\theta) \\ &- \sum_i \sum_{l=0}^{n_i-1} \ln(\theta) \end{aligned}$$

Persamaan (2.19) menjadi seperti :

$$G = -2 \left[ - \left[ \sum_i \sum_{l=0}^{y_i-1} \ln(1 - \theta) \exp(X_j, \boldsymbol{\beta}) - \sum_i \sum_{l=0}^{n_i-y_i-1} \ln(1 - \theta) \exp(X_j, \boldsymbol{\beta}) \right] \right]$$

Statistik uji  $G$  mengikuti sebaran  $\chi^2(v)$ , dengan derajat bebas  $v$  di mana  $v$  menyatakan banyak parameter di bawah  $H_0$ .

Hipotesis nol akan ditolak jika nilai  $\chi^2_{(1-\alpha; v)} < G$ . Hal ini mengindikasikan bahwa paling sedikit ada satu  $\beta_j$  yang tidak sama dengan nol dengan  $\alpha$  adalah taraf nyata yang digunakan.

2. Pengujian koefisien  $\hat{\beta}$  secara parsial dilandasi pada perbandingan penduga dengan ragam penduganya dengan hipotesis :

$$H_0 : \beta_j = 0 \quad \text{lawan} \quad H_1 : \beta_j \neq 0$$

Uji koefisien regresi secara parsial digunakan untuk memeriksa peranan koefisien regresi dari masing-masing peubah bebas secara individu harus berada dalam model (Hosmer dan Lemeshow, 1989). Apabila  $H_0$  benar, statistik uji Wald :

$$W_j = \frac{\hat{\beta}_j}{[SE(\hat{\beta}_j)]} \sim Z \quad (2.18)$$

Dengan  $\hat{\beta}_j$  adalah penduga bagi  $\beta_j$ , dan  $SE(\hat{\beta}_j)$  merupakan penduga galat baku bagi  $\beta_j$ . Nilai  $SE(\hat{\beta}_j)$  diperoleh dari nilai diagonal utama matriks kovarian yaitu :

$$\text{Cov}(\hat{\beta}_j) = [X'V^{-1}X]^{-1}$$

$$V^{-1} = \begin{bmatrix} \pi(X_1^{(l)})(1 - \pi(X_1)) & 0 & \dots & 0 \\ 0 & \pi(X_2^{(l)})(1 - \pi(X_2)) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \pi(X_n^{(l)})(1 - \pi(X_n)) \end{bmatrix}$$

$$\text{Var}(\hat{\beta}_j) = \text{diagonal utama matriks } [X'V^{-1}X]^{-1}$$

$$SE(\hat{\beta}_j) = \sqrt{\text{Var}(\hat{\beta}_j)}$$

Statistika uji  $W$  mengikuti sebaran normal baku. Hipotesis nol ditolak jika  $|W_j| > Z_{(1-\frac{\alpha}{2})}$  di mana  $\alpha$  adalah taraf nyata yang digunakan. Hal ini mengindikasikan bahwa peubah bebas  $X_j$  berpengaruh nyata terhadap peubah respons.

### 2.2.4 Uji Kesesuaian Model (Goodness of fit)

Pengujian kesesuaian model atau *goodness of fit* dilakukan untuk mengetahui apakah model yang dihasilkan sudah sesuai atau belum. Hipotesis yang digunakan dalam pengujian ini yaitu :

$$H_0 : \text{model sesuai} \quad \text{lawan} \quad H_1 : \text{model tidak sesuai}$$

Statistik uji yang digunakan untuk menguji kesesuaian model adalah uji *Pearson*.

$$\chi^2_{\text{Pearson}} = \sum_{i=1}^h r(y_i, \hat{\pi}_i)^2 \quad (2.20)$$



$$\text{di mana } r(y_i, \hat{\pi}_i) = \frac{(y_i - n_i \hat{\pi}_i)^2}{n_i \hat{\pi}_i (1 - \hat{\pi}_i)}$$

dengan

$\hat{\pi}_i$  = peluang Y pada saat X dengan kategori ke- $i$

$y_i$  = pengamatan  $n_i$  dengan  $x = x_i$  di mana  $y=1$

$n_i$  = pengamatan yang memenuhi persamaan  $x = x_i$

Statistik *Pearson* menyebar mengikuti sebaran khi-kuadrat dengan derajat bebas  $v$ . Keputusan menolak  $H_0$ , jika  $\chi^2_{Pearson} > \chi^2_{(v)}$  atau  $P[\chi^2_{(v)} > \chi^2_{Pearson}]$  lebih kecil dari peluang yang diinginkan ( $\alpha$ ), sehingga dapat diambil kesimpulan bahwa model yang diperoleh tidak sesuai (Fahrmeir dan Gerhard, 1994).

## 2.3 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

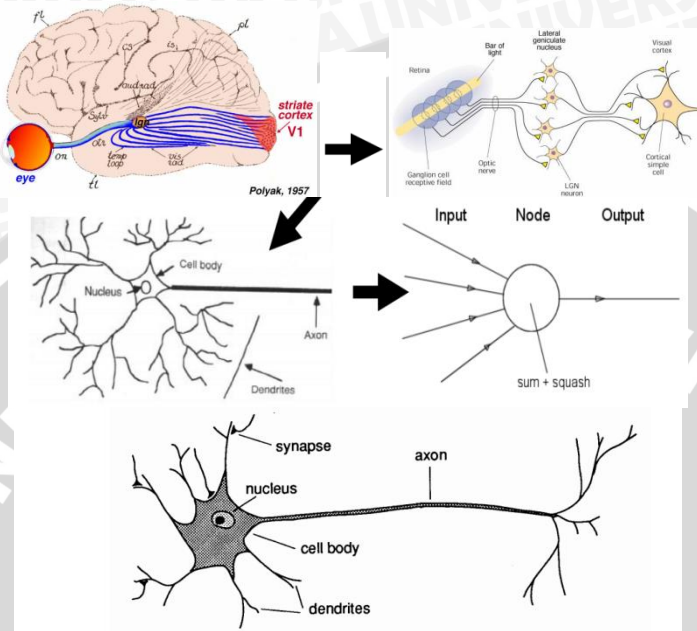
### 2.3.1 Sejarah Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) atau *Artificial Neural Network* (ANN) telah dikembangkan sejak tahun 1940 dan pertama kalinya pada tahun 1943 oleh *neurophysiologist*, Warren McCulloch dan *logician* Walter Pitts memperkenalkan pemodelan matematis neuron, namun teknologi yang tersedia pada saat itu belum memungkinkan mereka berbuat lebih jauh. Tahun 1949, Hebb mencoba mengkaji proses belajar yang dilakukan oleh neuron dan teori ini dikenal sebagai Hebbian Law. Tahun 1958, Rosenblatt memperkenalkan konsep perseptron suatu Jaringan yang terdiri dari beberapa lapisan yang saling berhubungan melalui umpan maju (*feed forward*). Tahun 1962, hasil kerja Rosenblatt yang sangat penting yaitu *perceptron convergence theorem* membuktikan bahwa bila setiap perseptron dapat memilah-milah dua buah pola yang berbeda maka siklus pelatihannya dapat dilakukan dalam jumlah yang terbatas. Tahun 1960 Widrow dan Hoff menemukan ADALINE (*Adaptive Linier Neuron*). Alat ini dapat beradaptasi dan beroperasi secara linier. Penemuan ini telah memperlebar aplikasi jaringan syaraf tiruan tidak hanya untuk pemilihan pola, tetapi juga untuk pengiriman sinyal khususnya dalam bidang *adaptive filtering*. Tahun 1969, Minsky dan Papert melontarkan suatu kritikan tentang kelemahan perseptron. Rosenblatt di dalam memilah-milah pola yang tidak linier. Sejak saat itu penelitian di bidang jaringan syaraf tiruan telah mengalami masa vakum untuk kurang lebih satu dasawarsa. Tahun 1982, Hopfield telah memperluas aplikasi JST untuk memecahkan

masalah-masalah optimasi. Hopfield telah berhasil memperhitungkan fungsi energi ke dalam jaringan syaraf yaitu agar jaringan memiliki kemampuan untuk mengingat atau memperhitungkan suatu objek dengan objek yang pernah dikenal atau diingat sebelumnya (*associative memory*). Konfigurasi jaringan yang demikian dikenal sebagai *recurrent network*. Salah satu aplikasinya adalah *Travelling Salesman Problem* (TSP). Tahun 1986, Rumelhart, Hinton dan William menciptakan suatu algoritma belajar yang dikenal sebagai propagasi balik (*backpropagation*). Bila algoritma ini diterapkan pada *perceptron* yang memiliki lapisan banyak (*multi layer perceptron*), maka dapat dibuktikan bahwa pemilihan pola-pola yang tidak linier dapat diselesaikan sehingga dapat mengatasi kritikan yang dilontarkan oleh Minsky dan Papert. Tahun 1988, Broomhead dan Lowe's memperkenalkan metode *Radial Basis Function* (RBF) melalui makalahnya yang berjudul, "*Multivariate functional interpolation and adaptive network*". RBF adalah salah satu bentuk *Multi Layer Perceptron* yang tidak ada *update* bobot *learning rate*. Model ini adalah model jaringan syaraf dengan satu unit dalam lapisan tersembunyi, di mana fungsi aktivasinya adalah fungsi basis (*Gaussian*) dan fungsi linier pada lapisan *output* (Darmawan, 2009).

JST adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi (Siang, 2005). JST seperti manusia, belajar dari suatu contoh. JST dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran. JST berkembang secara pesat pada beberapa tahun terakhir. JST telah dikembangkan sebelum adanya suatu konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun. Banyak model JST yang sama atau identik dengan model statistika, *terminology* dalam literatur JST berbeda dengan literatur yang ada dalam statistika. Istilah-istilah statistika yang ada dalam JST meliputi variabel bebas disebut *input*, nilai prediksi disebut *output*, variabel terikat disebut *target*, residual disebut *error*, dan *estimation* disebut *training*, *learning*, *adaption*.

Neuron adalah satuan unit pemroses terkecil pada otak, bentuk sederhana sebuah neuron yang oleh para ahli dianggap sebagai satuan unit pemroses tersebut digambarkan sebagai berikut :

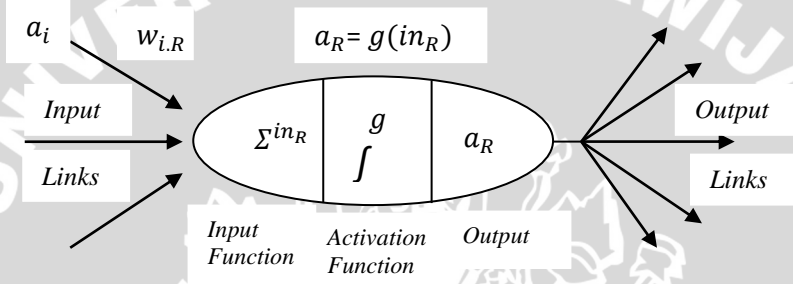


Gambar 2.1 Struktur dasar JST dan Struktur sederhana sebuah neuron (Pakaja, 2012).

Struktur pada Gambar 2.1 tersebut bentuk standar dasar satuan unit jaringan otak manusia yang telah disederhanakan. Bentuk standar ini mungkin dikemudian hari akan berubah bila ada ilmuwan yang dapat menciptakan bentuk standar yang lebih baik ataupun memperbaiki bentuk standar yang digunakan saat ini. Jaringan otak manusia tersusun tidak kurang dari  $10^{13}$  buah neuron yang masing-masing terhubung oleh sekitar  $10^{15}$  buah *dendrite*. Fungsi dendrit adalah sebagai penyampai sinyal dari neuron tersebut ke neuron yang terhubung dengannya. Sebagai keluaran, setiap neuron memiliki *axon*, sedangkan bagian penerima sinyal disebut *synapse*. Penjelasan lebih rinci tentang hal ini dapat diperoleh pada disiplin ilmu biologi *moleculer*. Secara umum jaringan syaraf terbentuk dari jutaan (bahkan lebih) struktur dasar neuron yang terinterkoneksi dan terintegrasi antara satu dengan yang lain sehingga dapat melaksanakan aktifitas secara teratur dan terus menerus sesuai dengan kebutuhan.

### 2.3.2 Konsep Dasar Pemodelan Jaringan Syaraf Tiruan

Tiruan neuron dalam struktur jaringan syaraf tiruan adalah sebagai elemen pemroses seperti Gambar 2.2 yang dapat berfungsi seperti halnya sebuah neuron. Sejumlah sinyal masukan  $a$  dikalikan dengan masing-masing penimbang yang bersesuaian  $w$  kemudian dilakukan penjumlahan dari seluruh hasil perkalian tersebut dan keluaran yang dihasilkan dilakukan kedalam fungsi pengaktif untuk mendapatkan tingkatan derajat sinyal keluarannya  $f(a, w)$ . Walaupun masih jauh dari sempurna, namun kinerja dari tiruan neuron ini identik dengan kinerja dari sel biologi yang kita kenal saat ini.



Gambar 2.2 Model tiruan sebuah neuron (Suhartono, 2007).

keterangan :

- $a_i$  : Nilai aktivasi dari unit  $i$
- $w_{i,R}$  : Bobot dari unit  $i$  ke unit  $R$
- $in_R$  : Penjumlahan bobot dan masukan ke unit  $R$
- $g$  : Fungsi aktivasi
- $a_R$  : Nilai aktivasi dari unit  $R$

Misalkan ada  $n$  buah sinyal masukan dan  $n$  buah penimbang, fungsi keluaran dari neuron adalah seperti persamaan (2.21) berikut,

$$in_R = \sum_i w_{iR} * a_i \quad (2.21)$$

kumpulan dari neuron dibuat menjadi sebuah jaringan yang akan berfungsi sebagai alat komputasi. Jumlah neuron dan struktur jaringan untuk setiap problema yang akan diselesaikan adalah berbeda (Suhartono, 2007).

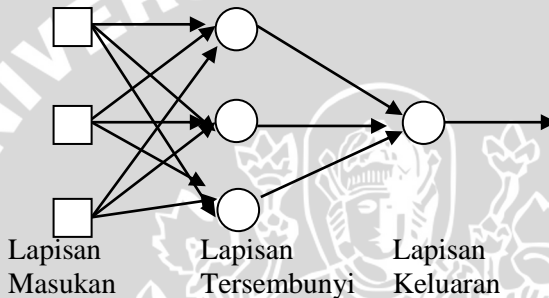
### 2.3.3 Arsitektur Jaringan Syaraf Tiruan

Jaringan melakukan proses pembelajaran yang mirip dengan proses pembelajaran pada otak manusia. Proses pembelajaran dan pengolahan informasi serta penyimpanan kecerdasan dalam JST sangat



bergantung pada struktur (arsitektur) dan algoritma pembelajaran yang digunakan dalam JST.

Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output*, seperti pada Gambar 2.3. Umumnya, ada lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit dibandingkan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun pada banyak kasus pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah (Orr, 1996).



Gambar 2.3 Jaringan syaraf dengan banyak lapisan (Orr, 1996).

Yao dan Tan (2001) menyatakan bahwa pada pemilihan arsitektur jaringan, tipe jaringan yang digunakan adalah tipe yang sesederhana mungkin namun dengan hasil yang seefisien mungkin karena semakin kompleks tipe jaringan yang digunakan dan semakin banyak *hidden layer* dan unit *hidden* yang digunakan tidak menjamin peramalan yang dihasilkan selalu lebih baik.

## 2.4 Metode Pelatihan Jaringan Syaraf Tiruan (JST)

### 2.4.1 Proses Pembelajaran JST

Prosedur yang digunakan JST dalam mencari pengaturan bobot yang tepat mengacu pada paradigma belajarnya (*Learning Paradigm*). Paradigma belajar inilah yang dikenal sebagai algoritma belajar dalam JST. Berdasarkan strategi pelatihan, paradigma belajar JST dapat diklasifikasikan menjadi dua paradigma :

1. Pembelajaran Terawasi (*Supervised Learning*)

Metode pembelajaran pada jaringan syaraf tersebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. JST akan mempelajari dan menentukan pola *input* dan *output*. Proses ini



berhenti jika selisih antara *output* dari JST dan *output* yang sebenarnya sudah mencapai titik konvergen. Contoh metode pelatihan terawasi adalah *perceptron learning*, *delta rule*, dan *backpropagation*.

## 2. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

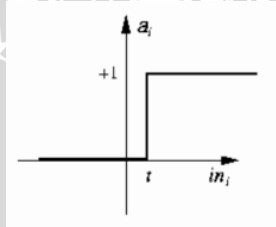
Pada metode pembelajaran tak terawasi ini tidak memerlukan target *output*. Pada metode ini, tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *output* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (*cluster*) pola. Contoh metode tak terawasi antara lain *hibblian learning* dan *competitive learning*.

Semakin tinggi tingkat ketidaklinieran suatu sistem, sering kali sulit diselesaikan dengan metode pelatihan terbimbing dan tak terbimbing, maka banyak periset yang mencoba menggabungkannya dan diperoleh hasil yang lebih baik dengan salah satu metode hibrida yang berhasil dan banyak digunakan yaitu metode *Radial Basis Function Neural Network* (RBFNN) (Patterson, 1996).

### 2.4.2 Fungsi Aktivasi pada JST

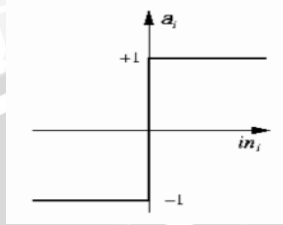
Mengaktifkan jaringan syaraf tiruan berarti mengaktifkan setiap neuron yang dipakai pada jaringan. Banyak fungsi yang dapat dipakai sebagai pengaktif, seperti fungsi-fungsi geometri dan hiperboliknya, fungsi unit *step*, *impulse*, *sigmoid*, dan lain sebagainya seperti pada Gambar 2.4 - 2.6, tetapi yang lazim digunakan adalah fungsi *sigmoid*, karena dianggap lebih mendekati kinerja sinyal pada otak.

$$f(x) = \begin{cases} 1, & x \geq t \\ 0, & \text{lainnya} \end{cases}$$



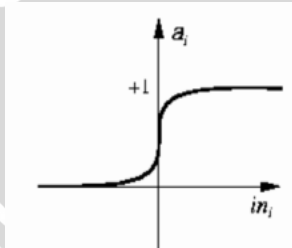
Gambar 2.4 Fungsi unit *step*

$$f(x) = \begin{cases} +1, & x \geq 0 \\ -1, & \text{lainnya} \end{cases}$$



Gambar 2.5 Fungsi *sign*

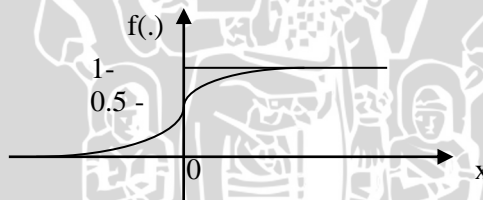
$$f(x) = 1/(1+e^{-x})$$



Gambar 2.6 Fungsi *sigmoid*

Ada dua jenis fungsi *sigmoid*, *unipolar* dan *bipolar*. Fungsi *sigmoid unipolar* ditunjukkan pada Gambar 2.7.

