

**MODIFIKASI ALGORITMA PEMBELAJARAN
BACKPROPAGATION NEURAL NETWORK
MENGUNAKAN ADAPTIVE GAIN
DAN METODE CONJUGATE GRADIENT**

SKRIPSI

Oleh:
TOMMY ANDRYANTO
0410940056-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



**MODIFIKASI ALGORITMA PEMBELAJARAN
BACKPROPAGATION NEURAL NETWORK
MENGUNAKAN ADAPTIVE GAIN
DAN METODE CONJUGATE GRADIENT**

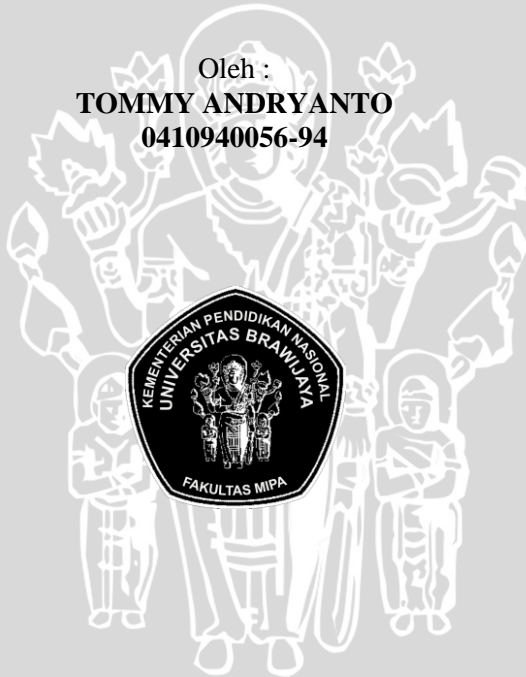
SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Matematika

Oleh :

TOMMY ANDRYANTO

0410940056-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

**MODIFIKASI ALGORITMA PEMBELAJARAN
BACKPROPAGATION NEURAL NETWORK
MENGUNAKAN *ADAPTIVE GAIN*
DAN METODE *CONJUGATE GRADIENT***

Oleh :

TOMMY ANDRYANTO

0410940056-94

Setelah dipertahankan di depan majelis penguji
pada tanggal 2 Februari 2011
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam Bidang Matematika

Dosen Pembimbing I

Dosen Pembimbing II

Syaiful Anam S.Si, MT
NIP. 197801152002121003

Dr. Wuryansari M. K, M.Si
NIP. 196607281993032001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf A., MSc
NIP. 196709071992031001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : TOMMY ANDRYANTO
NIM : 0410940056-94
Jurusan : MATEMATIKA
Penulis Skripsi berjudul : MODIFIKASI ALGORITMA
PEMBELAJARAN
BACKPROPAGATION
NEURAL NETWORK
MENGUNAKAN *ADAPTIVE*
GAIN DAN METODE
CONJUGATE GRADIENT

Dengan ini menyatakan bahwa :

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

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 2 Februari 2011
Yang menyatakan,

(Tommy Andryanto)
NIM. 0410940056-94

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, pertolongan dan petunjuk-Nya sehingga skripsi ini yang berjudul “Modifikasi Algoritma Pembelajaran *Backpropagation* Menggunakan *Adaptive Gain* dan Metode *Conjugate Gradient*” dapat diselesaikan dengan baik. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW sebagai suri tauladan bagi penulis.

Penulis menyadari bahwa penulisan skripsi ini tidak dapat terealisasi tanpa bimbingan dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Syaiful Anam, S.Si, M.T selaku pembimbing I dan Dr. Wuryansari M.K, M.Si selaku pembimbing II atas segala pengarahan, motivasi, nasihat, dukungan dan ilmu yang telah diberikan selama penyusunan skripsi ini.
2. Drs. Imam Nurhadi P., M.T dan Drs. Bambang Sugandi, M.Si, selaku dosen penguji atas segala masukan dan saran yang diberikan untuk perbaikan skripsi ini.
3. Dr.Abdul Rouf A., M.Sc selaku Ketua Jurusan Matematika, dan Dr. Agus Suryanto, M.Sc selaku dosen penasihat akademik dan dosen penguji atas segala pengarahan, motivasi dan ilmu yang telah diberikan selama penulis melaksanakan studi.
4. Ayah dan Ibu atas kasih sayang, dan do'a yang tiada henti demi kesuksesan putranya, juga kakak dan adik-adikku tersayang.
5. Seluruh dosen beserta staf pengajar yang telah mencurahkan ilmunya selama penulis melaksanakan studi.
6. Indah Suryandari dan sahabat-sahabat Matematika angkatan 2004, kakak-kakak tingkat Matematika 2003 serta teman-teman angkatan 2005 dan 2006 atas semua bantuan dan semangat yang diberikan kepada penulis.
7. Segenap staf dan karyawan TU Jurusan Matematika atas segala bantuannya.
8. Semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari masih terdapat kekurangan dalam penulisan skripsi ini. Oleh karena itu, semua kritik dan saran yang membangun sangat penulis harapkan demi perbaikan selanjutnya.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat dan sumbangan yang berarti di masa yang akan datang.

Malang, 2 Februari 2011

Penulis



MODIFIKASI ALGORITMA PEMBELAJARAN *BACKPROPAGATION NEURAL NETWORK* MENGUNAKAN *ADAPTIVE GAIN* DAN METODE *CONJUGATE GRADIENT*

ABSTRAK

Jaringan Saraf Tiruan (JST) merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan saraf otak manusia. Proses pembelajaran pada JST menggunakan model *Backpropagation* untuk mengenali objek yang memiliki pola tertentu. Model ini menggunakan metode *Conjugate Gradient* untuk mengubah bobot-bobotnya. Salah satu bentuk rumusan yang digunakan adalah *Conjugate Gradient Fletcher-Reeves (CGFR)*. Metode ini memiliki kelemahan, yaitu membutuhkan arsitektur yang besar untuk mendapatkan *error* yang kecil. Untuk mengatasinya ditambahkan suatu variabel yang disebut *Adaptive Gain* ke dalam fungsi aktivasi *Backpropagation*, sehingga metode ini disebut metode *Conjugate Gradient Fletcher-Reeves Adaptive Gain (CGFRAG)*. Penambahan *Adaptive Gain* bertujuan untuk meningkatkan laju kekonvergenan model *Backpropagation*. Dalam skripsi ini dikaji kinerja kedua metode tersebut dengan menerapkannya pada proses pembelajaran dan proses simulasi. Proses tersebut menggunakan data penderita diabetes suku Pima Indian. Hasil proses pembelajaran dan simulasi menunjukkan bahwa model *Backpropagation* dengan metode *CGFRAG* lebih baik daripada metode *CGFR* karena metode ini mampu mencapai kekonvergenan dengan *error* yang kecil untuk model arsitektur JST sederhana.

Kata kunci : Jaringan Saraf Tiruan (JST), *Backpropagation*, metode *Conjugate Gradient Fletcher-Reeves (CGFR)*, *Adaptive Gain*, metode *Conjugate Gradient Fletcher-Reeves Adaptive Gain (CGFRAG)*.

UNIVERSITAS BRAWIJAYA



A MODIFIED LEARNING ALGORITHM BACKPROPAGATION NEURAL NETWORK BASED ADAPTIVE GAIN AND CONJUGATE GRADIENT METHOD

ABSTRACT

Artificial Neural Network (ANN) is an information processing system that resembles human brain neural system. *Backpropagation* model is used in the training process to characterize such an object with a certain pattern. The model uses *Conjugate Gradient* method on its weight update process. One of formulas used is *Conjugate Gradient Fletcher-Reeves (CGFR)*. This method has weakness, it uses big architecture to reach small *error*. To solve this problem the *Backpropagation* model is combined with a variable called *Adaptive Gain* on its activation function. This combination is known as *Conjugate Gradient Fletcher-Reeves Adaptive Gain (CGFRAG)* method. The addition of *Adaptive Gain* is purposed to improve the learning rate of *Backpropagation* model. In this final project, performance of those two methods is investigated by applying them on learning process and simulation process. The process are performed to the data of Pima Indian diabetes problem. Learning process and simulation result showed that *CGFRAG* method is better than *CGFR* method because this method can reach the convergence with minimum *error* for simple architecture model of *ANN*.

Key words : *Artificial Neural Network (ANN)*, *Backpropagation*, *Conjugate Gradient Fletcher-Reeves (CGFR)* method, *Adaptive Gain*, *Conjugate Gradient Fletcher-Reeves Adaptive Gain (CGFRAG)* method.