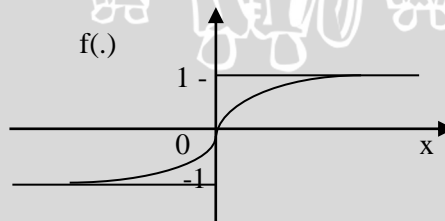
$$f(x) = \left( \frac{1}{1+e^{-x}} \right)$$



Gambar 2.7 Fungsi *sigmoid unipolar*

Sedangkan fungsi pengaktif *bipolar* adalah Gambar 2.8.

$$f(x) = \left( \frac{1-e^{-x}}{1+e^{-x}} \right)$$



Gambar 2.8 Fungsi *sigmoid bipolar*

Fungsi *tangent* hiperbolik merupakan fungsi sigmoid yang sangat menyerupai fungsi bipolar, dan sering digunakan sebagai fungsi aktivasi ketika rentang yang diinginkan antara -1 dan 1.

$$f(x) = \left( \frac{1 - e^{(-2x)}}{1 + e^{(-2x)}} \right)$$

(Siang, 2005).

## 2.5 *Radial Basis Function Neural Network (RBFNN)*

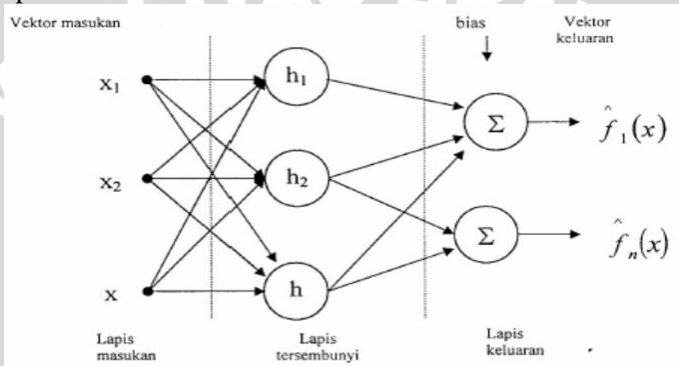
RBFNN biasanya membutuhkan neuron lebih banyak jika dibandingkan dengan jaringan *feed forward*. Jaringan ini akan bekerja dengan baik apabila data *input* yang diberikan cukup banyak. *Input* yang akan diolah oleh fungsi aktivasi RBFNN bukan merupakan hasil penjumlahan terbobot dari data *input*, namun berupa vektor jarak antara vektor bobot dan vektor input yang dikalikan dengan bobot bias. Fungsi aktivasi yang digunakan adalah :

$$radbas(n) = e^{-n^2}$$

RBFNN ini memiliki nilai maksimum 1 yang terjadi apabila *input* yang diterima bernilai 0 (jarak antara bobot dengan *input* 0). Sehingga apabila jarak antara bobot dengan *input* berkurang, fungsi ini akan memberikan *output* lebih besar (Kusumadewi, 2004). Model jaringan ini, neuron-neuron keluarannya adalah hasil kombinasi linier fungsi basis neuron-neuron pada lapisan tersembunyi. Sebagai fungsi basis yang umum digunakan adalah *Gaussian*. Perbedaan utama antara jaringan *multilayerperception* (MLP) dengan jaringan berbasis fungsi radial yaitu penggunaan *Gaussian* pada satu lapisan tersembunyi jaringan RBF, sedangkan jaringan MLP menggunakan fungsi sigmoid (Purnomo dan Kurniawan, 2006).

Model RBF adalah model jaringan syaraf dengan satu unit dalam lapisan tersembunyi, di mana fungsi aktivasinya adalah fungsi basis (*Gaussian*) dan fungsi linier pada lapisan *output*. Hubungan antara neuron masukan dan neuron *receptor* di latih terus menerus untuk mendapatkan bobot dari masing-masing *connection* sehingga di dapatkan model prediksi data yang paling tepat (Buhmann, 1998). Pada prinsipnya RBF adalah emulsi sifat jaringan biologi yang umumnya sel/neuron yang paling aktif adalah sel/neuron yang paling sensitif menerima rangsangan sinyal masukan, sehingga orientasi sensitifitas respon tersebut hanya terhadap beberapa daerah (*local response*) dalam wilayah masukan. JST dengan lapisan tersembunyi tunggal, pada

dasarnya lapisan tersebut berisi neuron-neuron (unit-unit) yang sensitif atau aktif secara lokal. Sedangkan keluarannya terdiri dari unit-unit linier. Pada unit-unit dalam lapisan tersembunyi, respon unitnya bersifat lokal dan berkurang sebagai fungsi jarak masukan dari pusat unit penerima rangsangan. Metode ini menjadi terkenal sejak Broomhead dan Lowe's pada tahun 1988 menyampaikan makalahnya yang berjudul "*Multivariate functional interpolation and adaptive network*". RBFNN mempunyai kesamaan dasar dengan MLPNN yang struktur dasarnya ditunjukkan pada Gambar 2.9 :



Gambar 2.9 Arsitektur RBFNN (Broomhead dan Lowe's, 1988). Hal yang bersifat khusus pada jaringan syaraf tiruan RBF ialah sebagai berikut :

- Pemrosesan sinyal dari *input layer* ke *hidden layer*, sifatnya non linier, sedangkan dari *hidden layer* ke *output layer* sifatnya linier.
- Pada *hidden layer* digunakan sebuah fungsi aktivasi berbasis radial, misalnya fungsi *Gaussian*.
- Pada *output unit*, sinyal dijumlahkan dengan metode *ordinary least square*.
- Sifat jaringannya ialah satu arah, tidak seperti pada jaringan *Feed Forward Neural Network (FFNN)* yang bersifat bolak-balik.

### 2.5.1 Fungsi Basis Radial

Fungsi radial adalah suatu fungsi yang mempunyai karakteristik merespon pengurangan ataupun penambahan secara monoton dengan

jarak yang berasal dari nilai tengahnya. Jenis fungsi radial yang banyak digunakan adalah fungsi *Gaussian* seperti pada persamaan (2.22) :

$$h_R = \exp\left(-\frac{\|x_R - \mu_m\|^2}{2\sigma_m^2}\right) \quad (2.22)$$

dengan

- $h_R$  = fungsi aktivasi *Gaussian* ke- $R$
- $x_R$  = penjumlahan bobot *input* dan nilai *input* ke- $R$
- $\mu_m$  = rata-rata dari tiap fitur pada kelas klasifikasi ke- $m$
- $\sigma_m^2$  = standar deviasi dari fungsi rata-rata ke- $m$

Beberapa tipe fungsi aktivasi RBF yang lain adalah sebagai berikut:

- Fungsi *Thinplate-Spline*  

$$\phi(z, 1) = z^2 \log(z)$$
- Fungsi *Multiquadratic*  

$$\phi(z, \sigma) = (z^2 + \sigma^2)^{1/2}$$
- Fungsi *Inverse Multiquadratic*  

$$\phi(z, \sigma) = \frac{1}{(z^2 + \sigma^2)^{1/2}}$$

Hasil penelitian menyatakan bahwa seleksi dari keempat fungsi nonlinier tersebut tidak dominan menentukan kinerja RBF. Bila jarak *Euclidian* antara vektor masukan dan unit-unit dalam lapis tersembunyi mempunyai nilai yang berbeda, maka jarak yang sama untuk setiap unitnya cukup untuk pendekatan secara universal. Ini berarti bahwa jarak dapat disesuaikan pada sebuah nilai  $\sigma$  untuk menyederhanakan strategi pelatihannya (Mark, 1999).

### 2.5.2 Optimalisasi Vektor Pembobot

Pada metode *Least Square* diterapkan pelatihan terbimbing (*supervised learning*) dengan perhitungan galatnya sebagai fungsi *Sum Square Error* (SSE).

$$SSE = \sum_{m=1}^p (y_m - \hat{y}_m)^2 \quad (2.23)$$

$y_m = [y_1, y_2, \dots, y_p]^T$  adalah target atau peubah respon di mana  $\hat{y}_m$  adalah nilai prediksi.

Nilai dasar fungsi disesuaikan dengan  $m$  buah persamaan linier secara simultan dengan pembobot dan persamaan linier tersebut dinyatakan sebagai persamaan matriks berikut :

$$A\hat{w} = h_R^T \hat{y}$$

Matriks  $h_R$  adalah :



$$\mathbf{h}_R = \begin{bmatrix} h_1(x)_1 & h_2(x)_1 & \dots & h_Q(x)_1 \\ h_1(x)_2 & h_2(x)_2 & \dots & h_Q(x)_2 \\ \dots & \dots & \dots & \dots \\ h_1(x)_p & h_2(x)_p & \dots & h_Q(x)_p \end{bmatrix}$$

$\mathbf{A}^{-1}$  adalah matriks varian :

$$\mathbf{A}^{-1} = (\mathbf{h}_R^T \mathbf{h}_R + \mathbf{L})^{-1}$$

Elemen matriks  $\mathbf{L}$  adalah semua yang bernilai nol kecuali diagonalnya sebagai parameter regulasi dan  $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_p]$  adalah nilai duga vektor prediksi. Hasil solusinya dinyatakan dengan persamaan berikut :

$$\hat{\mathbf{w}} = \mathbf{A}^{-1} \mathbf{h}_R^T \hat{\mathbf{y}}$$

Dengan  $\hat{\mathbf{w}} = [\hat{w}_1, \hat{w}_2, \dots, \hat{w}_p]$  adalah vektor pembobot.

(Purnomo dan Kurniawan, 2006).

### 2.5.3 Metode Pelatihan RBFNN

Proses pelatihan RBF dilakukan untuk perbaikan penimbang yang menghubungkan unit-unit masukan dengan lapis tersembunyi dan unit-unit lapis tersembunyi dengan unit-unit keluaran dari JST. Lapis yang berbeda dari suatu RBF mengerjakan tugas yang berbeda dan oleh karena optimasi lapis tersembunyi dan lapis keluaran dari jaringan dipisahkan dengan memakai teknik yang tidak sama. Perbedaan strategi pelatihan yang akan dijelaskan dalam merancang suatu RBF, dan tergantung pada bagaimana pusat-pusat RBF dari jaringan dispesifikan. Tiga macam pendekatan yang akan dijelaskan sebagai dasar teori untuk melakukan pelatihan adalah sebagai berikut :

#### 1. Seleksi Titik Pusat Secara Random

Pendekatan yang pertama dengan mengasumsikan fungsi-fungsi aktivasi dari unit-unit lapis tersembunyi adalah tetap/*fixed*. Secara khusus lokasi-lokasi dari pusat RBF dipilih secara random dari himpunan data pelatihan dengan menggunakan suatu fungsi *Gauss* yang sama sebagai standar deviasinya yang tepat dalam mengikuti penyebaran dari nilai rata-rata. Spesifiknya suatu RBF ternormalisasi dengan nilai rata-rata di  $\mu$  dan didefinisikan sebagai berikut :

$$H(\|x_R - \mu_m\|^2) = \exp\left(-\frac{M}{d^2} \|x_R - \mu_m\|^2\right)$$

dengan

$M$  = jumlah dari nilai rata-rata

$d$  = jarak maximum di antara nilai rata-rata yang dipilih

$m=1, 2, \dots, p$

standar deviasi (*width* dari RBF *Gauss fixed*)

$$\sigma = \frac{d}{\sqrt{2M}}$$

Sebagai pilihan standar deviasi  $\sigma$  yang sesuai, dihindari dua kondisi ekstrim sehingga fungsi-fungsi *Gauss* terlalu tajam atau landai. Parameter yang dibutuhkan untuk dilatihkan dalam pendekatan ini hanya pembobot linier pada lapis keluaran dari jaringan. Metode yang digunakan yaitu dengan menggunakan metode *pseudoinvers* dengan persamaan :

$$W = h_R^{-1} D$$

$D$  adalah vektor tanggapan yang dikehendaki dalam himpunan data pelatihan,  $h_R^{-1}$  adalah *pseudoinvers* dari matriks  $h_R$  yang didefinisikan sebagai berikut :

$$h_R = \{H\}$$

dengan  $H = \exp\left(-\frac{M}{d^2} \|x_R - \mu_m\|^2\right)$

Dasar dari semua algoritma untuk komputasi *pseudoinvers* dari suatu matriks adalah *singular value decomposition*.

## 2. Titik Pusat Diseleksi dengan Metode Pelatihan Mandiri

Pendekatan yang kedua, fungsi-fungsi radial basis yang diijinkan untuk memindahkan lokasi-lokasi dari rata-rata yang terorganisasi mandiri, di mana pembobot-pembobot linier dari lapis keluaran dihitung menggunakan aturan pelatihan terbimbing. Jaringan mengalami proses pelatihan hibrida. Proses pelatihan pada komponen terorganisasi mandiri menempatkan nilai rata-rata dari RBF hanya dalam daerah-daerah dari ruang masukan/*input space* di mana data yang signifikan muncul. RBFNN adalah jaringan dan pelatihan hibrida yang mengkombinasi paradigma tak terbimbing dan skema pelatihan terbimbing, di mana sebagian dari pembobot ditentukan dengan pelatihan terbimbing dan sebagian lain diperoleh dari pelatihan tak terbimbing. RBFN dilatih dengan aturan tak terbimbing pada lapis masukan dan aturan pelatihan terbimbing pada lapis keluaran.

## 3. Titik Pusat Diseleksi Secara Terbimbing

Pendekatan ketiga, pusat-pusat dari RBF dan semua parameter-parameter bebas dari jaringan mengalami suatu proses pelatihan

terbimbing. Langkah pertama dalam pengembangan suatu proses pelatihan adalah mendefinisikan nilai dari fungsi :

$$E = \frac{1}{2} \sum_{m=1}^p e_m^2$$

di mana  $p$  adalah jumlah sampel pelatihan yang digunakan dalam proses pelatihan, dan  $e$  adalah sinyal *error*. Parameter yang dibutuhkan adalah menemukan parameter-parameter bebas,  $w_m$ ,  $\mu_m$ , dan  $\sum_m^{-1}$  (dihubungkan ke normalisasi matriks-pembobot), yang meminimisasi E :

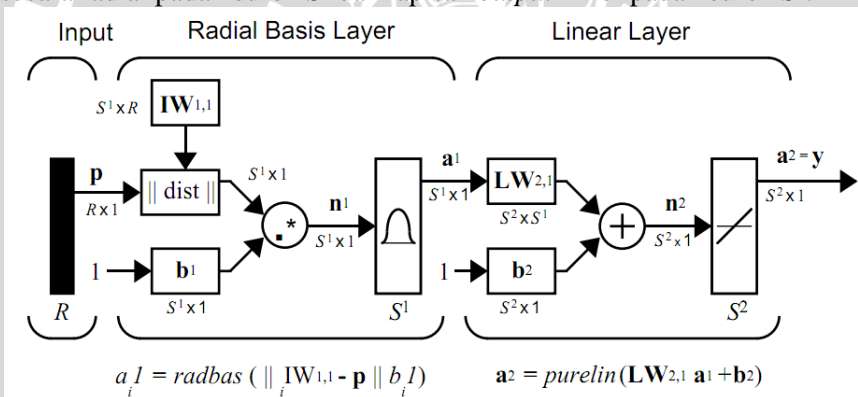
$$e_m = d_m - \hat{y}_m$$

$$e_m = d_m - \sum_{m=1}^p w_m H(\|x_R - \mu_m\|) \quad (2.24)$$

(Purnomo dan Kurniawan, 2006).

### 2.5.4 Metode Pelatihan RBFNN di Matlab

RBFNN terdiri dari dua lapisan yaitu lapisan tersembunyi secara radial pada neuron  $S^1$  dan lapisan *output* linier pada neuron  $S^2$ .



Gambar 2.10 Arsitektur RBFNN di Matlab (Demuth dan Mark, 2000).

$a_i^1$  adalah elemen ke- $i$  dari  $a^1$  dengan  $iW_{1,1}$  adalah vektor yang terbuat dari baris ke- $i$  dari  $IW_{1,1}$ .

dengan

$R$  = jumlah elemen dalam vektor *input*

$S^1$  = jumlah neuron pada lapisan *input*

$S^2$  = jumlah neuron pada lapisan *output*

Besar jarak pada kotak  $\|dist\|$  dalam Gambar 2.10 menerima vektor *input* ( $P$ ) dan matriks bobot *input*  $IW_{1,1}$  dan menghasilkan vektor yang memiliki elemen  $S^1$ . Unsur-unsur jarak antara vektor *input* dan vektor  $iW_{1,1}$  dibentuk dari baris-baris pada matriks bobot *input*. Bias vektor  $b^1$

dan vektor *output* dari nilai jarak  $\|dist\|$  digabungkan dengan operasi MATLAB `.*` yang berarti perkalian antara elemen dengan elemen. *Output* dari lapisan pertama untuk jaringan *feed forward* dapat diperoleh dengan kode berikut :

$$a\{1\} = radbas(netprod(dist(net.IW\{1,1\}, P), net.b\{1\}))$$

Baris kode `a{1}` tidak harus ditulis akan tetapi semua rincian dalam merancang jaringan ini dibangun ke fungsi desain *newrbe* dan *newrb*, dan *output-outputnya* dapat diperoleh dengan fungsi *sim*. Jaringan ini berperilaku dengan mengikuti vektor *input* (P) melalui jaringan *output* ke  $a^2$ . Jika disajikan vektor *input* ke jaringan seperti itu, maka setiap neuron pada lapisan RBF akan menghasilkan nilai yang sesuai dengan seberapa dekat vektor *input* ke vektor bobot masing-masing neuron. Dengan demikian, neuron RBF dengan vektor bobot sangat berbeda dari vektor input (P) dan memiliki *output* mendekati nol. *Output* kecil ini hanya memiliki efek yang dapat diabaikan pada neuron *output* linier. Sebaliknya, neuron RBF dengan vektor bobot dekat dengan vektor *input* (P) yang menghasilkan nilai mendekati 1. Jika sebuah neuron memiliki *output* 1, maka bobot *output* pada lapisan kedua melewati nilai-nilai mereka ke neuron secara linier pada lapisan kedua. Bahkan, jika hanya satu neuron RBF yang memiliki *output* 1 dan semua neuron yang lain memiliki *output* dari 0 (atau sangat dekat dengan 0), maka *output* dari lapisan linier akan mengaktifkan bobot-bobot *output* neuron tersebut. Bagaimanapun kasus ini akan menjadi ekstrim. Biasanya beberapa neuron selalu terhubung untuk derajat yang bervariasi. Operasi pada lapisan pertama secara detail dapat dijelaskan seperti berikut. Setiap bobot *input* pada neuron adalah jarak antara vektor input dan vektor bobot yang dihitung dengan *dist*. Masing-masing neuron *input* adalah perkalian antara elemen dengan elemen dari bobot *input* dan bobot biasanya yang dihitung dengan *netprod*. Setiap neuron *output* adalah *input* jaringan yang dilewatkan melalui *radbas*. Jika sebuah neuron vektor bobot adalah sama dengan vektor *input* (*transpose*), maka bobot inputnya adalah 0. *Input* jaringannya adalah 0 dan nilai *outputnya* adalah 1. Jika vektor bobot neuron adalah jarak *spread* dari vektor *input*, maka bobot *input* adalah *spread* dengan *input* jaringannya adalah  $\sqrt{-\log(0,5)}$  atau 0,8326) sehingga nilai *outputnya* adalah 0,5.

Fungsi iterasi *newrb* menciptakan satu neuron RBFNN pada suatu waktu. Neuron-neuron ditambahkan ke jaringan sampai jumlah



kuadrat kesalahan jatuh di bawah tujuan kesalahan atau jumlah maksimum neuron telah tercapai. Panggilan untuk fungsi ini adalah :

$$net = newrb(P, T, Goal, Spread)$$

Fungsi *newrb* ini mengambil matriks *input* (P), vektor *output* (T), parameter desain *Goal* dan *Spread*, dan mengembalikan jaringan yang diinginkan. Metode desain *newrb* mirip dengan fungsi *newrbe*. Perbedaannya adalah bahwa *newrb* menciptakan neuron satu per satu pada setiap iterasi vektor *input* yang menghasilkan kesalahan jaringan yang paling kecil, digunakan untuk membuat neuron *radbas*. Kesalahan jaringan baru diperiksa dan jika fungsi *newrb* cukup kecil maka fungsi *newrb* telah selesai, sebaliknya neuron berikutnya ditambahkan. Prosedur ini diulang sampai tujuan *error* terpenuhi atau jumlah maksimum neuron tercapai (Demuth dan Mark, 2000).

## 2.6 Prosedur Klasifikasi

Evaluasi prosedur klasifikasi adalah suatu evaluasi yang melihat peluang kesalahan klasifikasi yang dilakukan oleh suatu fungsi klasifikasi. Ukuran yang dipakai adalah *Hit Ratio* dan *apparent error rate* (APER). Nilai *Hit Ratio* menyatakan nilai proporsi sampel yang benar diklasifikasi oleh fungsi klasifikasi dan nilai APER menyatakan nilai proporsi sampel yang salah diklasifikasikan oleh fungsi klasifikasi (Johnson dan Wichern, 1992). Penentuan klasifikasi dapat diketahui melalui Tabel 2.1 :

Tabel 2.1 Tabel Ketepatan Klasifikasi Johnson dan Wichern (1992)

<i>Actual Membership</i>	<i>Predicted Membership</i>		<i>Total</i>
	$\hat{\pi}_1$	$\hat{\pi}_2$	
$\pi_1$	$n_{11}$	$n_{12}$	A
$\pi_2$	$n_{21}$	$n_{22}$	B
<i>Total</i>	C	D	E

Keterangan :

- $n_{11}$  = Jumlah  $y_p$  dari  $\pi_1$  tepat diklasifikasikan sebagai  $\hat{\pi}_1$
- $n_{12}$  = Jumlah  $y_p$  dari  $\pi_1$  tepat diklasifikasikan sebagai  $\hat{\pi}_2$
- $n_{21}$  = Jumlah  $y_p$  dari  $\pi_2$  tepat diklasifikasikan sebagai  $\hat{\pi}_1$
- $n_{22}$  = Jumlah  $y_p$  dari  $\pi_2$  tepat diklasifikasikan sebagai  $\hat{\pi}_2$



Rumus ketepatan klasifikasi tersebut adalah sebagai berikut :

1. Ketepatan pada  $\pi_1$  (dalam %) untuk nilai *Hit Ratio* adalah  $(n_{11}/A) \times 100\%$  dan nilai APER nya adalah  $1 - [(n_{11}/A) \times 100\%]$
2. Ketepatan pada  $\pi_2$  (dalam %) untuk nilai *Hit Ratio* adalah  $(n_{22}/B) \times 100\%$  dan nilai APER nya adalah  $1 - [(n_{22}/B) \times 100\%]$
3. Sedangkan untuk melihat ketepatan secara keseluruhan untuk nilai *Hit Ratio* adalah  $[(n_{11} + n_{22})/E] \times 100\%$ , dan nilai APER secara keseluruhan adalah  $1 - [[(n_{11} + n_{22})/E] \times 100\%]$ .

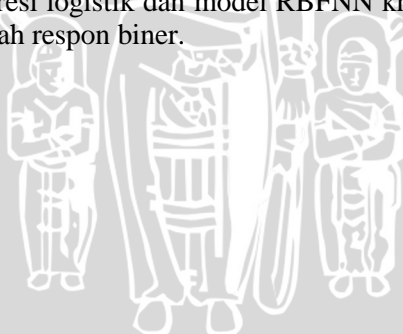
## 2.7 Perbandingan Algoritma Klasifikasi

Pada proses klasifikasi terdapat banyak algoritma klasifikasi yang dapat digunakan. Suatu algoritma yang dianggap paling baik pada proses klasifikasi suatu permasalahan belum tentu baik juga untuk memecahkan permasalahan yang lain. Beberapa algoritma klasifikasi dengan prediksi yang cukup signifikan adalah regresi logistik, Jaringan syaraf tiruan (JST), C5.0, *Classification via Regression*, dan *LogitBoost* (Saravanan and Malika, 2009). Metode statistika regresi logistik tidak memerlukan asumsi kenormalan baik pada sebaran galat maupun pada peubah respon, dan hubungan antara peubah respon dengan peubah bebas tidak perlu diasumsikan linier. Namun terdapat asumsi yang diberlakukan yaitu ketiadaan multikolinieritas antar peubah bebas (Hosmer dan Lemeshow, 1989).

Jaringan syaraf tiruan (JST) terdiri dari 6 metode, yaitu *Dynamic*, *Prune*, *Exhaustive Prune*, *Quick*, *RBFN* dan *Multiple* (Saravanan and Malika, 2009). JST telah banyak dan berhasil dilakukan oleh para ahli dibidangnya dalam berbagai aplikasi, metode pelatihan, dan model JST. Salah satunya adalah metode pelatihan hibrida. Semakin tinggi tingkat ketidaklinieran suatu sistem, seringkali sulit diselesaikan dengan metode pelatihan terbimbing dan tak terbimbing. Banyak peneliti yang mencoba menggabungkannya dan diperoleh hasil yang lebih baik dengan salah satu metode hibrida yaitu metode *Radial Basis Function Neural Network* (RBFNN). RBFNN merupakan salah satu jaringan syaraf yang dikenal dalam permasalahan regresi dan klasifikasi data (Du dan Swamy, 2006). Tahun 1996 oleh Sanchez menggunakan JST untuk klasifikasi dan pemodelan. Pada penelitian Sanchez dikaji klasifikasi dari dua metode JST yaitu *Multi Layer Feed Forward* (MLF)

dan RBFNN mengenai kemampuan pemodelan, stabilitas dan reproduktifitas jaringan berdasarkan berbagai *training* pada set data yang sama dengan yang telah ditetapkan nilai untuk kepekaan dan kekhususan. Hasil kedua metode cukup baik untuk mereproduksi ruang *input* namun, untuk RBFNN karena pendekatan lokal, studi menunjukkan beberapa *property* terkait dengan kepekaan dan kekhususan yang relevan dalam masalah-masalah praktis. Pada tahun 2003 oleh Gupta, Jin, dan Homma memperkenalkan model baru dari *Feed forward* Lapis Banyak atau *Multilayer Feed forward Neural Network* (MFNN) yaitu *Radial Basis Function Neural Network* (RBFNN), di mana jaringan ini memiliki struktur tiga lapis dan dalam proses pelatihannya lebih cepat dibanding *Perceptron* Lapis Banyak atau *Multilayer Perceptron* (MLP).

Pada tahun 2005, Sarimveis, Doganis, dan Alexandridis memperkenalkan teknik klasifikasi dengan menggunakan metode RBFNN dan (FNN). Kedua metode ini diaplikasikan pada kasus tingkat kelembutan *tissue*. Pada penelitian Sarimveis diperoleh hasil bahwa metode RBFNN lebih akurat jika dibandingkan dengan FNN. Pada penelitian ini ingin dibandingkan yang paling baik antara metode statistika regresi logistik dan RBFNN, karena pada penelitian Saravanan and Malika tahun 2009 di mana kedua algoritma klasifikasi tersebut sama-sama memiliki nilai akurasi yang signifikan. Perbandingan akan diteliti pada model regresi logistik dan model RBFNN khususnya untuk pengklasifikasian peubah respon biner.



## BAB III METODOLOGI

### 3.1 Sumber Data

Data yang digunakan adalah data sekunder yang diperoleh dari skripsi, untuk lebih jelas dapat dilihat pada Tabel 3.1.

Tabel 3.1 Sumber Data

Data	Sumber Data	Peubah Respon (Y)	Banyak dan Jenis Peubah Bebas (X)	Jumlah Data
Data 1	Skripsi Santoso (2012)	0 = pelanggaran ringan 1 = pelanggaran tidak ringan	6 (kategorikal)	137
Data 2	Skripsi Novianti (2013)	0 = kanker jinak 1 = kanker ganas	4 (kategorikal)	159
Data 3	Skripsi Arifanjuni (2013)	0 = pernah didiagnose menderita penyakit DB 1 = tidak pernah didiagnose menderita penyakit DB	5 (kategorikal)	150

Keterangan pada Tabel 3.1 mengenai sumber data dapat dilihat pada Lampiran 1, Lampiran 2, dan Lampiran 3.

### 3.2 Metode Penelitian

Langkah-langkah penelitian yang dilakukan adalah sebagai berikut :

- I. Membagi data menjadi dua bagian yaitu data pelatihan atau pembentukan model (*training*) dan *testing* atau validasi. Pada data 70% sebagai data *training* dan 30% sebagai data *testing* atau validasi.
- II. Tahapan untuk pembentukan model pada analisis regresi logistik :

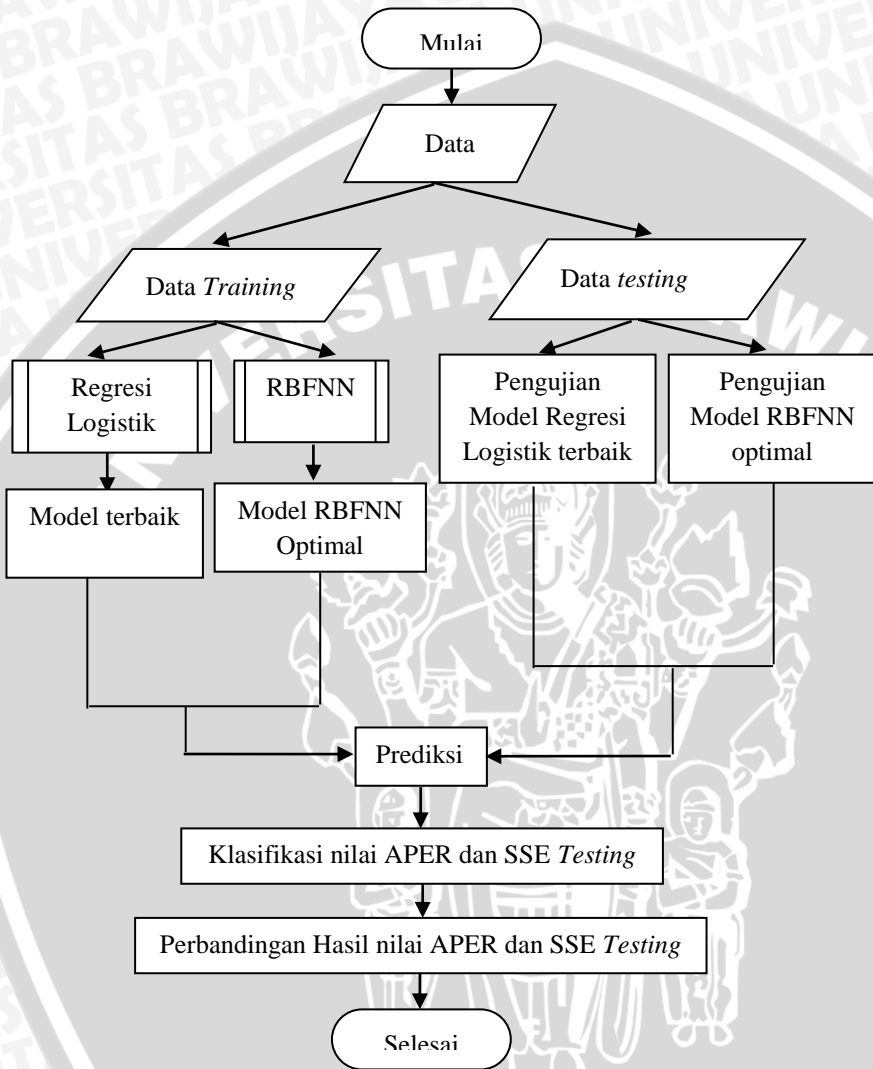
1. Memeriksa ada tidaknya multikolinieritas antara peubah bebas X pada persamaan (2.11).
  2. Pendugaan parameter pada persamaan (2.14).
  3. Pengujian parameter secara serentak pada persamaan (2.19).
  4. Pengujian parameter secara parsial pada persamaan (2.18).
  5. Pembentukan model regresi logistik pada persamaan (2.10).
  6. Pemeriksaan kesesuaian model yg terbentuk pada persamaan (2.20).
  7. Membentuk tabel ketepatan klasifikasi model *training* dan *testing* seperti pada Tabel 2.1.
- III. Langkah untuk pembentukan model RBFNN adalah :
1. Menyiapkan 70% data *training* dan 30% data *testing* yang telah ditentukan.
  2. Menentukan banyaknya peubah bebas (*input*) dan peubah respon (*output*). Peubah *input* dan peubah *output* untuk model ANN akan mengikuti banyaknya peubah bebas dan peubah respon pada model regresi logistik.
  3. Mencari arsitektur jaringan syaraf tiruan (JST) yang optimal dengan cara *trial* dan *error* sampai didapatkan nilai SSE yang paling minimum pada Sub bab 2.5.4.
  4. Menghitung keluaran tiap fungsi basis (fungsi aktivasi *Gaussian*) pada persamaan (2.22).
  5. Menentukan bobot lapisan dan bobot bias lapisan dengan metode *least square* pada Sub bab 2.5.2.
  6. Menghitung *output* jaringan dengan simulasi atau memprediksi kelas dari masing-masing grup sehingga kesalahan pengklasifikasian dapat dikalkulasi pada Sub bab 2.5.2 dan Sub bab 2.5.4.
  7. Menghitung kesalahan (*error*) antara keluaran (*target*) dengan keluaran RBF pada persamaan (2.24).
  8. Melihat ketepatan/akurasi pengklasifikasian antara data *training* dan *testing* pada Tabel 2.1.
- IV. Melakukan pemilihan metode terbaik dari model regresi logistik dan model RBFNN berdasarkan nilai APER dan SSE data *testing*.