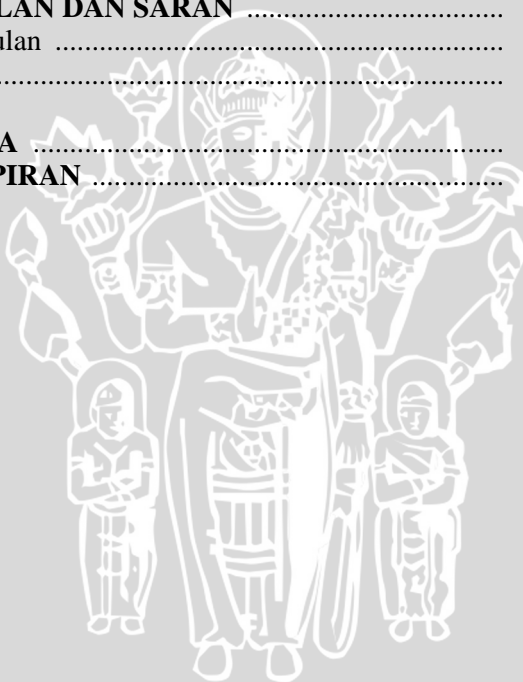
UNIVERSITAS BRAWIJAYA



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
BAB II TINJAUAN PUSTAKA	3
2.1 Optimasi.....	3
2.2 Metode <i>Conjugate Gradient</i>	7
2.2.1 Algoritma <i>CGFR</i>	8
2.3 Jaringan Saraf Tiruan.....	9
2.3.1 Arsitektur JST.....	10
2.3.2 Fungsi Aktivasi.....	12
2.3.3 Proses Pembelajaran.....	14
2.3.4 <i>Backpropagation</i>	14
2.4 JST dengan metode <i>CGFRAG</i>	17
2.4.1 Algoritma <i>CGFRAG</i> pada Perubahan Bobot $w_{(t)}$	18
2.4.2 Algoritma <i>CGFRAG</i> pada Perubahan <i>Adaptive Gain</i> $c_{(t)}$	19
2.5 Diabetes pada Suku Pima Indian	20

BAB III PEMBAHASAN	21
3.1 <i>Backpropagation</i> dengan Metode <i>CGFR</i>	21
3.2 <i>Backpropagation</i> dengan Metode <i>CGFRAG</i>	25
3.3 Membangun JST	32
3.4 Normalisasi Data.....	33
3.5 Hasil Pembelajaran.....	33
3.6 Analisa Hasil Pembelajaran	38
3.7 Pemilihan Model Arsitektur Terbaik.....	40
3.7.1 Nilai MSE	40
3.7.2 Waktu Komputasi.....	40
3.8 Proses Simulasi	41
 BAB IV KESIMPULAN DAN SARAN	 43
4.1 Kesimpulan	43
4.2 Saran	43
 DAFTAR PUSTAKA	 45
LAMPIRAN-LAMPIRAN	47



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur <i>neuron</i> pada JST.....	10
Gambar 2.2 Arsitektur jaringan Lapis Tunggal.....	11
Gambar 2.3 Arsitektur jaringan Lapis Ganda	11
Gambar 2.4 Arsitektur jaringan <i>Recurrent</i>	12
Gambar 2.5 Fungsi Sigmoid Biner	13
Gambar 2.6 Fungsi Sigmoid Bipolar	13
Gambar 3.1 Hasil pembelajaran model 8-3-1 dengan $\eta=0,00005$	35
Gambar 3.2 Hasil pembelajaran model 8-5-1 dengan $\eta=0,00005$	36
Gambar 3.3 Hasil pembelajaran model 8-8-1 dengan $\eta=0,0001$	37
Gambar 3.4 Hasil pembelajaran model 8-12-1 dengan $\eta=0,0001$	38

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

	Halaman
Tabel 2.1 Nilai $\beta_{(t+1)}$ yang digunakan dalam <i>Conjugate Gradient</i>	8
Tabel 3.1 Perbandingan hasil pembelajaran <i>Backpropagation</i> dengan metode <i>CGFR</i> dan <i>CGFRAG</i>	34



UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Nilai bobot dan bias hasil pembelajaran Metode <i>CGFR</i> dengan model 8-12-1	47
Lampiran 2 Nilai bobot dan bias hasil pembelajaran Metode <i>CGFRAG</i> dengan model 8-8-1.....	49
Lampiran 3 Perbandingan hasil simulasi metode <i>CGFR</i> dengan Metode <i>CGFRAG</i>	51
Lampiran 4 Diagram alir model <i>Backpropagation</i> dengan Metode <i>CGFR</i>	59
Lampiran 5 Diagram alir model <i>Backpropagation</i> dengan Metode <i>CGFRAG</i>	65



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang

Jaringan Saraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip jaringan saraf manusia, yang dapat dilatih untuk mengenali suatu objek dengan pola tertentu. Untuk memperoleh JST yang mampu menyelesaikan permasalahan maka perlu dilakukan beberapa langkah antara lain membangun jaringan, melatih jaringan, dan terakhir menguji kemampuan jaringan. JST memiliki beberapa arsitektur jaringan, di antaranya jaringan Lapis Tunggal (*Single Layer Network*), jaringan Lapis Ganda (*Multi Layer Network*) dan jaringan *Recurrent*. Jaringan Lapis Tunggal terdiri dari lapisan *input* dan lapisan *output*, jaringan Lapis Ganda dan *Recurrent* memiliki lapisan tersembunyi (*Hidden Layer*) yang terdapat di antara lapisan *input* dan lapisan *output*, tetapi *neuron* jaringan *Recurrent* saling terhubung membentuk suatu *looping*.

Pelatihan JST menggunakan model *Backpropagation* mengoreksi kesalahan yang muncul dengan memperbaharui nilai bobot hingga diperoleh kesalahan yang minimum. Metode yang sering digunakan dalam model ini adalah metode *Gradient Descent*. Namun metode ini memiliki beberapa kelemahan dalam mengendalikan laju pembelajaran. Laju pembelajaran yang terlalu besar menyebabkan algoritma menjadi tidak stabil. Sebaliknya jika laju pembelajaran yang diberikan terlalu kecil maka algoritma akan sangat lama untuk mencapai konvergensi (Chapra dan Steven, 2002). Laju pembelajaran diperoleh dengan menambahkan suatu perhitungan pada algoritma *Backpropagation* secara dinamis atau langsung ditentukan secara statis.

Beberapa metode telah digunakan untuk meningkatkan laju konvergensi *Backpropagation*. Salah satunya adalah metode *Conjugate Gradient*, yang memiliki nilai kekonvergenan lebih baik daripada metode *Gradient Descent*. Terdapat beberapa metode yang dapat digunakan untuk menentukan arah *gradient* dalam metode *Conjugate Gradient*, antara lain metode *Conjugate Gradient Fletcher-Reeves* (*CGFR*). Dalam perkembangannya, modifikasi algoritma *CGFR* pada titik awal pencarian dengan menambahkan

suatu nilai yang disebut *Adaptive Gain* pada fungsi aktivasi akan meningkatkan kekonvergenan. Modifikasi metode ini disebut *Conjugate Gradient Fletcher-Reeves Adaptive Gain (CGFRAG)* (Nawi dkk, 2008).

Dalam skripsi ini ditunjukkan penerapan metode *CGFR* dan *CGFRAG* dalam algoritma pembelajaran *Backpropagation*. Selanjutnya kedua metode tersebut diaplikasikan pada masalah pengklasifikasian penderita diabetes pada suku Pima Indian untuk dibandingkan hasilnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka permasalahan yang dibahas dalam skripsi ini adalah sebagai berikut.

1. Bagaimana penerapan metode *CGFR* dan *CGFRAG* dalam algoritma pembelajaran *Backpropagation*?
2. Bagaimana hasil perbandingan metode *CGFR* dan *CGFRAG* setelah diaplikasikan untuk pengklasifikasian penderita diabetes pada suku Pima Indian?

1.3 Batasan Masalah

Pembahasan masalah pada skripsi ini dibatasi oleh hal-hal berikut.

1. Arsitektur JST yang dibangun menggunakan satu *hidden layer*.
2. Laju pembelajaran pada masing-masing metode diberikan dan bersifat statis.

1.4 Tujuan

Penulisan skripsi ini bertujuan untuk

1. menerapkan metode *CGFR* dan *CGFRAG* pada algoritma pembelajaran *Backpropagation*,
2. membandingkan kinerja metode *CGFR* dan *CGFRAG* pada algoritma pembelajaran *Backpropagation* setelah diaplikasikan untuk pengklasifikasian penderita diabetes pada suku Pima Indian.

BAB II TINJAUAN PUSTAKA

Modifikasi algoritma *Backpropagation* menggunakan metode *Conjugate Gradient* dan *Adaptive Gain* bertujuan untuk meningkatkan laju kekonvergenan sehingga diperoleh nilai *error* yang minimum. Modifikasi tersebut digunakan dalam pengklasifikasian penderita diabetes pada suku Pima Indian. Akan ditunjukkan tinjauan pustaka yang digunakan dalam algoritma *Backpropagation*, metode *Conjugate Gradient* dan *Adaptive Gain*, serta penjelasan mengenai diabetes secara umum dan diabetes pada suku Pima Indian

2.1 Optimasi

Optimasi merupakan suatu cara untuk meminimumkan atau memaksimumkan fungsi tujuan dengan kendala berdasarkan variabel-variabel yang dicari (Nocedal dan Stephen, 1999). Permasalahan minimasi fungsi n variabel dinyatakan sebagai berikut.

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$$

dengan kendala berbentuk $\begin{cases} g_i(\mathbf{x}) = 0, & i \in \mathfrak{G}, \\ h_i(\mathbf{x}) \leq 0, & i \in \mathfrak{S}, \end{cases}$

dengan f adalah fungsi objektif, g_i dan h_i adalah fungsi bernilai skalar dengan variabel $\mathbf{x} \in \mathbb{R}^n$, \mathfrak{G} dan \mathfrak{S} adalah himpunan indeks.

Definisi 2.1.1 (*Gradient*)

Misalkan $f(\mathbf{x})$ adalah fungsi dari $\mathbf{x} = (x_1, x_2, \dots, x_n) \in S \subseteq \mathbb{R}^n$ sedemikian sehingga $\frac{\partial f(\mathbf{x})}{\partial x_i} = \text{ada } \forall i = 1, 2, \dots, n$. *Gradient* dari $f(\mathbf{x})$ yang dinotasikan dengan $g(\mathbf{x})$, adalah vektor kolom

$$g(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)$$

(Mathews dan Kurtis, 2004).