*Software* statistika yang digunakan untuk analisis regresi logistik adalah IBM SPSS 19 dan analisis RBFNN dengan Matlab 2008a. Diagram alir penelitian ini disajikan pada Gambar 3.1 – 3.3 :

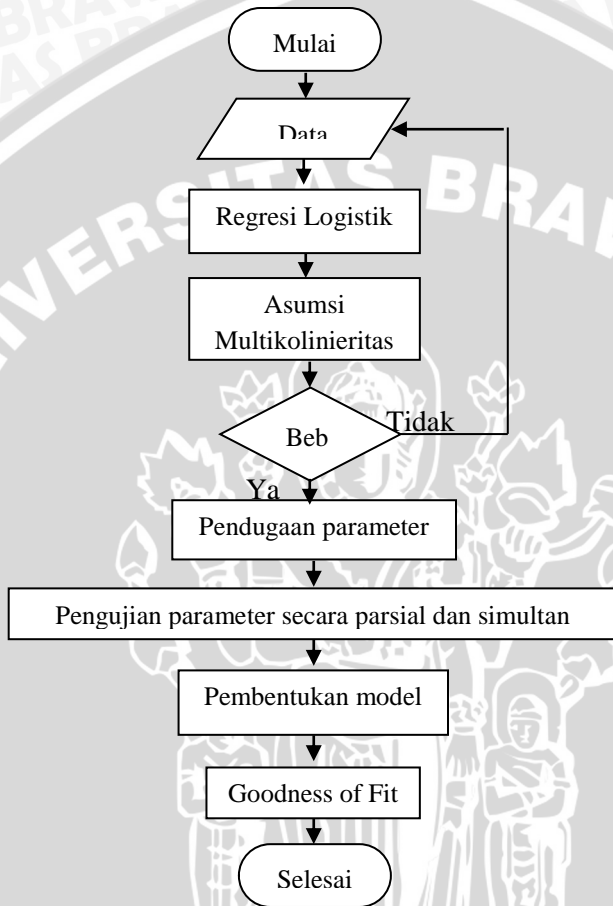
UNIVERSITAS BRAWIJAYA



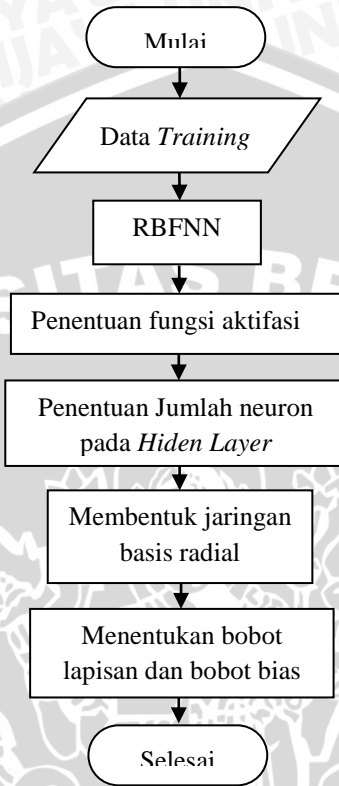




Gambar 3.1 Diagram Alir Penelitian



Gambar 3.2 Diagram Alir Model Regresi Logistik



Gambar 3.3 Diagram Alir Model RBFNN

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Analisis Regresi Logistik Biner

#### 4.1.1 Asumsi Multikolinieritas

Apabila dalam suatu model regresi terdapat korelasi antara dua peubah atau lebih penjelas, maka dapat mengacaukan hubungan antara peubah bebas terhadap peubah respon, sehingga penduga yang dihasilkan akan bersifat bias dan memiliki ragam yang mendekati tak hingga (Sembiring, 1995), untuk memilih model regresi terbaik harus bebas dari multikolinieritas diantara peubah bebasnya. Berikut hipotesisnya :

$H_0 : \rho_{x_j, x_j'} = 0$  (tidak ada hubungan antar kedua peubah)

$H_1 : \rho_{x_j, x_j'} \neq 0$  (ada hubungan antar kedua peubah)

Tabel 4.1 Hasil Korelasi *Rank Spearman* pada Data 1

	X1	X2	X3	X4	X5	X6
X1	1	-0,024	-0,136	-0,179	-0,141	-0,141
X2	0,024	1	0,077	-0,49	-0,111	-0,111
X3	-0,136	0,077	1	-0,101	-0,049	-0,049
X4	-0,179	-0,049	-0,101	1	0,131	0,131
X5	-0,141	-0,111	-0,049	0,131	1	0,232
X6	-0,141	-0,111	-0,049	0,131	0,232	1

Pada Tabel 4.1 untuk data 1 yaitu jenis pelanggaran terima  $H_0$  atau bebas dari multikolinieritas karena nilai korelasi *rank spearman* ada dalam kisaran -0,7 dan 0,7 (Lind, Robert, and William, 2000), sehingga tidak terdapat korelasi yang kuat antara peubah bebas X yaitu pekerjaan pelanggar, kendaraan pelanggar, usia, hari pelanggaran, waktu pelanggaran, dan wilayah hukum.

Tabel 4.2 Hasil Korelasi *Rank Spearman* pada Data 2

	X1	X2	X3	X4
X1	1	0,028	-0,161	-0,041
X2	0,028	1	0,284	0,069
X3	-0,161	0,284	1	0,038
X4	-0,041	0,069	0,038	1

Pada Tabel 4.2 untuk data 2 yaitu diagnosis kanker payudara bebas dari multikolinieritas atau terima  $H_0$  maka tidak terdapat korelasi

yang kuat antara peubah bebas X yaitu *intermediate findings*, kategori BIRAD-S, *Suspicious for Malignancy*, dan Letak abnormal.

Tabel 4.3 Hasil Korelasi Rank Spearman pada Data 3

	X1	X2	X3	X4	X5	X6
X1	1	0,270	0,520	-0,003	-0,323	0,271
X2	0,027	1	-0,021	-0,036	0,015	-0,082
X3	0,520	0,021	1	-0,055	-0,560	0,541
X4	-0,003	-0,036	-0,055	1	-0,032	-0,102
X5	-0,323	0,015	-0,563	-0,032	1	-0,338
X6	0,271	-0,082	0,541	-0,102	-0,338	1

Pada Tabel 4.3 untuk data 3 mengenai diagnose demam berdarah Ho di terima atau bebas dari multikolinieritas sehingga tidak terdapat korelasi yang kuat antara peubah bebas X yaitu jenis kelamin, pendidikan, tipe daerah, tidur dengan kelambu, sarana menampung air minum, dan penampungan air limbah.

#### 4.1.2 Pengujian secara Simultan/Serentak

Pengujian terhadap penduga parameter secara serentak dilakukan untuk melihat apakah ada peubah yang berpengaruh terhadap model dengan uji signifikansi menggunakan statistik uji rasio *likelihood* (G). Hipotesisnya adalah :

$$H_0 : \beta_1 = \beta_2 = \dots = \beta_j = 0$$

$$H_1 : \text{paling sedikit ada satu } \beta_j \neq 0$$

Hasil pada Lampiran 4 menunjukkan bahwa untuk data 1, data 2, dan data 3 Ho ditolak karena nilai *p-value* kurang dari 0,05 yaitu 0,009; 0,000; dan 0,000. Hal ini berarti bahwa minimal ada satu peubah bebas yang berpengaruh terhadap peubah respon dengan nilai statistik uji G yang diperoleh dari model sebesar 21,858 dengan derajat bebas sebesar 9; 59,510 dengan derajat bebas sebesar 9; dan 133,730 dengan derajat bebas sebesar 16. Model penuh yang terbentuk adalah :

$$g[\pi(X_1)] = 0,092 + 1,174X_{11} - 1,186X_{21} + 0,874X_{31} + 0,129X_{32} \\ + 0,458X_{33} + 0,966X_{41} - 0,448X_{51} - 0,367X_{52} - 0,153X_{53} \\ - 0,448X_{61} - 0,367X_{62} - 0,153X_{63}$$

$$g[\pi(X_2)] = 0,918 + 1,117X_{11} + 0,438X_{12} - 1,598X_{24} + 0,580X_{25} \\ - 3,075X_{32} - 2,053X_{33} - 1,200X_{35} - 1,509X_{37} - 1,780X_{41}$$

$$g[\pi(X_3)] = -3,429 + 1,599X_{11} - 1,267X_{21} + 0,377X_{22} - 1,298X_{23} \\ - 0,514X_{24} - 1,042X_{25} + 2,712X_{31} + 1,327X_{41} + 1,059X_{42}$$



$$-0,03X_{51} + 0,28X_{52} + 1,867X_{61} - 2,144X_{62} - 0,995X_{63} + 1,34X_{64}$$

#### 4.1.3 Pengujian secara Individu/Parsial

Menentukan peubah bebas yang signifikan terhadap peubah respon yaitu mengetahui peubah apa saja yang berpengaruh terhadap respon, perlu dilakukan uji signifikansi parameter pada masing-masing variabel menggunakan statistik uji *Wald* ( $W_j$ ). Hipotesis yang digunakan adalah :

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0$$

Statistik uji wald berdistribusi *chi-square* dengan derajat bebas satu dan akan dibandingkan dengan nilai  $\chi^2$  tabel sesuai derajat bebas data. Keputusan untuk menolak  $H_0$  diambil apabila nilai  $W_j > \chi^2_{(tabel)}$  atau apabila nilai *p-value*  $< 0,05$ . Nilai statistik dan taksiran parameter yang diperoleh dari model logistik secara individu disajikan pada Tabel 4.4-4.6 :

Tabel 4.4 Hasil Uji Wald pada Data 1

Predictor	B	S. E.	Wald	df	<i>p-value</i>
Constant	0,092	0,827	0,012	1	0,912
X1 (1)	1,174	0,446	6,919	1	0,009
X2 (1)	-1,186	0,488	5,903	1	0,015
X3			2,221	3	0,528
X3 (1)	0,874	0,683	1,637	1	0,201
X3 (2)	0,129	0,563	0,053	1	0,819
X3 (3)	0,458	0,610	0,563	1	0,453
X4 (1)	0,966	0,418	5,351	1	0,021
X5				3	0,881
X5 (1)	-0,448	0,602	0,553	1	0,457
X5 (2)	-0,367	0,610	0,363	1	0,547
X5 (3)	-0,153	0,702	0,047	1	0,827
X6				3	0,881
X6 (1)	-0,448	0,602	0,553	1	0,457
X6 (2)	-0,367	0,610	0,363	1	0,547
X6 (3)	-0,153	0,702	0,047	1	0,827

Berdasarkan nilai *p-value* pada Tabel 4.4 diketahui bahwa peubah X3 (usia), X5 (waktu pelanggaran), dan X6 (wilayah hukum)

tidak signifikan pada penentuan klasifikasi jenis pelanggaran lalu lintas atau terima  $H_0$  dan peubah yang berpengaruh terhadap respon atau tolak  $H_0$  yaitu peubah  $X_1$  (pekerjaan pelanggar),  $X_2$  (kendaraan pelanggar), dan  $X_4$  (hari pelanggaran). Sehingga model yang terbentuk adalah :

$$g[\pi(X_1)] = 0,092 + 1,174X_{11} - 1,186X_{21} + 0,966X_{41}$$

Tabel 4.5 Hasil Uji Wald pada Data 2

Predictor	B	S. E.	Wald	df	p-value
Constant	0,918	0,875	1,102	1	0,294
X1			2,912	2	0,233
X1 (1)	1,117	0,867	1,660	1	0,198
X1 (2)	0,438	0,914	0,230	1	0,632
X2			8,823	2	0,012
X2 (4)	-1,598	0,635	6,332	1	0,012
X2 (5)	0,580	0,584	0,985	1	0,321
X3			18,501	4	0,001
X3 (2)	-3,075	0,872	12,429	1	0,000
X3 (3)	-2,053	0,756	7,379	1	0,007
X3 (5)	-1,200	1,242	0,933	1	0,334
X3 (7)	-1,509	0,565	7,143	1	0,008
X4 (1)	-1,780	0,507	12,308	1	0,000

Pada Tabel 4.5 diketahui bahwa peubah yang berpengaruh terhadap respon atau tolak  $H_0$  yaitu peubah  $X_2$  (kategori BIRD-S),  $X_3$  (*suspicious for Malignancy*), dan  $X_4$  (letak abnormal). Sehingga model yang terbentuk adalah :

$$g[\pi(X_2)] = 0,918 - 1,598X_{24} + 0,580X_{25} - 3,075X_{32} - 2,053X_{33} - 1,200X_{35} - 1,509X_{37} - 1,780X_{41}$$

Tabel 4.6 Hasil Uji Wald pada Data 3

Predictor	B	S. E.	Wald	df	p-value
Constant	-3,429	2,358	2,115	1	0,146
X1 (1)	1,599	0,737	4,709	1	0,030
X2			2,087	5	0,837
X2 (1)	-1,267	1,846	0,472	1	0,492
X2 (2)	0,377	2,066	0,033	1	0,855
X2 (3)	-1,298	1,447	0,805	1	0,370
X2 (4)	-0,514	1,481	0,120	1	0,729
X2 (5)	-1,042	1,475	0,499	1	0,480
X3(1)	2,712	1,044	6,747	1	0,009

X4			1,094	2	0,579
X4 (1)	1,327	1,276	1,081	1	0,298
X4 (2)	1,059	1,309	0,655	1	0,418
X5			0,044	2	0,978
X5 (1)	-0,030	0,990	0,002	1	0,966
X5 (2)	0,280	1,414	0,039	1	0,843
X6			9,150	4	0,057
X6 (1)	1,867	1,202	2,412	1	0,120
X6 (2)	-2,144	1,496	2,053	1	0,152
X6 (3)	-0,995	1,651	0,363	1	0,547
X6 (4)	1,340	1,274	1,105	1	0,293

Pada Tabel 4.6 diketahui bahwa hanya peubah X1 ( jenis kelamin) dan X3 ( tipe daerah) yang signifikan berpengaruh terhadap respon pada penentuan klasifikasi seseorang yang pernah terdiagnose menderita penyakit demam berdarah atau tolak Ho. Sehingga model yang terbentuk adalah :

$$g[\pi(\mathbf{X}_3)] = - 3,429 + 1,599X_{11} + 2,712X_{31}$$

#### 4.1.4 Pembentukan Model Regresi Logistik Biner untuk Variabel yang Signifikan

Pada tahapan ini akan dilakukan pemodelan dengan menggunakan peubah bebas yang signifikan terhadap peubah respon. Hasil perhitungan uji statistik dan taksiran parameter disajikan pada tabel berikut :

Tabel 4.7 Taksiran Parameter Model Regresi Logistik Biner Signifikan pada Data 1

Predictor	B	S. E.	Wald	<i>p-value</i>
Constant	0,084	0,557	0,023	0,880
X1 (1)	1,198	0,425	7,944	0,005
X2 (1)	-1,120	0,461	5,904	0,015
X4 (1)	0,889	0,397	5,010	0,025

Berdasarkan hasil (Tabel 4.7) pada taksiran parameter model regresi logistik biner yang signifikan didapatkan model reduksi sebagai berikut:

$$g[\pi(\mathbf{X}_1)] = 0,084 + 1,198X_{11} - 1,120X_{21} - 0,889X_{41}$$

Tabel 4.8 Taksiran Parameter Model Regresi Logistik Biner Signifikan pada Data 2

Predictor	B	S. E.	Wald	p-value
Constant	1,646	0,546	9,077	0,003
X2			8,179	0,017
X2 (4)	-1,430	0,611	0,5487	0,019
X2 (5)	0,611	0,567	1,160	0,282
X3			19,587	0,001
X3 (2)	-3,130	0,845	13,712	0,000
X3 (3)	-2,123	0,762	7,771	0,005
X3 (5)	-1,022	1,220	0,701	0,402
X3 (7)	-1,478	0,552	7,168	0,007
X4 (1)	-1,761	0,498	12,487	0,000

Model reduksi yang terbentuk pada Tabel 4.8 adalah :

$$g[\pi(\mathbf{X}_2)] = 1,646 - 1,430X_{24} + 0,611X_{25} - 3,130X_{32} - 2,123X_{33} - 1,022X_{35} - 1,478X_{37} - 1,761X_{41}$$

Tabel 4.9 Taksiran Parameter Model Regresi Logistik Biner Signifikan pada Data 3

Predictor	B	S. E.	Wald	p-value
Constant	-3,253	0,575	31,976	0,000
X1 (1)	1,213	0,563	4,640	0,031
X3 (1)	4,316	0,618	48,775	0,000

Pada Tabel 4.9 didapatkan model reduksi sebagai berikut:

$$g[\pi(\mathbf{X}_3)] = - 3,253 + 1,213X_{11} + 4,316X_{31}$$

#### 4.1.5 Pengujian Kelayakan Model

Pengujian kelayakan model atau *goodness of fit* digunakan untuk mengetahui apakah model yang dihasilkan sudah sesuai atau belum. Hipotesis yang digunakan dalam pengujian ini yaitu :

$H_0$  : model sesuai

$H_1$  : model tidak sesuai

Tabel 4.10 Pengujian Hosmer-Lemeshow pada Data

Data	Chi-square	df	p-value
1	3,849	6	0,697
2	5,663	8	0,685
3	7,256	3	0,064

Berdasarkan Tabel 4.10 nilai *p-value* untuk semua data baik data 1,2, dan 3 mempunyai nilai lebih dari 0,05 sehingga keputusan terima  $H_0$  dan dapat disimpulkan bahwa model regresi logistik biner yang digunakan sesuai atau telah cukup mampu menjelaskan data 1 (jenis pelanggaran lalu lintas), data 2 (diagnosis kanker payudara), dan data 3 (diagnosis yang pernah menderita penyakit demam berdarah).

#### 4.1.6 Ketepatan Klasifikasi

Model regresi logistik yang diperoleh digunakan untuk mengklasifikasikan data *testing*. Kebaikan model regresi logistik biner dalam pengklasifikasian digunakan tabel ketepatan klasifikasi dengan melihat ketepatan model dalam mengklasifikasikan data. Berikut adalah tabel ketepatan klasifikasi berdasarkan data *training*.

Tabel 4.11 Tabel Ketepatan Klasifikasi Data *Training* pada Data 1

Kategori		Y (prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y (aktual)	0	18	5	23	78,3
	1	31	42	73	57,5
Total		49	47	96	62,5

Berdasarkan Tabel 4.11 data 1, diketahui bahwa dari 23 observasi pada kategori 0 (pelanggaran ringan) ada sebanyak 18 observasi yang tepat dikategorikan / dimasukkan pada kategori 0 (pelanggaran ringan) oleh model yang terbentuk dan 5 observasi pada katgeori 1 (pelanggaran tidak ringan). Jumlah observasi dikategori 1 (pelanggaran tidak ringan) sebanyak 73, ada sebanyak 31 observasi yang tepat dikategorikan / dimasukkan pada kategori 0 (pelanggaran ringan) oleh model yang terbentuk dan 42 observasi pada kategori 1 (pelanggaran tidak ringan). Ketepatan klasifikasi metode regresi logistik biner untuk data *training* jenis pelanggaran lalu lintas adalah sebesar 62,5% dengan ketepatan klasifikasi kategori 0 (palanggaran ringan) sebesar 78,3% dan kategori 1 (pelanggaran tidak ringan) sebesar 57,5%, untuk menguji akurasi / ketepatan model regresi logistik maka model tersebut digunakan untuk mengklasifikasikan data *testing* seperti tabel berikut :



Tabel 4.12 Tabel Ketepatan Klasifikasi Data *Testing* pada Data 1

Kategori		Y (prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y (aktual)	0	0	6	6	0
	1	0	35	35	100
Total		0	41	41	<b>85,4</b>

Berdasarkan Tabel 4.12 data 1, diketahui bahwa model jenis pelanggaran lalu lintas hanya mampu menunjukkan tingkat akurasi sebesar 85,4%, akan tetapi dengan menggunakan model yang terbentuk telah mampu mengategorikan 35 observasi dengan tepat pada kategori 1 (pelanggaran tidak ringan).

Tabel 4.13 Tabel Ketepatan Klasifikasi Data *Training* pada Data 2

Kategori		Y (prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y (aktual)	0	80	12	92	97,5
	1	6	13	19	68,4
Total		86	25	111	<b>83,7</b>

Pada Tabel 4.13, dari 111 observasi ada sebanyak 80 observasi yang tepat dikategorikan pada kategori 0 (kanker jinak) oleh model yang terbentuk dan 13 observasi pada kategori 1 (kanker ganas). Ketepatan klasifikasi metode regresi logistik biner untuk data *training* diagnosis kanker payudara adalah sebesar 83,7%. Pengklasifikasian data *testing* digunakan untuk menguji ketepatan model regresi logistik yang didapat. Berikut tabel klasifikasi data *testing* :

Tabel 4.14 Tabel Ketepatan Klasifikasi Data *Testing* pada Data 2

Kategori		Y (prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y (aktual)	0	29	3	32	90,6
	1	8	8	16	50
Total		37	11	48	<b>77,1</b>

Model diagnosis kanker payudara hanya mampu menunjukkan tingkat akurasi sebesar 77,1% dengan 29 observasi yang tepat pada kategori 0 (kanker jinak) dan 8 observasi pada kategori 1 (kanker ganas).

Tabel 4.15 Tabel Ketepatan Klasifikasi Data *Training* pada Data 3

Kategori		Y (prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y (aktual)	0	43	14	57	75,4
	1	3	45	48	93,7
Total		46	59	105	83,8

Observasi sebanyak 105 data pada Tabel 4.15 memberikan ketepatan klasifikasi sebesar 83,8% dengan 43 observasi yang tepat dikategorikan 0 (pernah menderita demam berdarah) dan 45 observasi pada kategori 1 (tidak pernah menderita demam berdarah). Pengujian ketepatan model regresi logistik yang didapat, diterapkan pada data *testing* seperti tabel berikut :

Tabel 4.16 Tabel Ketepatan Klasifikasi Data *Testing* pada Data 3

Kategori		Y		Total	Ketepatan Klasifikasi
		0	1		
Y	0	8	16	24	33,3
	1	0	21	21	100
Total		8	37	45	64,4

Berdasarkan Tabel 4.16, diketahui bahwa model diagnosis yang pernah penderita penyakit demam berdarah hanya mampu menunjukkan tingkat akurasi yang tinggi yaitu sebesar 64,4%.

## 4.2 *Artificial Neural Network (Jaringan Syaraf Tiruan)*

### 4.2.1 *Model Radial Basis Function Neural Network*

Arsitektur jaringan syaraf tiruan (JST) atau *neural network* fungsi radial basis terdiri dari tiga lapisan yaitu lapisan *input*, *hidden layer*, dan lapisan *output*. Pada RBFNN standar, apabila terdapat n buah observasi pada lapisan *input*, maka jumlah node pada *hidden layer* juga berjumlah n buah. Pada perkembangannya jumlah node pada *hidden layer* bisa kurang dari banyaknya observasi. Model fungsi aktivasi *Gaussian* adalah model fungsi aktivasi RBF beserta nilai taksiran parameter *hidden layer* pada lapisan *output* yang berasal dari taksiran matriks *Gaussian* ( $\mathbf{h}_R$ ) dengan menggunakan nilai *mean* dan standar deviasi yang didapatkan dengan parameter/bobot lapisan *output* menggunakan metode OLS. Penentuan model terbaik dilakukan dengan cara mencari nilai SSE terkecil, yaitu melalui uji coba dengan nilai *epoch*/iterasi yang maksimum. Berikut hasil nilai SSE pada data *training* dan *testing*:

Tabel 4.17 Daftar Jumlah Neuron Lapisan Tersembunyi dan Nilai SSE

Data 1			Data 2		
Jumlah Neuron	SSE Training	SSE Testing	Jumlah Neuron	SSE Training	SSE Testing
6	20,544	7,226	4	14,097	9,938
10	19,392	14,183	10	9,546	9,039
25	14,304	18,315	25	6,771	7,303
50	9,504	24,179	50	6,438	7,381
75	9,120	27,797	75	6,438	7,371
94	9,024	15,104	109	6,438	8,400

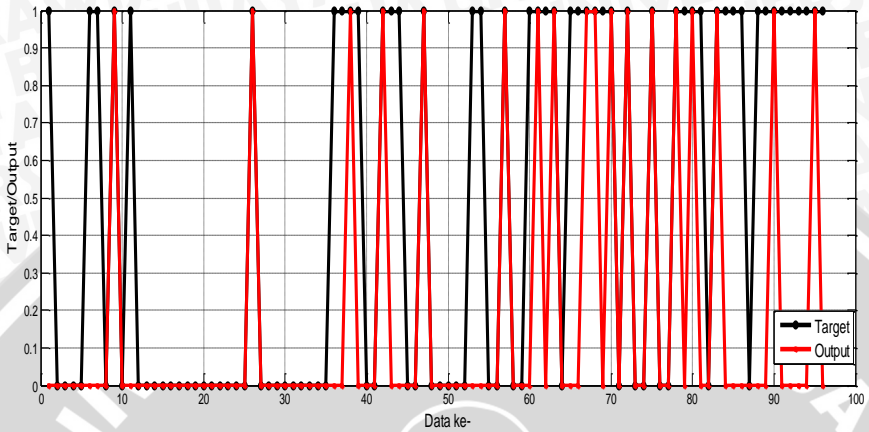
  

Data 3		
Jumlah Neuron	SSE Training	SSE Testing
6	21,210	10,082
10	19,530	10,113
25	13,020	9,173
50	2,940	10,353
75	3,045	10,276
103	3,045	17,283

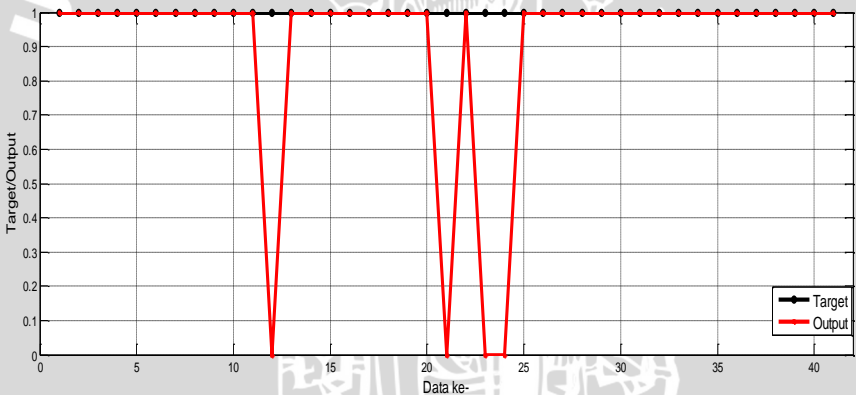
Berdasarkan hasil percobaan pada Tabel 4.17, hasil nilai SSE *testing* terkecil ada pada neuron ke 6 untuk data 1 (pelanggaran lalu lintas), neuron ke 25 untuk data 2 (diagnosis kanker payudara), dan neuron ke 25 untuk data 3 (diagnosis yang pernah menderita penyakit demam berdarah). Hasil tersebut menunjukkan bahwa besar pemilihan *epoch* berpengaruh terhadap tingkat *error* yang dihasilkan pada RBFNN, sehingga model arsitektur jaringan yang terbentuk untuk masing-masing data yaitu 6-6-1, 4-25-1, dan 6-25-1. Hasil prediksi klasifikasi dengan model RBFNN dapat dilihat pada Lampiran 6, Lampiran 7, dan Lampiran 8.

#### 4.2.2 Grafik Target dan Output

Pada klasifikasi peubah respon Y atau target akan ditampilkan pada grafik berikut beserta hasil klasifikasi keluaran (*output*) untuk masing-masing data *training* dan *testing*.

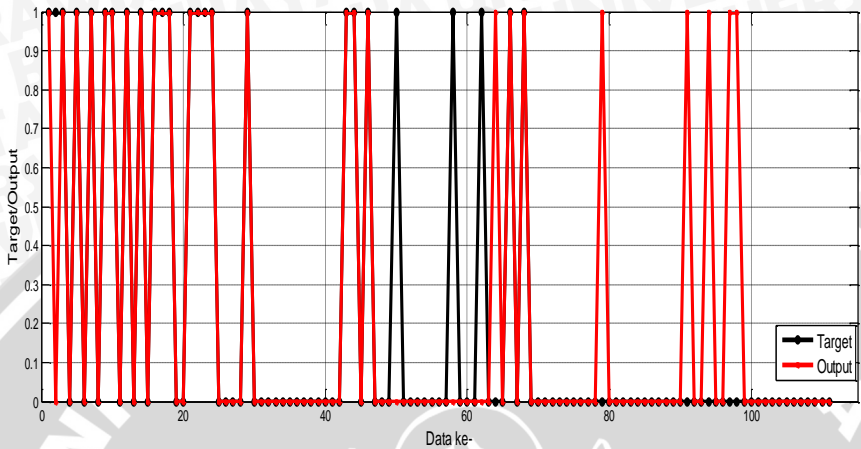


Gambar 4.1 Grafik hasil pengujian data 1 *training*

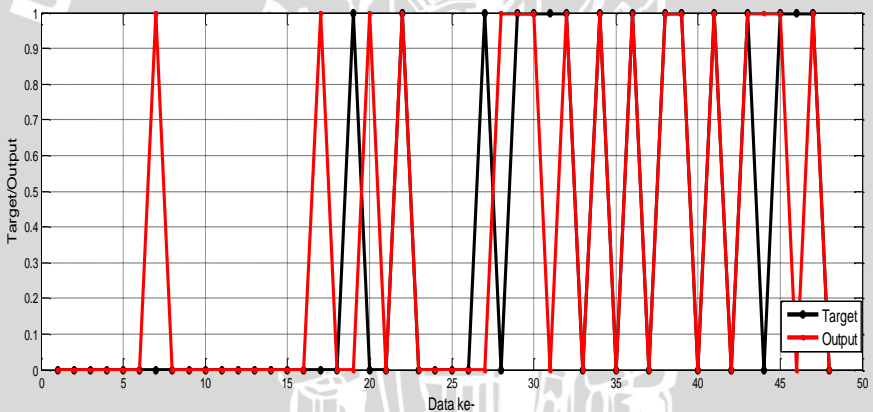


Gambar 4.2 Grafik hasil pengujian data 1 *testing*

Pada Gambar 4.1 dan Gambar 4.2 tanda bintang menunjukkan nilai data *output* dan tanda lingkaran menunjukkan nilai data target atau nilai data respon. Hasil grafik pada data *training* menunjukkan bahwa nilai *output* hampir mendekati nilai target dan hasil pada data *testing* menunjukkan bahwa nilai *output* hampir mendekati nilai target. Pada grafik data *training* menunjukkan banyak data yang salah dalam klasifikasi sebanyak 28 data dan pada grafik data *testing* menunjukkan kesalahan klasifikasi ada pada data ke-12, 21, 23 dan 24.



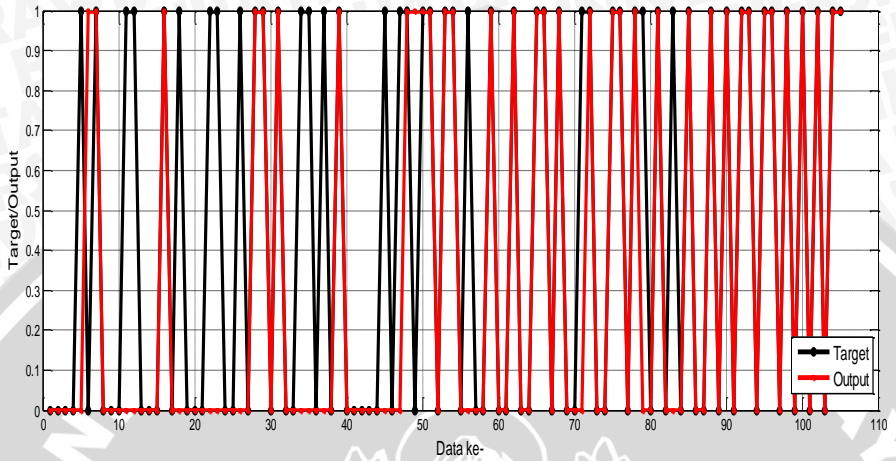
Gambar 4.3 Grafik hasil pengujian data 2 *training*



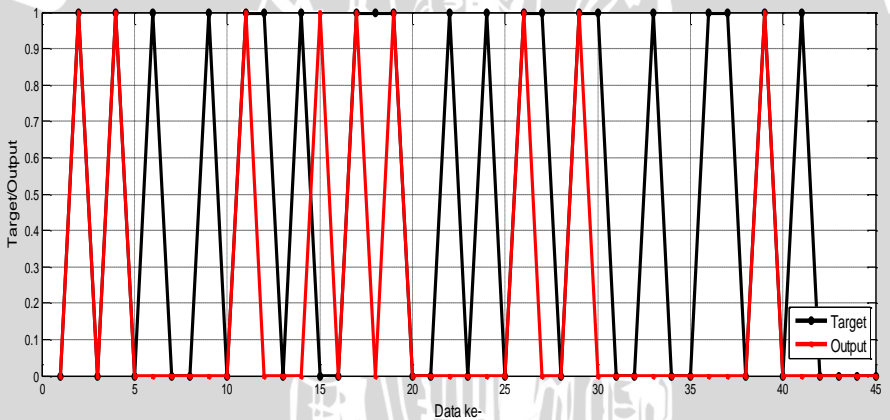
Gambar 4.4 Hasil pengujian data 2 *testing*

Pada Gambar 4.3 dan Gambar 4.4 garis merah menunjukkan pergerakan nilai data *output* dan garis hitam menunjukkan pergerakan nilai data *target* atau nilai data *respon*. Pada grafik data *training* menunjukkan sebanyak 10 data yang salah dalam pengklasifikasian dan pada grafik data *testing* menunjukkan kesalahan klasifikasi sebanyak 9 data.





Gambar 4.5 Grafik hasil pengujian data 3 *training*



Gambar 4.6 Grafik hasil pengujian data 3 *testing*

Hasil grafik pada Gambar 4.5 dan Gambar 4.6 menunjukkan bahwa nilai *output* (garis merah) dan nilai *target* (garis hitam), terdapat 18 data yang salah dalam pengklasifikasian, sedang pada data *testing* terdapat kesalahan klasifikasi sebanyak 14 data.

### 4.2.3 Pengukuran Ketepatan Pengklasifikasian

Menentukan keakuratan pengelompokkan baik pada kategori nol (0) dan satu (1) digunakan nilai ketepatan klasifikasi *Hit Ratio*. Nilai

*Hit Ratio* menyatakan nilai proporsi sampel yang benar diklasifikasikan oleh fungsi klasifikasi (Johnson and Wichern, 1992). Penentuan kesalahan pengklasifikasian dapat diketahui melalui tabel klasifikasi sebagai berikut :

Tabel 4.18 Ketepatan Klasifikasi Data 1 *Training* dan *Testing*

Aktual		Y (Prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y ( <i>Training</i> )	0	50	0	50	100
	1	28	18	46	39,1
<b>Total</b>		78	18	96	<b>70,8</b>
Y ( <i>Testing</i> )	0	0	0	0	0
	1	4	37	41	90,2
<b>Total</b>		4	37	41	<b>90,2</b>

Pada data 1 (pelanggaran lalu lintas) *training*, metode RBFNN dapat melakukan pengelompokkan dengan tepat pada kategori nol (jenis pelanggaran ringan) sebesar 100%, sedangkan pada kategori satu (jenis pelanggaran tidak ringan) mampu melakukan pengelompokkan sebesar 39,1% secara keseluruhan, model jenis pelanggaran lalu lintas telah mampu memberikan tingkat akurasi sebesar 70,8%. Pada data *testing*, keakuratan pengelompokkan pada kategori nol (pelanggaran ringan) sebesar 0%, pada kategori satu (pelanggaran tidak ringan) mampu melakukan pengklasifikasian dengan tepat sebesar 90,2% sehingga model jenis pelanggaran lalu lintas yang terbentuk telah mampu memberikan tingkat akurasi sebesar 90,2%.

Tabel 4.19 Ketepatan Klasifikasi Data 2 *Training* dan *Testing*

Aktual		Y (Prediksi)		Total	Ketepatan Klasifikasi
		0	1		
Y ( <i>Training</i> )	0	80	6	86	93
	1	4	21	25	84
<b>Total</b>		84	27	111	<b>90,9</b>
Y ( <i>Testing</i> )	0	27	5	32	84,4
	1	4	12	16	75
<b>Total</b>		31	17	48	<b>81,2</b>

Pada Tabel 4.19 untuk data *training*, model RBFNN secara keseluruhan mampu mendiagnosis kanker payudara dengan tingkat akurasi sebesar 90,9%. Pada data *testing*, model diagnosis kanker payudara yang terbentuk adalah sebesar 81,2%.

Tabel 4.20 Ketepatan Klasifikasi Data 3 *Training* dan *Testing*

Aktual	Y (Prediksi)		Total	Ketepatan Klasifikasi	
	0	1			
Y ( <i>Training</i> )	0	54	2	56	96,4
	1	16	33	49	67,3
<b>Total</b>		70	45	105	<b>82,8</b>
Y ( <i>Testing</i> )	0	23	1	24	95,8
	1	13	8	21	38,1
<b>Total</b>		39	9	45	<b>68,9</b>

Pada data 3 (diagnosis penderita demam berdarah) *training*, model RBFNN memberikan tingkat akurasi sebesar 82,8%. Pada data *testing*, keakuratan pengelompokkan model diagnosis pernah menderita penyakit demam berdarah yang terbentuk adalah sebesar 68,9%.

#### 4.3 Pemilihan Model Terbaik

Pemilihan model terbaik antara regresi logistik dengan RBFNN dilakukan dengan melihat keputusan klasifikasi data *testing* pada masing-masing kategori variabel respon. Berikut tabel hasil perbandingan ketepatan klasifikasi berdasarkan nilai APER dan model terbaik berdasarkan nilai SSE:

Tabel 4.21 Kesalahan Klasifikasi (APER) dan nilai SSE Data *Testing*

	Regresi Logistik Biner		RBFNN	
	APER	SSE	APER	SSE
<b>Data 1</b>	14,6	7,434	9,7	7,226
<b>Data 2</b>	22,9	7,538	20,8	7,303
<b>Data 3</b>	35,5	9,995	31,1	9,173

$$APER = \left[ 1 - \left( \frac{\text{jumlah individu yang diklasifikasi dengan tepat}}{\text{jumlah sampel}} \right) \right] \times 100\%$$

$$SSE = \sum_{m=1}^p (Y_m - \hat{Y}_m)^2$$

Pada Tabel 4.21 nilai persentase APER model regresi logistik untuk semua data lebih besar dari model RBFNN dan pada nilai SSE model regresi logistik untuk semua data lebih besar dari model RBFNN. Sehingga dapat disimpulkan bahwa dalam pengklasifikasian peubah respon biner model RBFNN memiliki kemampuan pengklasifikasian lebih tepat / lebih baik daripada model regresi logistik.