Contoh. *Gradient* dari $f(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2$ adalah

$$g(\mathbf{x}) = (2x_1 + x_2, 2x_2 + x_1).$$

Definisi 2.1.2 (Minimum Lokal)

Titik $(\mathbf{x}^*, f(\mathbf{x}^*))$, dengan $\mathbf{x}^* \in S \subseteq \mathbb{R}^n$, dikatakan sebagai titik minimum lokal fungsi f jika terdapat $\varepsilon > 0$ sedemikian sehingga $\forall \mathbf{x}^*$, dengan $|\mathbf{x} - \mathbf{x}^*| < \varepsilon$ berlaku $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

Definisi 2.1.3 (Minimum Global)

Titik $(\mathbf{x}^*, f(\mathbf{x}^*))$, dengan $\mathbf{x}^* \in S \subseteq \mathbb{R}^n$, disebut titik minimum global dari fungsi f jika $\forall \mathbf{x}^* \in S$ berlaku $f(\mathbf{x}^*) \leq f(\mathbf{x})$

(Bronshtein dkk, 2007).

Definisi 2.1.4 (Bentuk kuadrat)

Misalkan $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Bentuk

$$[x_1 \ x_2 \ \dots \ x_n] A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

dengan A matriks simetrik berukuran $n \times n$ disebut bentuk kuadrat (Anton, 1987).

Contoh. Bentuk $x_1^2 + 2x_2^2 + 4x_3^2 - 4x_1x_2 + 6x_1x_3 + 2x_2x_3$ adalah bentuk kuadrat sebab dapat dituliskan sebagai

$$[x_1 \ x_2 \ x_3] \begin{bmatrix} 1 & -2 & 3 \\ -2 & 2 & 1 \\ 3 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Definisi 2.1.5 (Definit Positif)

Bentuk kuadrat $\mathbf{x}^T A \mathbf{x}$ dikatakan bersifat definit positif jika $\mathbf{x}^T A \mathbf{x} > 0$ untuk semua $\mathbf{x} \neq 0$, sedangkan matriks simetrik A disebut matriks definit positif jika $\mathbf{x}^T A \mathbf{x}$ adalah bentuk kuadrat definit positif (Anton, 1987).

Definisi 2.1.6 (Minor)

Misalkan A matriks berukuran $K \times K$. Minor M_{ij} didefinisikan sebagai determinan submatriks yang diperoleh dengan menghilangkan elemen-elemen baris ke- i dan kolom ke- j matriks A (Anton, 1987).

Contoh : misalkan

$$\text{matriks } A = \begin{bmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{bmatrix}, \text{ maka}$$

$$M_{11} = \begin{vmatrix} \cancel{2} & \cancel{-1} & \cancel{-3} \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{vmatrix} = \begin{vmatrix} 2 & 4 \\ -1 & 9 \end{vmatrix} = 2,$$

$$M_{23} = \begin{vmatrix} 2 & -1 & \cancel{-3} \\ \cancel{-1} & \cancel{2} & \cancel{4} \\ -3 & 4 & 9 \end{vmatrix} = \begin{vmatrix} 2 & -1 \\ -3 & 4 \end{vmatrix} = 5.$$

Definisi 2.1.7 (Determinan)

Misalkan A matriks berukuran $K \times K$, determinan matriks $A = [a_{ij}]$ didefinisikan sebagai

$$\det(A) = |A| = \sum_{j=1}^k a_{ij} (-1)^{i+j} |M_{ij}|,$$

dengan $(-1)^{i+j} |M_{ij}|$ adalah kofaktor elemen a_{ij} dan M_{ij} merupakan minor elemen a_{ij} (Anton, 1987).

Teorema 2.1.8

Matriks simetrik A adalah definit positif jika dan hanya jika determinan setiap submatriks adalah positif.

Definisi 2.1.9 (Sub Matriks)

Submatriks ke- i dari A , yaitu A_i , adalah matriks yang diperoleh dari i baris pertama dan i kolom pertama matriks A (Anton, 1987).

Contoh : matriks $A = \begin{bmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{bmatrix}$ adalah matriks definit

positif sebab determinan submatriks-submatriksnya, yaitu

$$A_1 = |2| = 2, A_2 = \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} = 3 \text{ dan } A_3 = \begin{vmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{vmatrix} = 1,$$

semua bernilai positif.

Definisi 2.1.10 (Normalisasi Data)

Normalisasi bertujuan untuk menghindari pemrosesan data dengan nilai yang dapat menyebabkan perhitungan menjadi lama. Terdapat beberapa macam Normalisasi, di antaranya

1. Normalisasi *Z-Score*, yang berbentuk

$$x^* = \frac{x - \bar{x}}{\sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}}}$$

dengan x^* adalah data yang telah dinormalisasikan,

x adalah data sebelum dinormalisasikan,

\bar{x} adalah rata-rata dari seluruh data,

n adalah banyak data.

2. Normalisasi Desimal *Scaling*, merupakan normalisasi yang mengubah data ke dalam bentuk desimal terbesar, yaitu

$$x^* = \frac{x}{10^j}, \text{ dengan } j \in \mathbb{Z} \ni |x^*| < 1$$

(Al Shalabi dkk, 2006).

Misalkan $x=13$, maka bentuk desimal terbesarnya adalah

$x^*=0,13$ dengan $j = 2$.

Normalisasi yang digunakan dalam skripsi ini adalah Desimal *Scaling*.

2.2 Metode *Conjugate Gradient*

Metode *Conjugate Gradient* merupakan salah satu metode yang digunakan untuk memperoleh nilai optimum suatu fungsi. Pada umumnya metode *Conjugate Gradient* dapat digunakan untuk menyelesaikan permasalahan optimasi linier maupun optimasi nonlinier (Nocedal dan Setphen, 1999).

Misalkan akan diselesaikan masalah minimasi suatu fungsi n variabel, yaitu

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2.1)$$

dengan $f: \mathbb{R}^n \rightarrow \mathbb{R}$ merupakan fungsi kontinu dan *gradient*-nya $g(x) = \nabla f(x)$. Metode *Conjugate Gradient* untuk menyelesaikan (2.1) menggunakan metode iterasi yaitu

$$x_{(t+1)} = x_{(t)} + \eta d_{(t)},$$

dengan $x_{(t)}$ menyatakan hasil iterasi ke- t , η adalah *steep size* dan $d_{(t)}$ menyatakan arah pencarian. Misalkan $g_{(t)} = g(x_{(t)})$, maka arah pencarian pada iterasi pertama menggunakan arah *Gradient Descent* yaitu

$$d_{(0)} = -g_{(0)}.$$

Kemudian arah pencarian selanjutnya adalah

$$d_{(t+1)} = -g_{(t+1)} + \beta_{(t+1)} d_{(t)},$$

dengan $\beta_{(t+1)}$ adalah *steep size* dalam *Conjugate Gradient* yang bernilai skalar. Nilai $\beta_{(t+1)}$ bergantung pada metode yang digunakan sebagaimana diberikan pada Tabel 2.1.

Tabel 2.1 Nilai $\beta_{(t+1)}$ yang digunakan dalam *Conjugate Gradient* (Al Bayati dkk, 2009).

Metode	$\beta_{(t+1)}$
<i>Fletcher-Reeves</i>	$\beta_{(t+1)} = \frac{\mathbf{g}_{(t+1)}^T \mathbf{g}_{(t+1)}}{\mathbf{g}_{(t)}^T \mathbf{g}_{(t)}}$
<i>Polak-Ribere</i>	$\beta_{(t+1)} = \frac{\mathbf{g}_{(t+1)}^T (\mathbf{g}_{(t+1)} - \mathbf{g}_{(t)})}{\mathbf{g}_{(t)}^T \mathbf{g}_{(t)}}$
<i>Al Assady-Al Bayati</i>	$\beta_{(t+1)} = \frac{\mathbf{g}_{(t+1)}^T (\mathbf{g}_{(t+1)} - \mathbf{g}_{(t)})}{\mathbf{d}_{(t)}^T \mathbf{g}_{(t)}}$
<i>Hestene Steifel</i>	$\beta_{(t+1)} = \frac{\mathbf{g}_{(t+1)}^T (\mathbf{g}_{(t+1)} - \mathbf{g}_{(t)})}{\mathbf{d}_{(t)}^T (\mathbf{g}_{(t+1)} - \mathbf{g}_{(t)})}$

2.2.1 Algoritma CGFR

Sesuai Tabel 2.1, dalam metode *CGFR* digunakan

$\beta_{(t+1)} = \frac{\mathbf{g}_{(t+1)}^T \mathbf{g}_{(t+1)}}{\mathbf{g}_{(t)}^T \mathbf{g}_{(t)}}$, sehingga diperoleh algoritma *CGFR* sebagai berikut.

Input : $x_{(0)}$, η , t_{maks} .

Output : $d_{(t)}$.

Langkah-langkah.

Langkah 1. $t=0$.

2. Hitung $\mathbf{g}_{(0)} = \nabla f(\mathbf{x})$.

3. Hitung $\mathbf{d}_{(0)} = -\mathbf{g}_{(0)}$.

4. Selama $t \leq t_{maks}$ lakukan langkah 5 sampai dengan langkah 9.

5. Hitung $\mathbf{x}_{(t+1)} = \mathbf{x}_{(t)} + \eta \mathbf{d}_{(t)}$.

6. Hitung $\mathbf{g}_{(t+1)}$.

$$7. \text{ Hitung } \beta_{(t+1)} = \frac{g_{(t+1)}^T g_{(t+1)}}{g_{(t)}^T g_{(t)}}.$$

$$8. \text{ Hitung } d_{(t+1)} = -g_{(t+1)} + \beta_{(t+1)} d_{(t)}.$$

$$9. t = t + 1.$$

Nilai η harus memenuhi kondisi Wolfe kuat

$$f(x_{(t)}) - f(x_{(t+1)}) \geq -a g_{(t)}^T d_{(t)},$$

$$|g_{(t+1)}^T d_{(t)}| \leq b |g_{(t)}^T d_{(t)}|$$

dengan $0 < a < b < \frac{1}{2}$ (Nocedal dan Stephen, 1999).

Garis pencarian membutuhkan kondisi turunan

$$|g_{(t)}^T d_{(t)}| < 0,$$

dengan t adalah iterasi (Boyd dan Lieven, 2004).

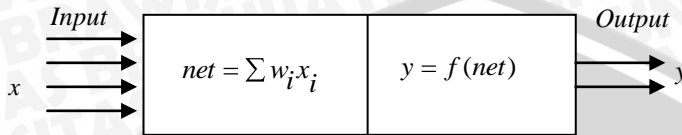
2.3 Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) didefinisikan sebagai representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia (Kusumadewi dan Hartati, 2006).

JST dikembangkan berdasarkan model matematis sebagai berikut.

1. Informasi diproses oleh elemen-elemen sederhana yang disebut *neuron*.
2. Sinyal-sinyal dilewatkan melalui *neuron* yang saling berhubungan.
3. Setiap sambungan antara dua *neuron* memiliki bobot yang akan menggandakan sinyal yang ditransmisikan.
4. Setiap *neuron* memiliki fungsi aktivasi yang akan menentukan besaran *output* (Muis, 2006).

Struktur JST ditunjukkan oleh Gambar 2.1 yaitu



Gambar 2.1 Struktur *neuron* pada JST

dengan x adalah *input*,
 y adalah *output*,
 w adalah bobot,
 i adalah banyaknya *neuron input*,
 $f(net)$ adalah fungsi aktivasi.

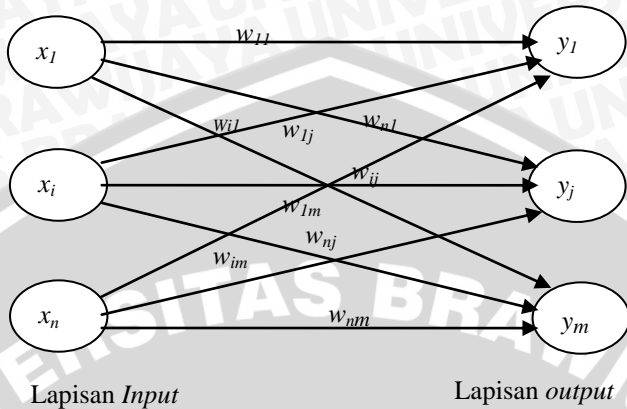
Gambar 2.1 menjelaskan tentang kinerja JST sederhana. Setiap *input* (x_i) yang masuk ke dalam jaringan akan dikalikan dengan masing-masing nilai bobot (w_i). Perkalian dari seluruh *input* (x) dan bobot (w) dijumlahkan, kemudian disebut *net*. Nilai *net* dimasukkan ke dalam fungsi aktivasi sehingga menghasilkan nilai *output* (y).

2.3.1 Arsitektur JST

Dalam arsitektur JST terdapat beberapa model di antaranya :

a. Jaringan Lapis Tunggal (*Single Layer Network*).

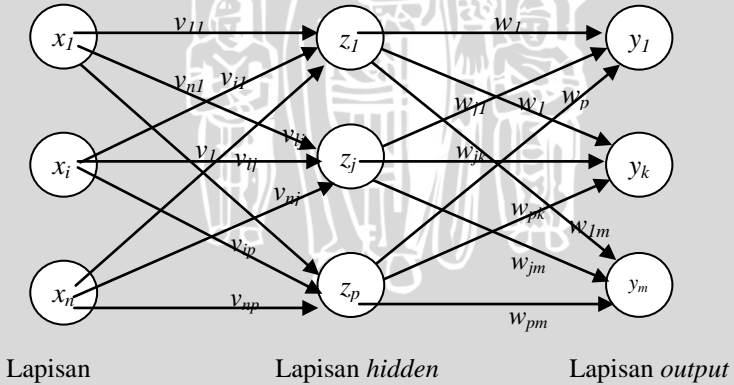
Jaringan Lapis Tunggal dikelompokkan menjadi dua lapisan yaitu lapisan *input* dan lapisan *output*. Gambar 2.2 menunjukkan lapisan *input* yang memiliki n *neuron* dan lapisan *output* yang memiliki m *neuron*.



Gambar 2.2 Arsitektur jaringan Lapis Tunggal

b. Jaringan Lapis Ganda (*Multi Layer Network*).

Jaringan Lapis Ganda merupakan perluasan dari jaringan Lapis Tunggal. Di antara lapisan *input* dan lapisan *output* ditambahkan lapisan lain yang disebut sebagai lapisan tersembunyi (*hidden layer*), yang jumlahnya bisa lebih dari satu. Gambar 2.3 menunjukkan jaringan yang memiliki n neuron *input* dan m neuron *output* serta sebuah lapisan *hidden* yang terdiri dari p buah neuron.

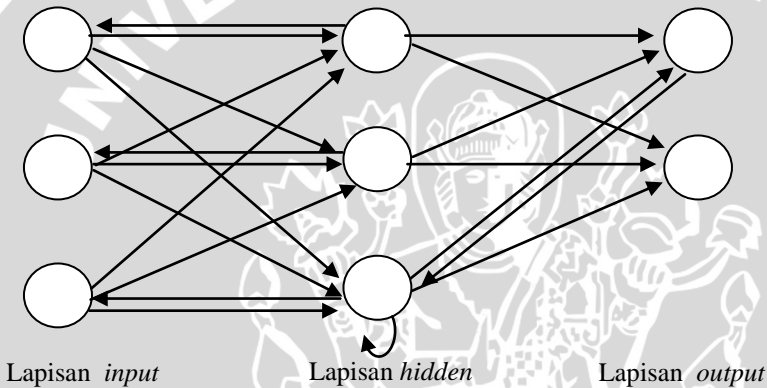


Gambar 2.3 Arsitektur jaringan Lapis Ganda

Dalam skema tersebut z adalah *hidden*, w_{jk} adalah bobot antara lapisan *hidden* dan lapisan *output*, dan v_{ij} adalah bobot antara lapisan *input* dan lapisan *hidden*.

c. Jaringan *Recurrent*.

Arsitektur jaringan *recurrent* memiliki bentuk yang berbeda dari arsitektur jaringan Lapis Tunggal maupun jaringan Lapis Ganda. Setiap *neuron* dalam jaringan *recurrent* dapat saling dihubungkan sehingga terbentuk suatu *loop*. *Loop* terbentuk antara *neuron* lapisan *input* dengan *hidden*, serta *neuron* lapisan *hidden* dengan *output*.



Gambar 2.4 Arsitektur jaringan *recurrent*

2.3.2 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang digunakan untuk menghitung nilai masukan tiap *neuron* dalam JST. Bentuk fungsi aktivasi adalah sebagai berikut.

$$f(net) = f(\sum x_i w_{ij} + w_{0j}),$$

dengan w_0 adalah bias.

Dalam JST dikenal tiga fungsi aktivasi, yaitu :

1. fungsi Tangga

$$f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ 0 & \text{jika } net < 0 \end{cases}$$

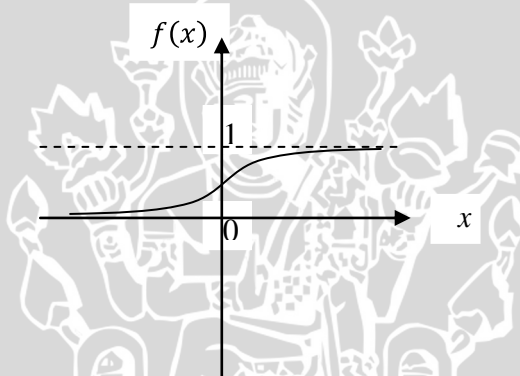
2. fungsi Sigmoid, yang dapat dibedakan menjadi dua jenis yaitu fungsi Sigmoid Biner dan fungsi Sigmoid Bipolar. Bentuk fungsi Sigmoid Biner adalah

$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}}},$$

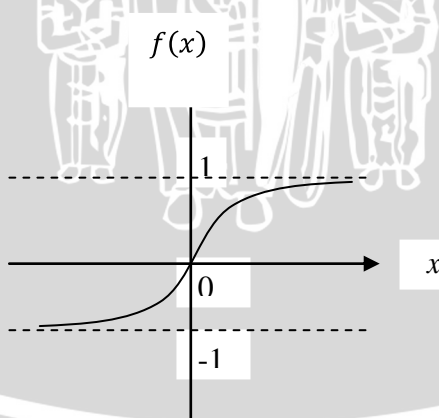
dan memiliki nilai pada selang $[0,1]$. Fungsi ini paling sering digunakan dalam JST. Fungsi Sigmoid Bipolar berbentuk

$$f(\text{net}) = \frac{2}{1 + e^{-\text{net}}} - 1,$$

dan memiliki nilai pada selang $[-1,1]$. Grafik kedua fungsi Sigmoid tersebut diperlihatkan pada Gambar 2.5 dan 2.6.



Gambar 2.5 Fungsi Sigmoid Biner



Gambar 2.6 Fungsi Sigmoid Bipolar

3. fungsi Identitas, yang berbentuk $f(net) = net$, dan digunakan bila nilai *output* jaringan yang diinginkan berupa sembarang bilangan riil (Muis, 2006).

2.3.3 Proses Pembelajaran

JST memiliki beberapa macam pembelajaran untuk memodifikasi bobotnya, yaitu pembelajaran terawasi dan pembelajaran tidak terawasi (Muis, 2006).

1. Pembelajaran Terawasi.

Metode pembelajaran pada JST disebut terawasi jika nilai *output* yang diharapkan telah diketahui sebelumnya. Nilai ini disebut nilai target. Jika terdapat selisih antara nilai *output* pembelajaran dan nilai target maka terjadi *error*. Jaringan akan melakukan perbaikan terhadap bobot sesuai *error* yang telah terjadi (Kusumadewi dan Hartati, 2006).

2. Pembelajaran Tidak Terawasi.

Pembelajaran tidak terawasi tidak memerlukan nilai target *output*, sehingga tidak dapat ditentukan hasil seperti apa yang diharapkan selama proses pembelajaran. Nilai bobot disusun dalam suatu batasan tertentu yang bergantung pada nilai *input* yang diberikan (Kusumadewi dan Hartati, 2006). Skripsi ini menggunakan JST dengan pembelajaran terawasi.