UNIVERSITAS BRAWIJAYA



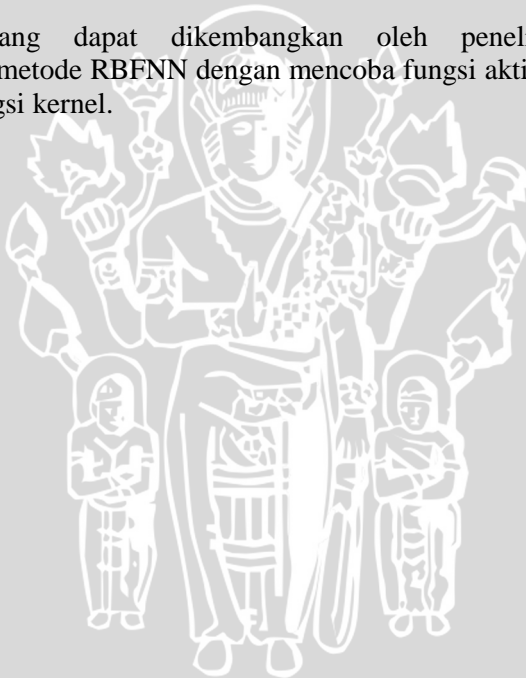
## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Perbandingan model regresi logistik dan model *radial basis function neural network* untuk pengklasifikasian peubah respon biner berdasarkan nilai *testing* APER dan SSE pada data 1, data 2, dan data 3 menunjukkan bahwa model RBFNN lebih baik dalam pengklasifikasian peubah respon biner karena nilai *testing* APER dan SSE model RBFNN lebih kecil dari pada nilai *testing* APER dan SSE model regresi logistik.

### 5.2 Saran

Saran yang dapat dikembangkan oleh peneliti yaitu mengembangkan metode RBFNN dengan mencoba fungsi aktivasi yang lain misalnya fungsi kernel.





UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Agresti, A. 1990. *Categorical Data Analysis*. John Willey and Sons, New York.
- Arifanjuni, R. 2013. *Faktor-Faktor yang Menyebabkan Terkena DB (Demam Berdarah) di Daerah Bengkulu dengan Menggunakan Regresi Logistik Biner*. Fakultas MIPA ITS, Surabaya.
- Broomhead, D. S. dan Lowe, D. 1998. Multivariable Functional Interpolation and Adaptive Network. *Jurnal Complex System*, 2 : 321-355.
- Buhmann, M. D. 1998. Radial Function on Compact Support. *Jurnal Proc. Edinburgh Math Soc*, 41 (2) : 33-46.
- Darmawan, R. 2009. *Perbandingan Jaringan Syaraf Tiruan Algoritma Backpropagation dan Algoritma Genetika untuk Peramalan Data Time Series*. Skripsi Program Studi Statistika Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang (Tidak Dipublikasikan).
- Demuth, H., dan Mark, B. 2000. *Neural Network Toolbox For Use with MATLAB User's Guide Version 4*. The MathWorks, Inc.
- Draper, N. R., dan Smith, H. 1992. *Analisis Regresi Terapan Edisi Kedua*. Gramedia, Jakarta.
- Du, K. L. dan Swamy, M. N. S. 2006. *Neural Networks in A Softcomputing Framework*. Springer-Verlag, London, Inggris.
- Fahrmeir, L. dan Gerhard, T. 1994. *Multivariate Statistical Modelling Based on Generalized Linier Models*. John Willey and Sons, New York.
- Gerald, C. F., dan Wheatley, P. O. 1996. *Applied Numerical Analysis Fifth Edition*. Wesley Company. California.
- Gujarati, D. 1991. *Ekonometrika Dasar*. Alih Bahasa S. Zain. Erlangga, Jakarta.
- Gupta, M. M., Jin, L., dan Homma, N. 2003. *Static and Dynamic Neural Networks : From Fundamental to Advanced Theory*, John Willey & Sons, New York, USA.
- Hosmer, D. W. dan Lemeshow, S. 1989. *Applied Logistic Regression*. John Willey and Sons Inc, New York.
- Johnson, R. A., dan Wichern, D. W. 1992. *Applied Multivariate Statistical Analysis*. Prentice Hall. New Jersey.

- Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan (Menggunakan MATLAB&Excel Link)*. Graha Ilmu, Yogyakarta.
- Lind, D.A., Robert D. M., dan William G. M. 2000. *Basic Statistics for Business and Economics*. 3<sup>rd</sup> edition. Mc-graw Hill. Inc, Singapore.
- Mark, J. L. 1999. *Matlab Function For Radial Basis Function*. Division of Informatics. Edinburgh University, Scotland.
- McClave, J.T dan Sincieh, T. 2003. *Statistics*. 9<sup>th</sup> edition. Prentice Hall, Singapore.
- Novianti, F. A. 2013. *Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasarkan Hasil Mamografi Studi Kasus Rumah Sakit Surabaya*. Fakultas MIPA ITS, Surabaya.
- Orr, M. J. L. 1996. *Introduction to Radial Basis Function Network*. Center for Cognitive Science, Scotland, UK.
- Pakaja, F. 2012. Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor. *Jurnal EECCIS*, 6(1) : 82-122.
- Patterson, D.W. 1996. *Artificial Neural Network Theory and Application*. Prentice Hall, Singapore.
- Purnomo, M.H. dan Kurniawan, A. 2006. *Supervised Neural Networks dan Aplikasinya*. Graha Ilmu, Yogyakarta.
- Sanchez, M. S., 1996. Performance of Multi Layer Feedforward and Radial Basis Functional Neural Networks in Classification and Modeling. *Jurnal Chemometrics and Intelligent Laboratory System*, 33 : 101-119.
- Santoso, F. H. 2012. *Faktor-Faktor yang Membedakan Jenis Pelanggaran Lalu Lintas di Polres Sidoarjo dengan Menggunakan Metode Regresi Logistik Biner*. Fakultas MIPA ITS, Surabaya.
- Saravanan, R., dan Malika. 2009. *An Effective Classification Model for Cancer Diagnosis Using Micro Array Gene Expression Data*. *Proceedings of the 2009 International Conference on Computer Engineer and Technology*. IEEE Computer Society, Washington.

- Sarimveis, H., Doganis, P., dan Alexandridis, A. 2005. A Classification Technique Based on Radial Basis Function Neural Networks. *Jurnal Advance in Engineering Software*, 37 : 218-221.
- Sembiring, R.K. 1995. *Analisis Regresi*. Penerbit ITB, Bandung.
- Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Andi Offset, Yogyakarta.
- Suhartono. 2007. *Feed Foward Neural Networks untuk Pemodelan Runtun Waktu*. Fakultas MIPA UGM, Yogyakarta.
- Sumodiningrat, G. 1994. *Ekonometrika Pengantar*. BPFE, Yogyakarta.
- Sutijo, B., Subanar dan Guritno S. 2006. Input-nodes Link Selection on Radial Basis Function Neural Networks. *Jurnal MIPA*, 16 (1) : 55-61.
- Yao, J. T. dan Tan, C. L. 2001. *Guidelines for Financial Forecasting with Neural Networks*.  
[http://www2.cs.uregina.ca/~jtiao/Papers/guide\\_iconip01.pdf](http://www2.cs.uregina.ca/~jtiao/Papers/guide_iconip01.pdf).  
tanggal akses 25 Januari 2013.



UNIVERSITAS BRAWIJAYA





## Lampiran 1. Data 1

No	Y	X1	X2	X3	X4	X5	X6
1.	1	1	0	1	1	1	1
2.	0	1	0	1	1	0	0
3.	0	0	0	1	1	0	0
4.	0	0	0	1	1	3	3
5.	0	0	0	3	1	2	2
6.	1	1	1	0	0	1	1
7.	1	1	1	0	1	2	2
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
136	1	0	0	1	0	0	0
137	1	0	0	1	0	0	0

Y = Jenis pelanggaran

0= pelanggaran ringan

1= pelanggaran tidak ringan

X1= Pekerjaan pelanggaran

0= bekerja

1= tidak bekerja

X2= Kendaraan pelanggar

0= roda dua

1= roda empat

X3= Usia

0= <20 th

1= 21-30 th

2= 31-40 th

3= ≥41 th

X5= Waktu pelanggaran

0= 05.00-10.00

1= 10.01-15.00

2= 15.01-18.00

3= 18.01-24.00

X6= Wilayah hukum

0= utara

1= selatan

2= timur

3= barat

X4= Hari pelanggaran

0= hari kerja

1= hari libur

UNIVERSITAS BRAWIJAYA



## Lampiran 2. Data 2

No	Y	X1	X2	X3	X4
1.	1	0	4	7	1
2.	1	0	5	5	0
3.	1	0	5	7	1
4.	0	2	2	0	0
5.	1	1	5	5	1
6.	0	0	2	0	1
7.	1	0	4	5	1
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
158.	1	0	5	7	0
159.	0	1	2	0	0

Y = Diagnosis kanker Payudara

0= kanker jinak

1= kanker ganas

X1= *Intermediate findings*

0= tidak ada tanda

1= ada 1 tanda

2= ada >1 tanda

X3= *Suspicious for Malignancy*

0= tidak ada ciri keganasan

1= *mass*

2= *calcification*

3= *speculatedsign*

4= *mass,calcification*

5= *mass, speculatedsign*

6= *calcification, speculated sign*

7= *mass, calcification, speculatedsign*

X2= Kategori BIRAD-S

0= C-0

1= C-1

2= C-2

3= C-3

4= C-4

5= C-5

X4= Letak abnormal

0= kanan

1= kiri

UNIVERSITAS BRAWIJAYA



### Lampiran 3. Data 3

No	Y	X1	X2	X3	X4	X5	X6
1.	0	1	0	1	0	1	1
2.	0	1	0	1	0	2	1
3.	0	0	0	0	1	0	1
4.	0	0	2	1	1	0	0
5.	1	0	0	1	0	2	3
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
149.	0	1	3	1	0	1	4
150.	0	1	5	1	1	1	1

Y = seseorang yang pernah didiagnose menderita penyakit demam berdarah

0= ya

1= tidak

X1= Jenis kelamin

0 = laki-laki

1 = perempuan

X2= Pendidikan

0= tidak pernah sekolah

1= tidak tamat SD

2= tamat SD

3= tamat SMP

4= tamat SMA

5= tamat PT

X3= Tipe Daerah

0= kota

1= desa

X4= Tidur dengan Kelambu

0= ya

1= tidak

2= tidak tahu

X5= Sarana Menampung Air Minum

0= langsung sumber

1= wadah terbuka

2= wadah tertutup

X6= Penampungan Air Limbah

0= penampungan tertutup

1= penampungan terbuka

2= luar pekarangan

3= didalam tanah

4= langsung ke got/sungai



UNIVERSITAS BRAWIJAYA



## Lampiran 4. Output SPSS Analisis Regresi Logistik Biner

### 1. Uji Multikolinieritas Data 1, 2, dan 3

Correlations

		x1	x2	x3	x4	x5	x6
Spearman's rho	x1 Correlation Coefficient	1.000	-.024	-.136	-.179	-.141	-.141
	Sig. (2-tailed)	.	.781	.113	.037	.101	.101
	N	137	137	137	137	137	137
x2	Correlation Coefficient	-.024	1.000	.077	-.049	-.111	-.111
	Sig. (2-tailed)	.781	.	.369	.569	.195	.195
	N	137	137	137	137	137	137
x3	Correlation Coefficient	-.136	.077	1.000	-.101	-.049	-.049
	Sig. (2-tailed)	.113	.369	.	.239	.572	.572
	N	137	137	137	137	137	137
x4	Correlation Coefficient	-.179	-.049	-.101	1.000	.131	.131
	Sig. (2-tailed)	.037	.569	.239	.	.126	.126
	N	137	137	137	137	137	137
x5	Correlation Coefficient	-.141	-.111	-.049	.131	1.000	0.232
	Sig. (2-tailed)	.101	.195	.572	.126	.	.
	N	137	137	137	137	137	137
x6	Correlation Coefficient	-.141	-.111	-.049	.131	0.232	1.000
	Sig. (2-tailed)	.101	.195	.572	.126	.	.
	N	137	137	137	137	137	137

\*. Correlation is significant at the 0.05 level (2-tailed).

\*\* Correlation is significant at the 0.01 level (2-tailed).

## Lampiran 4. (Lanjutan)

### 1. Uji Multikolinieritas Data 1, 2, dan 3

Correlations

		x1	x2	x3	x4
Spearman's rho x1	Correlation Coefficient	1.000	.028	-.161*	-.041
	Sig. (2-tailed)	.	.727	.043	.606
	N	159	159	159	159
x2	Correlation Coefficient	.028	1.000	.284**	.069
	Sig. (2-tailed)	.727	.	.000	.387
	N	159	159	159	159
x3	Correlation Coefficient	-.161*	.284**	1.000	.038
	Sig. (2-tailed)	.043	.000	.	.631
	N	159	159	159	159
x4	Correlation Coefficient	-.041	.069	.038	1.000
	Sig. (2-tailed)	.606	.387	.631	.
	N	159	159	159	159

\*. Correlation is significant at the 0.05 level (2-tailed).

\*\* . Correlation is significant at the 0.01 level (2-tailed).



## Lampiran 4. (Lanjutan)

### 1. Uji Multikolinieritas Data 1, 2, dan 3

		Correlations					
		x1	x2	x3	x4	x5	x6
Spearman's rho	x1 Correlation Coefficient	1.000	.027	.502**	-.003	-.323**	.271**
	Sig. (2-tailed)	.	.743	.000	.973	.000	.001
	N	150	150	150	150	150	150
	x2 Correlation Coefficient	.027	1.000	-.021	-.036	.015	-.082
	Sig. (2-tailed)	.743	.	.798	.663	.852	.316
	N	150	150	150	150	150	150
	x3 Correlation Coefficient	.502**	-.021	1.000	-.055	-.563**	.541**
	Sig. (2-tailed)	.000	.798	.	.505	.000	.000
	N	150	150	150	150	150	150
	x4 Correlation Coefficient	-.003	-.036	-.055	1.000	-.032	-.102
	Sig. (2-tailed)	.973	.663	.505	.	.699	.216
	N	150	150	150	150	150	150
	x5 Correlation Coefficient	-.323**	.015	-.563**	-.032	1.000	-.338**
	Sig. (2-tailed)	.000	.852	.000	.699	.	.000
	N	150	150	150	150	150	150
	x6 Correlation Coefficient	.271**	-.082	.541**	-.102	-.338**	1.000
	Sig. (2-tailed)	.001	.316	.000	.216	.000	.
	N	150	150	150	150	150	150

\*\* . Correlation is significant at the 0.01 level (2-tailed).

### 2. Estimasi Parameter (Uji Simultan dan Parsial) Data 1, 2, dan 3

		Omnibus Tests of Model Coefficients		
		Chi-square	df	Sig.
Step 1	Step	21.858	9	.009
	Block	21.858	9	.009
	Model	21.858	9	.009

## Lampiran 4. (Lanjutan)

### 2. Estimasi Parameter (Uji Simultan dan Parsial) Data 1, 2, dan 3

Omnibus Tests of Model Coefficients				
		Chi-square	df	Sig.
Step 1	Step	59.510	9	.000
	Block	59.510	9	.000
	Model	59.510	9	.000

Omnibus Tests of Model Coefficients				
		Chi-square	df	Sig.
Step 1	Step	133.730	16	.000
	Block	133.730	16	.000
	Model	133.730	16	.000

Variables in the Equation						
	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 <sup>a</sup> x1(1)	1.174	.446	6.919	1	.009	3.235
x2(1)	-1.186	.488	5.903	1	.015	.305
x3			2.221	3	.528	
x3(1)	.874	.683	1.637	1	.201	2.396
x3(2)	.129	.563	.053	1	.819	1.138
x3(3)	.458	.610	.563	1	.453	1.580
x4(1)	.966	.418	5.351	1	.021	2.629
x5			.667	3	.881	
x5(1)	-.448	.602	.553	1	.457	.639
x5(2)	-.367	.610	.363	1	.547	.693
x5(3)	-.153	.702	.047	1	.827	.858
x6			.667	3	.881	
x6(1)	-.448	.602	.553	1	.457	.639
x6(2)	-.367	.610	.363	1	.547	.693
x6(3)	-.153	.702	.047	1	.827	.858
Constant	.092	.827	.012	1	.912	1.096

a. Variable(s) entered on step 1: x1, x2, x3, x4, x5, x6.



## Lampiran 4. (Lanjutan)

### 2. Estimasi Parameter (Uji Simultan dan Parsial) Data 1, 2, dan 3

Variables in the Equation

	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 <sup>a</sup> x1			2.912	2	.233	
x1(1)	1.117	.867	1.660	1	.198	3.055
x1(2)	.438	.914	.230	1	.632	1.550
x2			8.823	2	.012	
x2(4)	-1.598	.635	6.332	1	.012	.202
x2(5)	.580	.584	.985	1	.321	1.785
x3			18.501	4	.001	
x3(2)	-3.075	.872	12.429	1	.000	.046
x3(3)	-2.053	.756	7.379	1	.007	.128
x3(5)	-1.200	1.242	.933	1	.334	.301
x3(7)	-1.509	.565	7.143	1	.008	.221
x4(1)	-1.780	.507	12.308	1	.000	.169
Constant	.918	.875	1.102	1	.294	2.505

a. Variable(s) entered on step 1: x1, x2, x3, x4.

## Lampiran 4. (Lanjutan)

### 2. Estimasi Parameter (Uji Simultan dan Parsial) Data 1, 2, dan 3

Variables in the Equation

	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 <sup>a</sup> x1(1)	1.599	.737	4.709	1	.030	4.947
x2			2.087	5	.837	
x2(1)	-1.267	1.846	.472	1	.492	.282
x2(2)	.377	2.066	.033	1	.855	1.458
x2(3)	-1.298	1.447	.805	1	.370	.273
x2(4)	-.514	1.481	.120	1	.729	.598
x2(5)	-1.042	1.475	.499	1	.480	.353
x3(1)	2.712	1.044	6.747	1	.009	15.059
x4			1.094	2	.579	
x4(1)	1.327	1.276	1.081	1	.298	3.771
x4(2)	1.059	1.309	.655	1	.418	2.884
x5			.044	2	.978	
x5(1)	-.030	.991	.001	1	.976	.971
x5(2)	.280	1.414	.039	1	.843	1.323
x6			9.150	4	.057	
x6(1)	1.867	1.202	2.412	1	.120	6.472
x6(2)	-2.144	1.496	2.053	1	.152	.117
x6(3)	-.995	1.651	.363	1	.547	.370
x6(4)	1.340	1.274	1.105	1	.293	3.818
Constant	-3.429	2.358	2.115	1	.146	.032

a. Variable(s) entered on step 1: x1, x2, x3, x4, x5, x6.