2.3.4 Backpropagation

Suatu proses pembelajaran dikatakan baik jika *output* jaringan mendekati target. Dengan demikian, untuk memperoleh proses pembelajaran yang baik perlu diminimumkan *error* antara *output* dan target, sehingga diperoleh suatu masalah minimasi dengan fungsi tujuan

$$E_s = \frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2, \quad (2.3)$$

yang merupakan *error* antara *output* dan target data ke- s , T_k adalah target, y_k adalah *output* jaringan, dan m adalah jumlah *neuron output*.

Output jaringan didapatkan melalui perhitungan

$$y_k = f(\text{net}y_k), \quad (2.4)$$

dengan $f(\text{net}y_k)$ adalah fungsi aktivasi yang digunakan. $\text{Net}y_k$ didefinisikan sebagai berikut.

$$\text{net}y_k = w_{0k} + \sum_{j=1}^p z_j w_{jk},$$

dengan w_{jk} adalah bobot antara *neuron* ke- j lapisan *hidden* dan *neuron* ke- k lapisan *output*, w_{0k} adalah bias *neuron* ke- k lapisan *output*. z_j merupakan nilai masukan pada *neuron* ke- j lapisan *hidden* yang berbentuk

$$z_j = f(\text{net}z_j), \quad (2.5)$$

di mana $f(\text{net}z_j)$ merupakan fungsi aktivasi yang digunakan.

$\text{Net}z_j$ didefinisikan sebagai

$$\text{net}z_j = v_{0j} + \sum_{i=1}^n x_i v_{ij},$$

dengan v_{ij} adalah bobot antara *neuron* ke- i lapisan *input* dan *neuron* ke- j lapisan *hidden*, v_{0j} adalah bias *neuron* ke- j lapisan *hidden*, dan x merupakan data pembelajaran pada *neuron* ke- i lapisan *input*. Model pembelajaran *Backpropagation* memiliki kriteria penghentian yang didefinisikan sebagai *Mean Square Error* (MSE) yaitu

$$MSE = \frac{\sum_{s=1}^r E_s}{r}, \quad (2.6)$$

dengan r adalah jumlah data yang dilatih (Patterson, 1996).

Pelatihan *Backpropagation* terdiri dari tiga tahap yaitu propagasi maju (*Feedforward*), propagasi mundur (*Backward propagation*) dan perubahan bobot.

Algoritma pembelajaran *Backpropagation* adalah sebagai berikut.

Input : Banyak data (s_{maks}), data pembelajaran (x), data target (T), banyak *neuron input* (n), banyak *neuron hidden* (p), banyak *neuron output* (k), $v, v_0, w, w_0, \eta, \text{iterasi}_{maks}, \text{error}_{maks}$.

Output : MSE, iterasi.

Langkah-langkah.

Langkah 1. $s=1, \text{MSE}=1, \text{iterasi}=0$.

2. Selama ($\text{iterasi} < \text{iterasi}_{maks}$) dan ($\text{MSE} > \text{error}_{maks}$),

kerjakan langkah 3 sampai dengan langkah 19.

3. Selama ($s < s_{maks}$), kerjakan langkah 4 sampai dengan langkah 18.

Feedforward :

4. Hitung nilai awal lapisan *hidden* berdasarkan v, v_0 , dan x

$$\text{net}z_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}, \quad j=0,1,\dots,p.$$

5. Hitung nilai akhir lapisan *hidden* berdasarkan fungsi aktivasi yang digunakan

$$z_j = f(\text{net}z_j).$$

6. Hitung nilai awal *neuron output* berdasarkan w, w_0 , dan z

$$\text{net}2_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}, \quad k=1,2,\dots,m.$$

7. Hitung nilai akhir *neuron output* berdasarkan fungsi aktivasi yang digunakan

$$y_k = f(\text{net}2_k).$$

8. Hitung MSE berdasarkan persamaan (2.6).

Backward propagation :

9. Hitung faktor δ pada *neuron output* yang berbentuk

$$\delta_k = (t_k - y_k)f'(\text{net}2_k).$$

10. Hitung perubahan bobot w_{jk} yang berbentuk

$$\Delta w_{jk} = \eta \delta_k z_j.$$

11. Hitung perubahan bobot bias w_{0k} yang berbentuk

$$\Delta w_{0k} = \eta \delta_k.$$

12. Hitung masukan masing-masing *neuron* lapisan *hidden*

$$\gamma_{in_j} = \sum_{k=1}^m \delta_k w_{jk}.$$

13. Hitung informasi kesalahannya

$$\gamma_j = \gamma_{in_j} f'(netz_j).$$

14. Hitung perubahan bobot v_{ij} yang berbentuk

$$\Delta v_{ij} = \eta \gamma_j x_i$$

15. Hitung perubahan bobot v_{0j} yang berbentuk

$$\Delta v_{0j} = \eta \gamma_j, \quad i = 0, 1, \dots, n.$$

Perubahan Bobot dan Bias:

16. Perubahan bobot dan bias dari setiap *neuron output* berbentuk

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}.$$

17. Perubahan bobot dan bias dari setiap *neuron hidden* berbentuk

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}.$$

$$18. s = s + 1.$$

$$19. \text{Iterasi} = \text{iterasi} + 1.$$

2.4 JST dengan Metode CGFRAG

Metode CGFRAG merupakan modifikasi algoritma *CGFR* dengan ditambahkan variabel *Adaptive Gain* (c_i) pada fungsi aktivasi (Nawi dkk, 2008). *Gradient* persamaan (2.1) didefinisikan $g(\mathbf{x}) = \nabla f(\mathbf{x})$. Misalkan $g_{(t)} = g(\mathbf{x}_{(t)})$, maka arah pencarian pada iterasi pertama metode *CGFRAG* adalah sebagai berikut.

$$d_{(0)} = -g_{(0)},$$

Kemudian pada iterasi $(t+1)$ dihitung menggunakan persamaan

$$d_{(t+1)} = -g_{(t+1)} + \beta_{(t+1)} d_{(t)},$$

Modifikasi *Backpropagation* diperoleh dengan menambahkan *Adaptive Gain* pada persamaan (2.4) dan (2.5), sehingga diperoleh fungsi aktivasi masing-masing *neuron* lapisan *output* yaitu $y_k = f(c2_k \cdot nety_k)$, dan fungsi aktivasi masing-masing *neuron* lapisan *hidden* $z_j = f(c1_j \cdot netz_j)$. Variabel $c1_j$ merupakan *Adaptive Gain* pada fungsi aktivasi *neuron* ke- j lapisan *hidden* dan $c2_k$ merupakan *Adaptive Gain* pada fungsi aktivasi *neuron* ke- k lapisan *output*.

Algoritma *CGFRAG* diaplikasikan ke dalam algoritma *Backpropagation* untuk mengubah bobot dan *Adaptive Gain* sebagai berikut.

2.4.1 Algoritma *CGFRAG* untuk mengubah Bobot $w_{(t)}$

Perubahan bobot $w_{(t)}$ diperoleh dengan menurunkan persamaan (2.3) terhadap $w_{(t)}$ yang berbentuk

$$g_{(t)}(w_{(t)}) = \frac{\partial E}{\partial w_{(t)}}.$$

Algoritma *CGFRAG* untuk mengubah bobot $w_{(t)}$ adalah sebagai berikut.

Input : $g_{(0)}, w_{(0)}, c_{(0)}, \text{MSE}, error_{maks}$.

Output : $w_{(t)}$.

Langkah-langkah.v

Langkah 1. $t = 0, \beta_{(0)} = 0$.

2. Hitung nilai $g_{(t)}(w_{(t)})$.

3. Jika $(t < t_{maks})$ dan $(\text{MSE} > error_{maks})$ lakukan

langkah 4 sampai dengan langkah 10.

4. Jika $(t = 0)$ maka $d_{(t)} = g_{(t)}(w_{(t)})$.

5. Jika $(t > 0)$ maka lakukan langkah 6 sampai dengan langkah 8.
6. Hitung $\beta(t) = \frac{g_{(t)}^T(w_{(t)})g_{(t)}(w_{(t)})}{g_{(t-1)}^T(w_{(t-1)})g_{(t-1)}(w_{(t-1)})}$.
7. Jika $g_{(t-1)}^T(w_{(t-1)})g_{(t-1)}(w_{(t-1)}) \approx 0$ maka $\beta(t) = 0$.
8. Hitung nilai $w_{(t+1)} = w_{(t)} - \eta d_{(t)}$.
9. Hitung $d_{(t)} = -g_{(t)}(w_{(t)}) + \beta(t)d_{(t-1)}$.
10. $t = t+1$.

2.4.2 Algoritma CGFRAG untuk mengubah Adaptive Gain $c_{(t)}$

Perubahan Adaptive Gain $c_{(t)}$ diperoleh dengan menurunkan persamaan (2.3) terhadap Adaptive Gain $c_{(t)}$ yang berbentuk

$$g_{(t)}(c_{(t)}) = \frac{\partial E}{\partial c_{(t)}}$$

Algoritma CGFRAG untuk mengubah bobot $c_{(t)}$ adalah sebagai berikut.

Input : $g_{(0)}, w_{(0)}, c_{(0)}, \text{MSE}, \text{error}_{maks}$.

Output : $c_{(t)}$.

Langkah-langkah.v

Langkah 1. $t = 0, \beta_{(0)} = 0$.

2. Hitung nilai $g_{(t)}(c_{(t)})$.

3. Jika $(t < t_{maks})$ dan $(\text{MSE} > \text{error}_{maks})$ lakukan langkah 4 sampai dengan langkah 10.

4. Jika $(t = 0)$ maka $d_{(t)} = g_{(t)}(c_{(t)})$.

5. Jika $(t > 0)$ maka lakukan langkah 6 sampai dengan langkah 8.

6. Hitung $\beta(t) = \frac{g_{(t)}^T(c_{(t)})g_{(t)}(c_{(t)})}{g_{(t-1)}^T(c_{(t-1)})g_{(t-1)}(c_{(t-1)})}$.

7. Jika $g_{(t-1)}^T(c_{(t-1)})g_{(t-1)}(c_{(t-1)}) \approx 0$ maka $\beta_{(t)} = 0$.
8. Hitung nilai $w_{(t+1)} = w_{(t)} - \eta d_{(t)}$.
9. Hitung $d_{(t)} = -g_{(t)}(c_{(t)}) + \beta_{(t)}d_{(t-1)}$.
10. $t = t+1$.

2.5 Diabetes pada Suku Pima Indian

Diabetes *Mellitus* adalah gangguan akibat ketidakmampuan tubuh dalam mengubah glukosa menjadi energi. Diabetes *Mellitus* disebabkan oleh rendahnya produksi hormon insulin dalam pankreas atau akibat penurunan daya kerja insulin. Hal tersebut dikarenakan kekurangan nutrisi akibat rendahnya kadar kalsium dalam tubuh. Seseorang dikatakan menderita diabetes bila kadar gula darahnya melebihi batas normal, yaitu sebesar 140 mg/dl (Martin, 2009).

Penentuan arsitektur jaringan pada proses pembelajaran dan simulasi menggunakan data diabetes pada suku Pima Indian sebagai variabel-variabel *input* dan target JST. Data tersebut adalah sebagai berikut .

1. Umur.
2. Waktu tenggang melahirkan.
3. Indeks massa tubuh (berat kg/ tinggi m²).
4. Toleransi glukosa darah dalam plasma darah (mg/dl).
5. Tekanan darah *diastolic* (mmHg).
6. Ketebalan lipatan kulit lengan bawah (mm).
7. 2-jam serum insulin (mu U/ml).
8. Fungsi silsilah diabetes.
9. *Class Variable* (0 atau 1) (Sigillito, 1990).

Data diabetes pada Suku Pima Indian yang digunakan sebanyak 768 kasus, dengan 500 data digunakan sebagai nilai *input* proses pembelajaran *Backpropagation* dan 268 data digunakan pada proses simulasi. Proses pembelajaran bertujuan untuk mendapatkan informasi tentang model JST terbaik, bobot, bias dan nilai *Adaptive Gain*, di mana informasi tersebut digunakan dalam proses simulasi. Hasil simulasi menunjukkan perbandingan nilai target dan nilai *output* simulasi, sehingga dapat diketahui proses tersebut dapat mengklasifikasikan seseorang terkena diabetes atau tidak. Nilai awal dan keterangan tentang data diabetes tersebut terdapat pada situs <ftp://ftp.ics.uci.edu/pub/machine-learningdatabases/diabetes>.

BAB III HASIL DAN PEMBAHASAN

3.1 *Backpropagation* dengan metode *CGFR*

Metode *CGFR* dalam model pembelajaran *Backpropagation* bertujuan untuk meminimumkan *error* pada proses pembelajaran dengan mengubah bobot dan bias masing-masing lapisan. Perubahan bobot dan bias didapatkan dengan menurunkan persamaan (2.3) dan menggunakan algoritma *CGFR* sebagai berikut.

1. Perubahan bobot antara lapisan *output* dan lapisan *hidden*.

a) Perubahan bobot $w_{jl}, j=1, \dots, p; l=1, \dots, m$

Perubahan bobot w_{jl} diperoleh dengan menurunkan persamaan (2.3) terhadap w_{jl} .

$$\begin{aligned} \frac{\partial E_s}{\partial w_{jl}} &= \frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_l} \frac{\partial y_l}{\partial \text{net}_l} \frac{\partial \text{net}_l}{\partial w_{jl}} \\ \frac{\partial E_s}{\partial w_{jl}} &= \frac{\partial}{\partial y_l} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial \text{net}_l} (f(\text{net}_l)) \\ &\quad \frac{\partial}{\partial w_{jl}} \left(w_{0l} + \sum_{j=1}^p w_{jl} z_j \right) \\ \frac{\partial E_s}{\partial w_{jl}} &= -(T_l - y_l) f'(\text{net}_l) z_j, j \neq 0. \end{aligned} \quad (3.1)$$

$$\text{Misalkan } \delta_l = (T_l - y_l) f'(\text{net}_l) z_j. \quad (3.2)$$

Substitusi persamaan (3.2) ke persamaan (3.1) menghasilkan

$$\frac{\partial E_s}{\partial w_{jl}} = -\delta_l z_j,$$

$$g_{w_{jl}} = \frac{\partial E_s}{\partial w_{jl}}.$$

Berdasarkan algoritma CGFR didapatkan

$$d_{w_{jl}(t)} = -g_{w_{jl}(t)}, \text{ untuk } t=0,$$

$$d_{w_{jl}(t)} = -g_{w_{jl}(t)} + \beta_{w_{jl}(t)} d_{w_{jl}(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{w_{jl}(t+1)} = \frac{g_{w_{jl}(t)}^T g_{w_{jl}(t)}}{g_{w_{jl}(t-1)}^T g_{w_{jl}(t-1)}}.$$

Perubahan bobot w_{jl} pada iterasi ke-($t+1$) adalah sebagai berikut.

$$w_{jl(t+1)} = w_{jl(t)} + \eta d_{w_{jl}(t)}.$$

b) Perubahan bias w_{0l} , $l=1, \dots, m$

Perubahan bias w_{0l} diperoleh dengan menurunkan persamaan (2.3) terhadap w_{0l} .

$$\begin{aligned} \frac{\partial E_s}{\partial w_{0l}} &= \frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_l} \frac{\partial y_l}{\partial \text{net}_l} \frac{\partial \text{net}_l}{\partial w_{0l}} \\ &= \frac{\partial}{\partial y_l} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial \text{net}_l} (f(\text{net}_l)) \\ &\quad \frac{\partial}{\partial w_{0l}} \left(w_{0l} + \sum_{j=1}^p w_{jl} z_j \right). \end{aligned}$$

$$\frac{\partial E_s}{\partial w_{0l}} = -(T_l - y_l) f'(\text{net}_l). \quad (3.3)$$

Substitusikan persamaan (3.2) ke persamaan (3.3), sehingga diperoleh

$$g_{w_{0l}} = \frac{\partial E_s}{\partial w_{0l}} = -\delta_l.$$

Berdasarkan algoritma CGFR diperoleh

$$d_{w_{0l}(t)} = -g_{w_{0l}(t)}, \text{ untuk } t=0,$$

$$d_{w_{0l}(t)} = -g_{w_{0l}(t)} + \beta_{w_{0l}(t)} d_{w_{0l}(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{w_{0l}(t+1)} = \frac{g_{w_{0l}(t)}^T g_{w_{0l}(t)}}{g_{w_{0l}(t-1)}^T g_{w_{0l}(t-1)}}.$$

Perubahan bias w_{0l} pada iterasi ke- $(t+1)$ adalah sebagai berikut.

$$w_{0l(t+1)} = w_{0l(t)} + \eta d_{w_{0l}(t)}.$$

2. Perubahan bobot antara lapisan *hidden* dan lapisan *input*

a) Perubahan bobot v_{ij} , $i=1, \dots, n$; $j=1, \dots, p$;

Perubahan bobot v_{ij} diperoleh dengan menurunkan persamaan (2.3) terhadap v_{ij} .

$$\begin{aligned} \frac{\partial E_s}{\partial v_{ij}} &= \sum_{k=1}^m \left(\frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_k} \frac{\partial y_k}{\partial \text{net}y_k} \frac{\partial \text{net}y_k}{\partial z_j} \frac{\partial z_j}{\partial \text{net}z_j} \frac{\partial \text{net}z_j}{\partial v_{ij}} \right) \\ &= \sum_{k=1}^m \left(\frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \right) \frac{\partial}{\partial \text{net}y_k} (f(\text{net}y_k)) \\ &\quad \frac{\partial}{\partial z_j} \left(w_{0k} + \sum_{j=1}^p w_{jk} z_j \right) \frac{\partial}{\partial \text{net}z_j} (f(\text{net}z_j)) \\ &\quad \frac{\partial}{\partial v_{ij}} \left(v_{0j} + \sum_{i=1}^n v_{ij} x_i \right). \end{aligned}$$

$$\frac{\partial E_s}{\partial v_{ij}} = - \sum_{k=1}^m \left((T_k - y_k) f'(\text{net}y_k) w_{jk} \right) f'(\text{net}z_j) x_i, i \neq 0 \quad (3.4)$$

Misalkan

$$\gamma_j = \sum_{k=1}^m \left((T_k - y_k) f'(\text{net}y_k) w_{jk} \right) f'(\text{net}z_j). \quad (3.5)$$

Substitusi persamaan (3.5) pada persamaan (3.4), menghasilkan

$$g_{v_{ij}} = \frac{\partial E_s}{\partial v_{ij}} = -\gamma_j x_i.$$

Berdasarkan algoritma *CGFR* didapatkan persamaan sebagai berikut.

$$d_{v_{ij}(t)} = -g_{v_{ij}(t)}, \text{ untuk } t=0,$$

$$d_{v_{ij}(t)} = -g_{v_{ij}(t)} + \beta_{v_{ij}(t)} d_{v_{ij}(t-1)}, \text{ untuk } t=1, \dots, n_x$$

dengan $\beta_{v_{ij}(t+1)} = \frac{g_{v_{ij}(t)}^T g_{v_{ij}(t)}}{g_{v_{ij}(t-1)}^T g_{v_{ij}(t-1)}}$.