## Lampiran 4. (Lanjutan)

### 3. Uji Kelayakan Model Data 1, 2, dan 3

**Variables in the Equation**

	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 <sup>a</sup> x1(1)	1.198	.425	7.944	1	.005	3.314
x2(1)	-1.120	.461	5.904	1	.015	.326
x4(1)	.889	.397	5.010	1	.025	2.433
Constant	.084	.557	.023	1	.880	1.088

a. Variable(s) entered on step 1: x1, x2, x4.

**Variables in the Equation**

	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 <sup>a</sup> x2			8.179	2	.017	
x2(4)	-1.430	.611	5.487	1	.019	.239
x2(5)	.611	.567	1.160	1	.282	1.841
x3			19.587	4	.001	
x3(2)	-3.130	.845	13.712	1	.000	.044
x3(3)	-2.123	.762	7.771	1	.005	.120
x3(5)	-1.022	1.220	.701	1	.402	.360
x3(7)	-1.478	.552	7.168	1	.007	.228
x4(1)	-1.761	.498	12.487	1	.000	.172
Constant	1.646	.546	9.077	1	.003	5.187

a. Variable(s) entered on step 1: x2, x3, x4.

**Variables in the Equation**

	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 <sup>a</sup> x1(1)	1.213	.563	4.640	1	.031	3.363
x3(1)	4.316	.618	48.775	1	.000	74.922
Constant	-3.253	.575	31.976	1	.000	.039

a. Variable(s) entered on step 1: x1, x3.

## Lampiran 4. (Lanjutan)

### 3. Uji Kelayakan Model Data 1, 2, dan 3

#### Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	3.849	6	.697

#### Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	161.439 <sup>a</sup>	.125	.172

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than .001.

#### Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	5.663	8	.685

#### Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	125.050 <sup>a</sup>	.299	.439

a. Estimation terminated at iteration number 6 because parameter estimates changed by less than .001.

#### Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	7.256	3	.064

#### Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	89.833 <sup>a</sup>	.543	.725





## Lampiran 5.(Lanjutan)

### 2. Output Bobot Bias Input Data 1, 2, dan 3

	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651
	1,9870	1,6651

### 3. Output Bobot Lapisan Data 1, 2, dan 3

bobotlapisan data 1= 0,5765    0    0,6378    0 0,5875    0,5709	bobotlapisan data 2= Columns 1 through 7 0    0,7673    0,9577    0,9167 0    0,9263    0,9495 Columns 8 through 14 0,9533    0,9683    0,9664 0,9320    0,4660    0 0,4320 Columns 15 through 21 0    -0,0382    0    0 0,4555    0    0,4496 Columns 22 through 25 0    0,4306    0    0,3007	bobotlapisan data 3= Columns 1 through 7 0    0,7443    0,8213    0,7986 0    0,4853    0,8546 Columns 8 through 14 0,8358    0,8331    0,7757 0,7555,7443    0    0,5189 Columns 15 through 21 0    -0,1053    0    0 0,3545    0    0,8114 Columns 22 through 25 0    0,1453    0    0,2134
--	---	--

### 4. Output Bobot Bias Lapisan Data 1, 2, dan 3

<b>bobotbiaslapisan data 1=</b> <b>0,4139</b>	<b>bobotbiaslapisan data 2=</b> <b>0,0316</b>	<b>bobotbiaslapisan data 3=</b> <b>0,1998</b>
--	--	--

**Lampiran 6. Hasil Prediksi Klasifikasi Data 1 dengan Model RegLog Biner dan RBFNN**

No	Y aktual	Prediksi (RLB)	Prediksi (RBF)	Error (RBL)	Error (RBF)	Klasifikasi (RLB)	Klasifikasi (RBF)
1	1	0	0	0,73801	0,586	tidak	tidak
2	0	0	0	-0,26199	-0,414	sesuai	sesuai
3	0	1	0	-0,54052	-0,414	tidak	sesuai
4	0	1	0	-0,54052	-0,418	tidak	sesuai
5	0	1	0	-0,54052	-0,416	tidak	sesuai
6	1	1	0	0,27417	0,586	sesuai	tidak
7	1	1	0	0,47891	0,586	sesuai	tidak
8	0	0	0	-0,46345	-0,416	sesuai	sesuai
9	1	1	1	0,21713	0,007	sesuai	sesuai
10	0	1	0	-0,74109	-0,416	tidak	sesuai
11	1	1	0	0,21713	0,586	sesuai	tidak
12	0	1	0	-0,54052	-0,414	tidak	sesuai
13	0	0	0	-0,46345	-0,416	sesuai	sesuai
14	0	0	0	-0,26199	-0,414	sesuai	sesuai
15	0	1	0	-0,74109	-0,416	tidak	sesuai
16	0	1	0	-0,78287	-0,451	tidak	sesuai
17	0	1	0	-0,54052	-0,487	tidak	sesuai
18	0	1	0	-0,74109	-0,455	tidak	sesuai
19	0	1	0	-0,52108	-0,414	tidak	sesuai
20	0	0	0	-0,26199	-0,414	sesuai	sesuai
21	0	1	0	-0,89768	-0,455	tidak	sesuai
22	0	1	0	-0,54052	-0,419	tidak	sesuai
23	0	1	0	-0,74109	-0,414	tidak	sesuai
24	0	1	0	-0,54052	-0,416	tidak	sesuai
25	0	0	0	-0,46345	-0,414	sesuai	sesuai
26	1	1	1	0,21713	0,007	sesuai	sesuai
27	0	0	0	-0,46345	-0,414	sesuai	sesuai
28	0	1	0	-0,78287	-0,414	tidak	sesuai
29	0	0	0	-0,26199	-0,414	sesuai	sesuai
30	0	1	0	-0,74109	-0,414	tidak	sesuai
31	0	1	0	-0,54052	-0,416	tidak	sesuai
32	0	0	0	-0,46345	-0,455	sesuai	sesuai
33	0	0	0	-0,26199	-0,414	sesuai	sesuai
34	0	1	0	-0,72583	-0,414	tidak	sesuai
35	0	1	0	-0,54052	-0,414	tidak	sesuai
36	1	1	0	0,45948	0,581	sesuai	tidak
37	1	1	0	0,45948	0,583	sesuai	tidak

## Lampiran 6. (Lanjutan)

38	1	1	1	0,45948	0,487	sesuai	sesuai
39	1	1	0	0,45948	0,581	sesuai	tidak
40	0	0	0	-0,26199	-0,414	sesuai	sesuai
41	0	0	0	-0,46345	-0,414	sesuai	sesuai
42	1	1	1	0,21713	0,007	sesuai	sesuai
43	1	1	0	0,10232	0,586	sesuai	tidak
44	1	1	0	0,21713	0,586	sesuai	tidak
45	0	1	0	-0,74109	-0,451	tidak	sesuai
46	0	0	0	-0,46345	-0,455	sesuai	sesuai
47	1	1	1	0,45948	-0,001	sesuai	sesuai
48	0	1	0	-0,89768	-0,414	tidak	sesuai
49	0	1	0	-0,89768	-0,416	tidak	sesuai
50	0	1	0	-0,54052	-0,414	tidak	sesuai
51	0	1	0	-0,72583	-0,414	tidak	sesuai
52	0	1	0	-0,54052	-0,487	tidak	sesuai
53	1	1	0	0,25891	0,583	sesuai	tidak
54	1	1	0	0,45948	0,548	sesuai	tidak
55	0	1	0	-0,74109	-0,416	tidak	sesuai
56	0	1	0	-0,74109	-0,414	tidak	sesuai
57	1	1	1	0,45948	-0,001	sesuai	sesuai
58	0	1	0	-0,74109	-0,455	tidak	sesuai
59	0	0	0	-0,26199	-0,414	sesuai	sesuai
60	1	0	0	0,53655	0,586	tidak	tidak
61	1	1	1	0,10232	0,455	sesuai	sesuai
62	1	1	0	0,45948	0,586	sesuai	tidak
63	1	1	1	0,25891	-0,052	sesuai	sesuai
64	0	1	0	-0,54052	-0,414	tidak	sesuai
65	1	1	0	0,10232	0,586	sesuai	tidak
66	1	1	0	0,21713	0,583	sesuai	tidak
67	1	1	1	0,25891	-0,052	sesuai	sesuai
68	1	1	1	0,45948	0,455	sesuai	sesuai
69	1	1	0	0,10232	0,583	sesuai	tidak
70	1	1	1	0,10232	0,455	sesuai	sesuai
71	0	0	0	-0,26199	-0,414	sesuai	sesuai
72	1	1	1	0,25891	0,455	sesuai	sesuai
73	0	0	0	-0,46345	-0,414	sesuai	sesuai
74	0	1	0	-0,54052	-0,414	tidak	sesuai
75	1	1	1	0,45948	0,455	sesuai	sesuai
76	0	1	0	-0,54052	-0,414	tidak	sesuai
77	0	0	0	-0,26199	-0,416	sesuai	sesuai
78	1	1	1	0,25891	0,455	sesuai	sesuai
79	1	1	0	0,21713	0,586	sesuai	tidak

## Lampiran 6. (Lanjutan)

80	1	1	1	0,25891	0,455	sesuai	sesuai
81	1	1	0	0,10232	0,586	sesuai	tidak
82	0	1	0	-0,54052	-0,414	tidak	sesuai
83	1	1	1	0,21713	0,455	sesuai	sesuai
84	1	1	0	0,45948	0,583	sesuai	tidak
85	1	1	0	0,10232	0,583	sesuai	tidak
86	1	1	0	0,45948	0,586	sesuai	tidak
87	0	0	0	-0,46345	-0,414	sesuai	sesuai
88	1	1	0	0,45948	0,586	sesuai	tidak
89	1	1	0	0,21713	0,586	sesuai	tidak
90	1	1	1	0,25891	0,013	sesuai	sesuai
91	1	0	0	0,53655	0,586	tidak	tidak
92	1	1	0	0,25891	0,583	sesuai	tidak
93	1	0	0	0,53655	0,583	tidak	tidak
94	1	0	0	0,53655	0,583	tidak	tidak
95	1	1	1	0,25891	0,012	sesuai	sesuai
96	1	1	0	0,10232	0,586	sesuai	tidak
97	1	1	1	0,45948	0,414	sesuai	sesuai
98	1	1	1	0,45948	0,414	sesuai	sesuai
99	1	0	1	0,53655	0,416	tidak	sesuai
100	1	1	1	0,45948	0,416	sesuai	sesuai
101	1	1	1	0,21713	0,414	sesuai	sesuai
102	1	1	1	0,10232	0,414	sesuai	sesuai
103	1	1	1	0,27417	0,414	sesuai	sesuai
104	1	1	1	0,45948	0,414	sesuai	sesuai
105	1	1	1	0,45948	0,414	sesuai	sesuai
106	1	1	1	0,10232	0,414	sesuai	sesuai
107	1	1	1	0,45948	0,416	sesuai	sesuai
108	1	0	0	0,53655	0,549	tidak	tidak
109	1	0	1	0,53655	0,414	tidak	sesuai
110	1	1	1	0,45948	0,414	sesuai	sesuai
111	1	1	1	0,45948	0,414	sesuai	sesuai
112	1	1	1	0,45948	0,416	sesuai	sesuai
113	1	1	1	0,45948	0,414	sesuai	sesuai
114	1	1	1	0,45948	0,414	sesuai	sesuai
115	1	1	1	0,45948	0,414	sesuai	sesuai
116	1	1	1	0,45948	0,419	sesuai	sesuai
117	1	1	0	0,45948	0,549	sesuai	tidak
118	1	1	1	0,45948	-0,001	sesuai	sesuai
119	1	0	0	0,53655	0,549	tidak	tidak
120	1	1	0	0,45948	0,549	sesuai	tidak
121	1	1	1	0,49479	0,416	sesuai	sesuai

## Lampiran 6. (Lanjutan)

122	1	1	1	0,45948	0,416	sesuai	sesuai
123	1	1	1	0,10232	0,416	sesuai	sesuai
124	1	1	1	0,45948	-0,052	sesuai	sesuai
125	1	1	1	0,45948	0,414	sesuai	sesuai
126	1	1	1	0,27417	0,414	sesuai	sesuai
127	1	1	1	0,45948	0,414	sesuai	sesuai
128	1	1	1	0,45948	0,414	sesuai	sesuai
129	1	1	1	0,25891	0,414	sesuai	sesuai
130	1	1	1	0,27417	0,414	sesuai	sesuai
131	1	1	1	0,21713	0,414	sesuai	sesuai
132	1	0	1	0,53655	0,414	tidak	sesuai
133	1	0	1	0,53655	0,414	tidak	sesuai
134	1	1	1	0,45948	0,414	sesuai	sesuai
135	1	1	1	0,10232	0,414	sesuai	sesuai
136	1	1	1	0,45948	0,414	sesuai	sesuai
137	1	1	1	0,45948	0,414	sesuai	sesuai





## Lampiran 7. Hasil Prediksi Klasifikasi Data 2 dengan Model RegLog Biner dan RBFNN

No	Y aktual	Prediksi (RLB)	Prediksi (RBF)	Error (RBL)	Error (RBF)	Klasifikasi (RLB)	Klasifikasi (RBF)
1	1	1	1	1	0	sesuai	sesuai
2	1	0	0	0,04970	0,950	tidak	tidak
3	1	1	1	0,99990	1E-04	sesuai	sesuai
4	0	0	0	0,03160	-0,032	sesuai	sesuai
5	1	1	1	0,49990	0,500	sesuai	sesuai
6	0	0	0	0,03160	-0,032	sesuai	sesuai
7	1	1	1	0,49920	0,501	sesuai	sesuai
8	0	0	0	0,03160	-0,032	sesuai	sesuai
9	1	0	1	0,99800	0,002	sesuai	sesuai
10	1	0	1	0,83540	0,165	sesuai	sesuai
11	0	0	0	0,03160	-0,032	sesuai	sesuai
12	1	1	1	0,99810	0,002	sesuai	sesuai
13	0	0	0	0,03160	-0,032	sesuai	sesuai
14	1	1	1	0,99990	1E-04	sesuai	sesuai
15	0	1	0	0,03160	-0,032	sesuai	sesuai
16	1	1	1	1,01210	-0,012	sesuai	sesuai
17	1	1	1	0,50600	0,494	sesuai	sesuai
18	1	0	1	0,83540	0,165	sesuai	sesuai
19	0	0	0	0,03160	-0,032	sesuai	sesuai
20	0	0	0	0,03160	-0,032	sesuai	sesuai
21	1	1	1	1,00630	-0,006	sesuai	sesuai
22	1	1	1	0,49960	0,500	sesuai	sesuai
23	1	0	1	1	0	sesuai	sesuai
24	1	1	1	1,00630	-0,006	sesuai	sesuai
25	0	0	0	0,03180	-0,032	sesuai	sesuai
26	0	0	0	0,03740	-0,037	sesuai	sesuai
27	0	0	0	0,03160	-0,032	sesuai	sesuai
28	0	0	0	0,03160	-0,032	sesuai	sesuai
29	1	0	1	0,83540	0,165	sesuai	sesuai
30	0	0	0	0,03160	-0,032	sesuai	sesuai
31	0	0	0	0,07450	-0,074	sesuai	sesuai
32	0	0	0	0,03200	-0,032	sesuai	sesuai
33	0	0	0	0,03160	-0,032	sesuai	sesuai
34	0	0	0	0,04060	-0,041	sesuai	sesuai
35	0	0	0	0,03160	-0,032	sesuai	sesuai
36	0	0	0	0,03200	-0,032	sesuai	sesuai
37	0	0	0	0,07450	-0,074	sesuai	sesuai
38	0	0	0	0,03160	-0,032	sesuai	sesuai
39	0	1	0	1	-0,031	sesuai	sesuai

## Lampiran 7. (Lanjutan)

40	0	0	0	-0,16904	-0,037	sesuai	sesuai
41	0	0	0	-0,04640	-0,002	sesuai	sesuai
42	0	0	0	-0,16904	-0,032	sesuai	sesuai
43	1	0	1	-0,16422	1E-04	sesuai	sesuai
44	1	1	1	0,37852	0,165	sesuai	sesuai
45	0	0	0	0,52864	-0,032	tidak	sesuai
46	1	0	1	-0,00923	0,165	sesuai	sesuai
47	0	0	0	0,52864	-0,041	tidak	sesuai
48	0	0	0	-0,03750	-0,032	sesuai	sesuai
49	0	0	0	-0,00923	-0,037	sesuai	sesuai
50	1	0	0	-0,04640	0,925	sesuai	tidak
51	0	0	0	0,52864	-0,032	tidak	sesuai
52	0	0	0	-0,05143	-0,032	sesuai	sesuai
53	0	0	0	-0,09642	-0,032	sesuai	sesuai
54	0	0	0	-0,18478	-0,041	sesuai	sesuai
55	0	0	0	-0,27250	-0,032	sesuai	sesuai
56	0	0	0	-0,17580	-0,041	sesuai	sesuai
57	0	0	0	-0,47136	-0,032	sesuai	sesuai
58	1	0	0	-0,17580	0,968	sesuai	tidak
59	0	0	0	0,99077	-0,332	tidak	sesuai
60	0	0	0	-0,04640	-0,041	sesuai	sesuai
61	0	0	0	-0,06694	-0,041	sesuai	sesuai
62	1	0	0	-0,03750	0,668	sesuai	tidak
63	0	0	0	0,95359	-0,032	tidak	sesuai
64	0	0	1	-0,09642	-0,499	sesuai	tidak
65	0	1	0	-0,54199	-0,032	tidak	sesuai
66	1	0	1	-0,04640	1E-04	sesuai	sesuai
67	0	1	0	0,37852	-0,050	sesuai	sesuai
68	1	0	1	-0,38300	0,502	sesuai	sesuai
69	0	0	0	0,96249	-0,049	tidak	sesuai
70	0	0	0	-0,16904	-0,031	sesuai	sesuai
71	0	0	0	-0,17580	-0,032	sesuai	sesuai
72	0	0	0	-0,24297	-0,032	sesuai	sesuai
73	0	1	0	-0,55374	-0,032	tidak	sesuai
74	0	0	0	-0,00923	-0,049	sesuai	sesuai
75	0	0	0	-0,16904	-0,032	sesuai	sesuai
76	0	0	0	-0,16422	-0,032	sesuai	sesuai
77	0	0	0	-0,38300	-0,074	sesuai	sesuai
78	0	0	0	-0,47136	-0,049	sesuai	sesuai
79	0	0	1	-0,16904	-0,499	sesuai	tidak
80	0	1	0	-0,83837	-0,032	tidak	sesuai
81	0	0	0	-0,00923	-0,041	sesuai	sesuai