Perubahan bobot v_{ij} pada iterasi ke- $(t+1)$ adalah sebagai berikut

$$v_{ij(t+1)} = v_{ij(t)} + \eta d_{v_{ij}(t)}.$$

b) Perubahan bias $v_{0j}, j=1, \dots, p$;

Perubahan bias v_{0j} diperoleh dengan menurunkan persamaan (2.3) terhadap v_{0j} .

$$\begin{aligned} \frac{\partial E_s}{\partial v_{0j}} &= \sum_{k=1}^m \left(\frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_k} \frac{\partial y_k}{\partial nety_k} \frac{\partial nety_k}{\partial z_j} \frac{\partial z_j}{\partial netz_j} \frac{\partial netz_j}{\partial v_{0j}} \right) \\ &= \sum_{k=1}^m \left(\frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial nety_k} (f(nety_k)) \right. \\ &\quad \left. \frac{\partial}{\partial z_j} \left(w_{0k} + \sum_{j=1}^p w_{jk} z_j \right) \frac{\partial}{\partial netz_j} (f(netz_j)) \right. \\ &\quad \left. \frac{\partial}{\partial v_{0j}} \left(v_{0j} + \sum_{i=1}^n v_{0j} x_i \right) \right). \end{aligned}$$

$$\frac{\partial E_s}{\partial v_{0j}} = - \sum_{k=1}^m \left(((T_k - y_k) f'(nety_k) w_{jk}) f'(netz_j) \right). \quad (3.6)$$

Substitusi persamaan (3.5) pada persamaan (3.6)

$$g_{v_{0j}} = \frac{\partial E_s}{\partial v_{0j}} = -\gamma_j.$$

Berdasarkan algoritma *CGFR* diperoleh persamaan berikut.

$$d_{v_{0j}(t)} = -g_{v_{0j}(t)}, \text{ untuk } t=0,$$

$$d_{v_{0j}(t)} = -g_{v_{0j}(t)} + \beta_{v_{0j}(t)} d_{v_{0j}(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{v_{0j}(t+1)} = \frac{g_{v_{0j}(t)}^T g_{v_{0j}(t)}}{g_{v_{0j}(t-1)}^T g_{v_{0j}(t-1)}}.$$

Perubahan bobot v_{0j} pada iterasi ke-($t+1$) adalah sebagai berikut

$$v_{0j(t+1)} = v_{0j(t)} + \eta d_{v_{0j}(t)}.$$

3.2 Backpropagation dengan metode CGFRAG

Perubahan bobot dan bias masing-masing lapisan pada model pembelajaran *Backpropagation* dengan menggunakan metode *CGFRAG* adalah sebagai berikut.

1. Perubahan bobot antara lapisan *output* dan lapisan *hidden*

a) Perubahan bobot w_{jl} , $j=1, \dots, p$; $l=1, \dots, m$

Perubahan bobot w_{jl} diperoleh dengan menurunkan persamaan (2.3) terhadap w_{jl} .

$$\begin{aligned} \frac{\partial E_s}{\partial w_{jl}} &= \frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_l} \frac{\partial y_l}{\partial \text{net}_{yl}} \frac{\partial \text{net}_{yl}}{\partial w_{jl}} \\ \frac{\partial E_s}{\partial w_{jl}} &= \frac{\partial}{\partial y_l} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial \text{net}_l} (f(c2_l \text{net}_l)) \\ &\quad \frac{\partial}{\partial w_{jl}} \left(w_{0l} + \sum_{j=1}^p w_{jl} z_j \right). \\ \frac{\partial E_s}{\partial w_{jl}} &= -(T_l - y_l) f'(c2_l \text{net}_{yl}) c2_l z_j, \quad j \neq 0. \end{aligned} \quad (3.7)$$

$$\text{Misalkan } \delta_l = (T_l - y_l) f'(c2_l \text{net}_{yl}) c2_l, \quad (3.8)$$

kemudian substitusikan persamaan (3.8) ke persamaan (3.7) sehingga diperoleh

$$g_{w_{jl}} \frac{\partial E_s}{\partial w_{jl}} = -\delta_l z_j.$$

Berdasarkan algoritma *CGFRAG* diperoleh persamaan persamaan berikut.

$$d_{w_{jl}(t)} = -g_{w_{jl}(t)}, \text{ untuk } t=0,$$

$$d_{w_{jl}(t)} = -g_{w_{jl}(t)} + \beta_{w_{jl}(t)} d_{w_{jl}(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{w_{jl}(t+1)} = \frac{g_{w_{jl}(t)}^T g_{w_{jl}(t)}}{g_{w_{jl}(t-1)}^T g_{w_{jl}(t-1)}}.$$

Perubahan bobot w_{jl} pada iterasi ke- $(t+1)$ adalah sebagai berikut.

$$w_{jl(t+1)} = w_{jl(t)} + \eta d_{w_{jl}(t)}.$$

b) Perubahan bias w_{0l} , $l=1, \dots, m$

Perubahan bias w_{0l} diperoleh dengan menurunkan persamaan (2.3) terhadap w_{0l} .

$$\begin{aligned} \frac{\partial E_s}{\partial w_{0l}} &= \frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_l} \frac{\partial y_l}{\partial \text{net}_l} \frac{\partial \text{net}_l}{\partial w_{0l}} \\ &= \frac{\partial}{\partial y_l} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial \text{net}_l} (f(c2_l \text{net}_l)) \\ &\quad \frac{\partial}{\partial w_{0l}} \left(w_{0l} + \sum_{j=1}^p w_{jl} z_j \right). \end{aligned}$$

$$\frac{\partial E_s}{\partial w_{0l}} = -(T_l - y_l) f'(c2_l \text{net}_l) c2_l, \quad (3.9)$$

Substitusi persamaan (3.8) pada persamaan (3.9) menghasilkan

$$g_{w_{0l}} = \frac{\partial E_s}{\partial w_{0l}} = -\delta_l.$$

Berdasarkan algoritma *CGFRAG* diperoleh persamaan berikut.

$$d_{w_{0l}(t)} = -g_{w_{0l}(t)}, \text{ untuk } t=0,$$

$$d_{w_{0l}(t)} = -g_{w_{0l}(t)} + \beta_{w_{0l}(t)} d_{w_{0l}(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{w_{0l}(t+1)} = \frac{g_{w_{0l}(t)}^T g_{w_{0l}(t)}}{g_{w_{0l}(t-1)}^T g_{w_{0l}(t-1)}}$$

Perubahan bias w_{0l} pada iterasi ke- $(t+1)$ adalah sebagai berikut.

$$w_{0l(t+1)} = w_{0l(t)} + \eta d_{w_{0l}(t)}$$

2. Perubahan bobot antara lapisan *hidden* dan lapisan *input*

a) Perubahan bobot v_{ij} , $i=1, \dots, n$; $j=1, \dots, p$;

Perubahan bobot v_{ij} diperoleh dengan menurunkan persamaan (2.3) terhadap v_{ij} .

$$\begin{aligned} \frac{\partial E_s}{\partial v_{ij}} &= \sum_{k=1}^m \left(\frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_k} \frac{\partial y_k}{\partial nety_k} \frac{\partial nety_k}{\partial z_j} \frac{\partial z_j}{\partial netz_j} \frac{\partial netz_j}{\partial v_{ij}} \right) \\ &= \sum_{k=1}^m \left(\frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial nety_k} (f(c2_k nety_k)) \right. \\ &\quad \left. \frac{\partial}{\partial z_j} \left(w_{0k} + \sum_{j=1}^p w_{jk} z_j \right) \frac{\partial}{\partial netz_j} (f(c1_j netz_j)) \right. \\ &\quad \left. \frac{\partial}{\partial v_{ij}} \left(v_{0j} + \sum_{i=1}^n v_{ij} x_i \right) \right). \end{aligned}$$

$$\frac{\partial E_s}{\partial v_{ij}} = - \sum_{k=1}^m \left(((T_k - y_k) f'(c2_k nety_k) w_{jk} f'(c1_j netz_j) c1_j) x_i \right), i \neq 0 \quad (3.10)$$

Misalkan

$$\gamma_j = \sum_{k=1}^m (f'(c2_k nety_k) w_{jk} f'(c1_j netz_j) c1_j) \quad (3.11)$$

Substitusikan persamaan (3.11) pada persamaan (3.10), sehingga diperoleh

$$g_{v_{ij}} = \frac{\partial E_s}{\partial v_{ij}} = -\gamma_j x_i.$$

Berdasarkan algoritma *CGFRAG* diperoleh persamaan sebagai berikut.

$$d_{v_{ij}(t)} = -g_{v_{ij}(t)}, \text{ untuk } t=0,$$

$$d_{v_{ij}(t)} = -g_{v_{ij}(t)} + \beta_{v_{ij}(t)} d_{v_{ij}(t-1)}, \text{ untuk } t=1, \dots, n_2$$

$$\text{dengan } \beta_{v_{ij}(t+1)} = \frac{g_{v_{ij}(t)}^T g_{v_{ij}(t)}}{g_{v_{ij}(t-1)}^T g_{v_{ij}(t-1)}}.$$

Perubahan bobot v_{ij} pada iterasi ke- $(t+1)$ adalah sebagai berikut

$$v_{ij(t+1)} = v_{ij(t)} + \eta d_{v_{ij}(t)}.$$

b) Perubahan bias v_{0j} , $j=1, \dots, p$;

Perubahan bias v_{0j} diperoleh dengan menurunkan persamaan (2.3) terhadap v_{0j} .