## Lampiran 7. (Lanjutan)

82	0	0	0	-0,05143	-0,032	sesuai	sesuai
83	0	0	0	-0,16904	-0,049	sesuai	sesuai
84	0	0	0	-0,04640	-0,332	sesuai	sesuai
85	0	0	0	-0,47136	-0,074	sesuai	sesuai
86	0	0	0	-0,16422	-0,032	sesuai	sesuai
87	0	0	0	-0,17580	-0,032	sesuai	sesuai
88	0	0	0	-0,07130	-0,032	sesuai	sesuai
89	0	0	0	-0,16904	-0,002	sesuai	sesuai
90	0	0	0	-0,16904	-0,002	sesuai	sesuai
91	0	1	1	-0,62148	-0,506	tidak	tidak
92	0	0	0	-0,27250	-0,049	sesuai	sesuai
93	0	0	0	-0,00923	-0,032	sesuai	sesuai
94	0	0	1	-0,03750	-0,498	sesuai	tidak
95	0	0	0	-0,16904	-0,049	sesuai	sesuai
96	0	0	0	-0,00923	-0,032	sesuai	sesuai
97	0	1	1	-0,68544	-0,499	tidak	tidak
98	0	0	1	-0,47136	-0,835	sesuai	tidak
99	0	0	0	-0,02489	-0,032	sesuai	sesuai
100	0	0	0	-0,04640	-0,037	sesuai	sesuai
101	0	0	0	-0,27250	-0,049	sesuai	sesuai
102	0	0	0	-0,03750	-0,041	sesuai	sesuai
103	0	0	0	-0,16904	-0,002	sesuai	sesuai
104	0	0	0	-0,00923	-0,032	sesuai	sesuai
105	0	0	0	-0,00923	-0,032	sesuai	sesuai
106	0	0	0	-0,16904	-0,049	sesuai	sesuai
107	0	0	0	-0,03750	-0,041	sesuai	sesuai
108	0	0	0	-0,09642	-0,032	sesuai	sesuai
109	0	0	0	-0,29446	-0,032	sesuai	sesuai
110	0	0	0	-0,17580	-0,032	sesuai	sesuai
111	0	0	0	-0,09642	-0,050	sesuai	sesuai
112	0	0	0	-0,04640	-0,332	sesuai	sesuai
113	0	0	0	-0,09642	-0,032	sesuai	sesuai
114	0	0	0	-0,02489	-0,032	sesuai	sesuai
115	0	0	0	-0,00923	-0,032	sesuai	sesuai
116	0	0	0	-0,24297	-0,032	sesuai	sesuai
117	0	1	0	-0,55374	-0,041	sesuai	sesuai
118	0	0	1	-0,47136	-0,835	sesuai	tidak
119	0	0	0	-0,03750	-0,041	sesuai	sesuai
120	0	0	0	-0,17580	-0,032	sesuai	sesuai
121	0	0	0	-0,18478	-0,032	sesuai	sesuai
122	0	0	0	-0,00923	-0,032	sesuai	sesuai
123	0	0	0	-0,00923	-0,032	sesuai	sesuai

## Lampiran 7. (Lanjutan)

124	0	0	0	-0,09642	-0,032	sesuai	sesuai
125	0	0	0	-0,00923	-0,032	sesuai	sesuai
126	0	1	0	-0,55374	-0,032	tidak	sesuai
127	0	0	0	-0,00923	-0,032	sesuai	sesuai
128	0	0	1	-0,18478	-0,835	sesuai	tidak
129	0	0	0	-0,18478	-0,032	sesuai	sesuai
130	1	0	0	0,52864	0,925	tidak	tidak
131	0	0	1	-0,03750	-0,498	sesuai	tidak
132	0	0	0	-0,00923	-0,032	sesuai	sesuai
133	1	0	1	0,81522	0,165	tidak	sesuai
134	0	0	0	-0,16904	-0,002	sesuai	sesuai
135	0	0	0	-0,09642	-0,032	sesuai	sesuai
136	0	0	0	-0,16904	-0,002	sesuai	sesuai
137	0	0	0	-0,03750	-0,041	sesuai	sesuai
138	1	0	0	0,81522	0,968	tidak	tidak
139	0	0	1	-0,27250	-0,049	sesuai	tidak
140	1	1	1	0,37852	0,494	sesuai	sesuai
141	1	1	1	0,09477	0	sesuai	sesuai
142	1	0	0	0,82419	0,968	tidak	tidak
143	1	1	1	0,16163	1E-04	sesuai	sesuai
144	0	0	0	-0,00923	-0,031	sesuai	sesuai
145	1	1	1	0,45801	0,5001	sesuai	sesuai
146	0	0	0	-0,05143	-0,031	sesuai	sesuai
147	1	1	1	0,31456	0,501	sesuai	sesuai
148	0	0	0	-0,05143	-0,032	sesuai	sesuai
149	1	0	1	0,81522	0,002	tidak	sesuai
150	1	0	1	0,81522	0,165	tidak	sesuai
151	0	0	0	-0,05143	-0,032	sesuai	sesuai
152	1	1	1	0,31456	0,002	sesuai	sesuai
153	0	0	0	-0,00923	-0,032	sesuai	sesuai
154	1	1	1	0,16163	-0,012	sesuai	sesuai
155	0	1	1	-0,83837	-1,012	tidak	tidak
156	1	1	1	0,45801	-0,006	sesuai	sesuai
157	1	0	0	0,81522	0,968	tidak	tidak
158	1	0	1	0,72750	0,165	tidak	sesuai
159	0	0	0	-0,00923	-0,032	sesuai	sesuai

## Lampiran 8. Hasil Prediksi Klasifikasi Data 3 dengan Model RegLog Biner dan RBFNN

No	Y aktual	Prediksi (RLB)	Prediksi (RBF)	Error (RBL)	Error (RBF)	Klasifikasi (RLB)	Klasifikasi (RBF)
1	0	0	0	-0,03876	-0,199	sesuai	sesuai
2	0	0	0	-0,03876	-0,199	sesuai	sesuai
3	0	1	0	-0,90713	-0,199	tidak	sesuai
4	0	0	0	-0,11433	-0,200	sesuai	sesuai
5	1	0	0	0,88567	0,800	tidak	tidak
6	0	1	1	-0,75317	-0,783	tidak	tidak
7	1	1	1	0,09287	-0,046	sesuai	sesuai
8	0	0	0	-0,03876	-0,199	sesuai	sesuai
9	0	0	0	-0,03876	-0,199	sesuai	sesuai
10	0	1	0	-0,90713	-0,347	tidak	sesuai
11	1	0	0	0,88567	0,765	tidak	tidak
12	1	1	0	0,09287	0,800	sesuai	tidak
13	0	1	0	-0,90713	-0,199	tidak	sesuai
14	0	1	0	-0,90713	-0,212	tidak	sesuai
15	0	0	0	-0,03876	-0,199	sesuai	sesuai
16	1	1	1	0,09287	-0,022	sesuai	sesuai
17	0	1	0	-0,75317	-0,199	tidak	sesuai
18	1	1	0	0,24683	0,761	sesuai	tidak
19	0	0	0	-0,03876	-0,199	sesuai	sesuai
20	0	1	0	-0,75317	-0,247	tidak	sesuai
21	0	0	0	-0,03876	-0,199	sesuai	sesuai
22	1	1	0	0,09287	0,691	sesuai	tidak
23	1	1	0	0,24683	0,797	sesuai	tidak
24	0	0	0	-0,03876	-0,200	sesuai	sesuai
25	0	0	0	-0,03876	-0,200	sesuai	sesuai
26	1	1	0	0,09287	0,744	sesuai	tidak
27	0	0	0	-0,03876	-0,200	sesuai	sesuai
28	1	1	1	0,09287	-0,002	sesuai	sesuai
29	1	1	1	0,24683	0,001	sesuai	sesuai
30	0	1	0	-0,75317	-0,203	tidak	sesuai
31	1	0	1	0,88567	0,051	tidak	sesuai
32	0	0	0	-0,03876	-0,199	sesuai	sesuai
33	0	0	0	-0,03876	-0,199	sesuai	sesuai
34	1	1	0	0,09287	0,800	sesuai	tidak
35	1	1	0	0,24683	0,715	sesuai	tidak
36	0	1	0	-0,75317	-0,247	tidak	sesuai
37	1	1	0	0,09287	0,653	sesuai	tidak
38	0	0	0	-0,03876	-0,202	sesuai	sesuai
39	1	1	1	0,09287	-0,002	sesuai	sesuai



## Lampiran 8. (Lanjutan)

40	0	0	0	-0,03876	-0,199	sesuai	sesuai
41	0	0	0	-0,03876	-0,200	sesuai	sesuai
42	0	0	0	-0,03876	-0,247	sesuai	sesuai
43	0	0	0	-0,03876	-0,199	sesuai	sesuai
44	0	0	0	-0,03876	-0,199	sesuai	sesuai
45	1	1	0	0,09287	0,800	sesuai	tidak
46	0	1	0	-0,75317	-0,200	tidak	sesuai
47	1	1	0	0,24683	0,661	sesuai	tidak
48	1	1	1	0,24683	0,217	sesuai	sesuai
49	0	1	1	-0,90713	-0,760	tidak	tidak
50	1	1	1	0,09287	-0,009	sesuai	sesuai
51	1	1	1	0,09287	-0,006	sesuai	sesuai
52	0	0	0	-0,03876	-0,199	sesuai	sesuai
53	1	1	1	0,09287	-0,022	sesuai	sesuai
54	1	1	1	0,09287	-0,009	sesuai	sesuai
55	0	1	0	-0,75317	-0,200	tidak	sesuai
56	1	1	0	0,09287	0,794	sesuai	tidak
57	0	0	0	-0,03876	-0,199	sesuai	sesuai
58	0	0	0	-0,03876	-0,203	sesuai	sesuai
59	1	1	1	0,09287	-0,022	sesuai	sesuai
60	0	0	0	-0,03876	-0,199	sesuai	sesuai
61	0	0	0	-0,03876	-0,199	sesuai	sesuai
62	1	1	1	0,09287	-0,022	sesuai	sesuai
63	0	0	0	-0,03876	-0,200	sesuai	sesuai
64	0	0	0	-0,03876	-0,247	sesuai	sesuai
65	1	1	1	0,09287	-0,009	sesuai	sesuai
66	1	1	1	0,24683	0,217	sesuai	sesuai
67	0	0	0	-0,03876	-0,203	sesuai	sesuai
68	1	1	1	0,09287	-0,009	sesuai	sesuai
69	0	0	0	-0,03876	-0,199	sesuai	sesuai
70	0	1	0	-0,75317	-0,301	tidak	sesuai
71	1	1	0	0,09287	0,693	sesuai	tidak
72	1	1	1	0,09287	-0,002	sesuai	sesuai
73	0	0	0	-0,03876	-0,199	sesuai	sesuai
74	0	0	0	-0,03876	-0,199	sesuai	sesuai
75	1	1	1	0,24683	-0,067	sesuai	sesuai
76	1	1	1	0,09287	0	sesuai	sesuai
77	0	0	0	-0,03876	-0,199	sesuai	sesuai
78	1	1	1	0,09287	-0,009	sesuai	sesuai
79	1	0	0	0,88567	0,753	tidak	tidak
80	0	0	0	-0,03876	-0,199	sesuai	sesuai
81	1	1	1	0,09287	-0,009	sesuai	sesuai

## Lampiran 8. (Lanjutan)

82	0	0	0	-0,03876	-0,203	sesuai	sesuai
83	1	1	0	0,24683	0,747	sesuai	tidak
84	0	1	0	-0,75317	-0,247	tidak	sesuai
85	1	1	1	0,09287	-0,002	sesuai	sesuai
86	0	0	0	-0,03876	-0,199	sesuai	sesuai
87	0	0	0	-0,11433	-0,199	sesuai	sesuai
88	1	1	1	0,09287	0,239	sesuai	sesuai
89	0	0	0	-0,03876	-0,203	sesuai	sesuai
90	1	1	1	0,09287	0,000	sesuai	sesuai
91	0	0	0	-0,03876	-0,199	sesuai	sesuai
92	1	1	1	0,09287	-0,044	sesuai	sesuai
93	1	1	1	0,09287	0,239	sesuai	sesuai
94	0	0	0	-0,03876	-0,247	sesuai	sesuai
95	1	1	1	0,09287	0,239	sesuai	sesuai
96	1	1	1	0,24683	0,217	sesuai	sesuai
97	0	0	0	-0,03876	-0,199	sesuai	sesuai
98	1	1	1	0,24683	-0,046	sesuai	sesuai
99	0	1	0	-0,90713	-0,309	tidak	sesuai
100	1	1	1	0,09287	-0,002	sesuai	sesuai
101	0	0	0	-0,03876	-0,203	sesuai	sesuai
102	1	1	1	0,09287	-0,009	sesuai	sesuai
103	0	0	0	-0,03876	-0,199	sesuai	sesuai
104	1	1	1	0,09287	-0,022	sesuai	sesuai
105	1	1	1	0,24683	0,001	sesuai	sesuai
106	0	0	0	-0,03876	-0,199	sesuai	sesuai
107	1	1	1	0,09287	-0,022	sesuai	sesuai
108	0	1	0	-0,75317	-0,206	tidak	sesuai
109	1	1	1	0,09287	-0,022	sesuai	sesuai
110	0	0	0	-0,03876	-0,199	sesuai	sesuai
111	1	1	0	0,09287	0,797	sesuai	tidak
112	0	1	0	-0,75317	-0,206	tidak	sesuai
113	0	0	0	-0,03876	-0,200	sesuai	sesuai
114	1	1	0	0,24683	0,715	sesuai	tidak
115	0	1	0	-0,75317	-0,199	tidak	sesuai
116	1	1	1	0,09287	-0,045	sesuai	sesuai
117	1	1	0	0,24683	0,800	sesuai	tidak
118	0	0	0	-0,03876	-0,199	sesuai	sesuai
119	1	1	0	0,09287	0,693	sesuai	tidak
120	0	1	1	-0,75317	-0,783	tidak	tidak
121	0	1	0	-0,75317	-0,250	tidak	sesuai
122	1	1	1	0,24683	-0,067	sesuai	sesuai
123	1	1	0	0,09287	0,691	sesuai	tidak

## Lampiran 8. (Lanjutan)

124	1	1	1	0,24683	0,217	sesuai	sesuai
125	0	1	0	-0,75317	-0,203	tidak	sesuai
126	0	1	0	-0,75317	-0,202	tidak	sesuai
127	1	1	0	0,24683	0,699	sesuai	tidak
128	0	1	0	-0,75317	-0,199	tidak	sesuai
129	1	1	0	0,09287	0,797	sesuai	tidak
130	0	1	0	-0,75317	-0,199	tidak	sesuai
131	1	1	1	0,09287	-0,022	sesuai	sesuai
132	1	1	0	0,09287	0,721	sesuai	tidak
133	0	0	0	-0,03876	-0,203	sesuai	sesuai
134	1	1	1	0,09287	-0,045	sesuai	sesuai
135	1	1	0	0,24683	0,799	sesuai	tidak
136	0	1	0	-0,75317	-0,199	tidak	sesuai
137	0	0	0	-0,03876	-0,199	sesuai	sesuai
138	1	1	0	0,09287	0,747	sesuai	tidak
139	0	0	0	-0,03876	-0,199	sesuai	sesuai
140	0	1	0	-0,75317	-0,199	tidak	sesuai
141	1	1	0	0,09287	0,800	sesuai	tidak
142	1	1	0	0,24683	0,800	sesuai	tidak
143	0	1	0	-0,90713	-0,199	tidak	sesuai
144	1	1	1	0,24683	0,217	sesuai	sesuai
145	0	1	0	-0,75317	-0,199	tidak	sesuai
146	1	1	0	0,24683	0,800	sesuai	tidak
147	0	0	0	-0,03876	-0,199	sesuai	sesuai
148	0	1	0	-0,75317	-0,199	tidak	sesuai
149	0	1	0	-0,75317	-0,199	tidak	sesuai
150	0	1	0	-0,75317	-0,199	tidak	sesuai

## Lampiran 9. Sintax RBFNN

<pre>%Data input Data=[ ] P=Data(:,2:3:4:5:6:7)' T=Data(:,1)' %Data testing Cek=[] Q=Cek(:,2:3:4:5:6:7)' R=Cek(:,1)'  %Fungsi Radial Basis net=newrb(P,T,0.01,0.5,4,1); net=newrb(P,T,0.01,0.5,6,1); net=newrb(P,T,0.01,0.5,10,1); net=newrb(P,T,0.01,0.5,25,1); net=newrb(P,T,0.01,0.5,50,1); net=newrb(P,T,0.01,0.5,75,1); net=newrb(P,T,0.01,0.5,100,1);  %Menentukan Bobot Awal bobotinput=net.IW{1,1} bobotbiasinput=net.b{1,1} bobotlapisan=net.LW{2,1} bobotbiaslapisan=net.b{2,1}  %Proses Simulasi Training a=sim(net,P) H=[(1:size(P,2))' T' a' T'-a] sprintf('%5d      %9.2f      %9.2f %9.2f\n',H)</pre>	<pre>%Proses Testing b=sim(net,Q) L=[(1:size(Q,2))' R' b' R'-b'] sprintf('%5d %9.2f %9.2f %9.2f\n',L)  %mencari nilai SSE Testing x=(R'-b) e=sse(x)  %Menggambar Grafik k=[1:size(T,2)]' s=[1:size(R,2)]' plot(k,T,'bo-',k,a,'m*-' ) title('Hasil Pengujian Data Training: Target (0), Output (*)') xlabel('Data ke-') ylabel('Target/Output') grid;pause; plottools plot(s,R,'bo-',s,b,'m*-' ) title('Hasil Pengujian Data Testing: Target (0), Output (*)') xlabel('Data ke-') ylabel('Target/Output') grid;pause;</pre>
--	--