$$\begin{aligned} \frac{\partial E_s}{\partial v_{0j}} &= \sum_{k=1}^m \left(\frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_k} \frac{\partial y_k}{\partial \text{net}y_k} \frac{\partial \text{net}y_k}{\partial z_j} \frac{\partial z_j}{\partial \text{net}z_j} \frac{\partial \text{net}z_j}{\partial v_{0j}} \right) \\ &= \sum_{k=1}^m \left(\frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial \text{net}y_k} (f(c2_k \text{net}y_k)) \right. \\ &\quad \left. \frac{\partial}{\partial z_j} \left(w_{0k} + \sum_{j=1}^p w_{jk} z_j \right) \frac{\partial}{\partial \text{net}z_j} (f(c1_j \text{net}z_j)) \right. \\ &\quad \left. \frac{\partial}{\partial v_{0j}} \left(v_{0j} + \sum_{i=1}^n v_{0j} x_i \right) \right) \end{aligned}$$

$$\frac{\partial E_s}{\partial v_{0j}} = - \sum_{k=1}^m \left(((T_k - y_k) f'(c2_k \text{net}y_k) w_{jk}) f'(c1_j \text{net}z_j) c1_j \right). \quad (3.12)$$

Substitusikan persamaan (3.11) pada persamaan (3.12) menghasilkan

$$g_{v_{0j}} = \frac{\partial E_s}{\partial v_{0j}} = -\gamma_j.$$

Berdasarkan algoritma *CGFRAG* diperoleh persamaan sebagai berikut.

$$d_{v_{0j}(t)} = -g_{v_{0j}(t)}, \text{ untuk } t=0,$$

$$d_{v_{0j}(t)} = -g_{v_{0j}(t)} + \beta_{v_{0j}(t)} d_{v_{0j}(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{v_{0j}(t+1)} = \frac{g_{v_{0j}(t)}^T g_{v_{0j}(t)}}{g_{v_{0j}(t-1)}^T g_{v_{0j}(t-1)}}.$$

Perubahan bobot v_{0j} pada iterasi ke-($t+1$) adalah sebagai berikut

$$v_{0j(t+1)} = v_{0j(t)} + \eta d_{v_{0j}(t)}.$$

3. Perubahan *Adaptive Gain* antara lapisan *output* dan lapisan *hidden*

a) Perubahan *Adaptive Gain* c_{2_l} ; $l=1, \dots, m$

Perubahan *Adaptive Gain* c_{2_l} diperoleh dengan menurunkan persamaan (2.3) terhadap c_{2_l} .

$$\begin{aligned} \frac{\partial E_s}{\partial c_{2_l}} &= \frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_l} \frac{\partial y_l}{\partial c_{2_l}} \\ &= \frac{\partial}{\partial y_l} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial c_{2_l}} (f(c_{2_l} net_l)) \\ \frac{\partial E_s}{\partial c_{2_l}} &= -(T_l - y_l) f'(c_{2_l} net_l) net_l. \end{aligned} \quad (3.13)$$

Persamaan (3.13) dapat ditulis sebagai berikut

$$\frac{\partial E_s}{\partial c_{2_l}} = -(T_l - y_l) f'(c_{2_l} net_l) net_l \frac{c_{2_l}}{c_{2_l}}, \quad (3.14)$$

kemudian substitusikan persamaan (3.8) pada persamaan (3.14), sehingga diperoleh

$$g c_{2_l} = \frac{\partial E_s}{\partial c_{2_l}} = -\delta_l \frac{net_l}{c_{2_l}}.$$

Berdasarkan algoritma *CGFRAG* diperoleh persamaan sebagai berikut.

$$\begin{aligned}
d_{c2_l(t)} &= -g_{c2_l(t)}, \text{ untuk } t=0, \\
d_{c2_l(t)} &= -g_{c2_l(t)} + \beta_{c2_l(t)} d_{c2_l(t-1)}, \text{ untuk } t=1, \dots, n, \\
\text{dengan } \beta_{c2_l(t+1)} &= \frac{g_{c2_l(t)}^T g_{c2_l(t)}}{g_{c2_l(t-1)}^T g_{c2_l(t-1)}}.
\end{aligned}$$

Perubahan bobot c_{2_l} pada iterasi ke-($t+1$) adalah sebagai berikut.

$$c_{2_l(t+1)} = c_{2_l(t)} + \eta d_{c2_l(t)}.$$

c) Perubahan *Adaptive Gain* c_{1_j} ; $j=1, \dots, p$;

Perubahan *Adaptive Gain* c_{1_j} diperoleh dengan menurunkan persamaan (2.3) terhadap c_{1_j} .

$$\begin{aligned}
\frac{\partial E_s}{\partial c_{1_j}} &= \sum_{k=1}^m \left(\frac{\partial \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right)}{\partial y_k} \frac{\partial y_k}{\partial \text{net}y_k} \frac{\partial \text{net}y_k}{\partial z_j} \frac{\partial z_j}{\partial c_{1_j}} \right) \\
&= \sum_{k=1}^m \left(\frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (T_k - y_k)^2 \right) \frac{\partial}{\partial \text{net}y_k} (f(c_{2_k} \text{net}y_k)) \right. \\
&\quad \left. \frac{\partial}{\partial z_j} (w_{0k} + \sum_{j=1}^p w_{jk} z_j) \frac{\partial}{\partial \text{net}z_j} (f(c_{1_j} \text{net}z_j)) \right). \\
\frac{\partial E_s}{\partial c_{1_j}} &= - \sum_{k=1}^m \left((T_k - y_k) f'(c_{2_k} \text{net}y_k) c_{2_k} w_{jk} \right. \\
&\quad \left. f'(c_{1_j} \text{net}z_j) \text{net}z_j \right). \tag{3.15}
\end{aligned}$$

Persamaan (3.15) dapat ditulis sebagai berikut.

$$\begin{aligned}
\frac{\partial E_s}{\partial c_{1_j}} &= - \sum_{k=1}^m \left((T_k - y_k) f'(c_{2_k} \text{net}y_k) c_{2_k} w_{jk} \right. \\
&\quad \left. f'(c_{1_j} \text{net}z_j) \text{net}z_j \frac{c_{1_j}}{c_{1_j}} \right). \tag{3.16}
\end{aligned}$$

kemudian substitusikan persamaan (3.11) pada persamaan (3.16), sehingga diperoleh

$$g_{c_{1_j}} = \frac{\partial E_s}{\partial c_{1_j}} = -\gamma_j \frac{\text{net}z_j}{c_{1_j}}.$$

Berdasarkan algoritma *CGFRAG* diperoleh persamaan berikut.

$$d_{c1_j(t)} = -g_{c1_j(t)}, \text{ untuk } t=0,$$

$$d_{c1_j(t)} = -g_{c1_j(t)} + \beta_{c1_j(t)} d_{c1_j(t-1)}, \text{ untuk } t=1, \dots, n,$$

$$\text{dengan } \beta_{c1_j(t+1)} = \frac{g_{c1_j(t)} g_{c1_j(t)}}{g_{c1_j(t-1)} g_{c1_j(t-1)}}.$$

Perubahan bobot $c1_j$ pada iterasi ke- $(t+1)$ adalah sebagai berikut.

$$c1_{j(t+1)} = c1_{j(t)} + \eta d_{c1_j(t)}.$$

3.3 Membangun JST

1. *Arsitektur Jaringan*

Arsitektur JST dalam skripsi ini adalah jaringan Lapis Ganda. Arsitektur jaringan dibangun berdasarkan variabel-variabel yang mempengaruhi data penderita diabetes pada suku Pima Indian (Lampiran 3). Variabel-variabel tersebut antara lain: umur, waktu tenggang melahirkan, indeks berat tubuh (berat kg/ tinggi m²), toleransi glukosa darah dalam plasma darah (mg/dl), tekanan darah *diastolic* (mmHg), ketebalan lipatan kulit lengan bawah (mm), 2-jam serum insulin (mu U/ml) dan fungsi silsilah diabetes. *Neuron* lapisan *hidden* yang digunakan antara lain: 3,5, 8 dan 12 *neuron*. Variabel yang digunakan sebagai *output* jaringan adalah *class variable*.

Model arsitektur 8-5-1 berarti arsitektur jaringan menggunakan 8 *neuron* lapisan *input*, 5 *neuron* lapisan *hidden* dan 1 *neuron* lapisan *output*.

2. *Fungsi Aktivasi*

Fungsi aktivasi diperoleh dengan mencari kombinasi terbaik dari fungsi Identitas, sigmoid Biner dan sigmoid Bipolar sehingga diperoleh nilai *output* yang minimum. Fungsi aktivasi yang digunakan adalah fungsi Sigmoid Biner pada Lapisan *hidden*, sedangkan pada lapisan *output* menggunakan fungsi Identitas.

3. *Laju Pembelajaran*

Laju pembelajaran merupakan ukuran langkah yang berfungsi untuk menentukan nilai bobot dan bias pada iterasi ke- $(t+1)$. Nilai laju pembelajaran bersifat statis, dipilih antara lain: 0,0001; 0,00008; 0,00005 dan 0,000035.

4. *Kriteria Pemberhentian*

Kriteria pemberhentian yang diberikan pada model *Backpropagation* ini adalah MSE kurang dari 0,000001 dan iterasi maksimal sebesar 1000.

5. *Piranti komputer dan software*

Proses pembelajaran dan simulasi menggunakan piranti komputer pentium IV 1,8 GHz, 1 GB RAM, Windows XP, dan *software* Matlab versi 5.3.1.29215a (R11.1).

3.4 Normalisasi Data

Data yang digunakan adalah data penderita diabetes pada suku Pima Indian sebanyak 768 (Lampiran 3), dengan 500 data sebagai data pelatihan dan 268 data digunakan dalam proses simulasi. Normalisasi bertujuan untuk menghindari perhitungan dengan angka-angka besar yang menyebabkan proses perhitungan membutuhkan waktu lebih lama. Data-data tersebut dinormalisasikan dengan normalisasi Desimal *Scalling*.

3.5 Hasil Pembelajaran

Hasil pembelajaran *Backpropagation* dengan metode *CGFR* dan *CGFRAG* menggunakan beberapa model dan nilai laju pembelajaran yang diberikan pada Tabel 3.1.

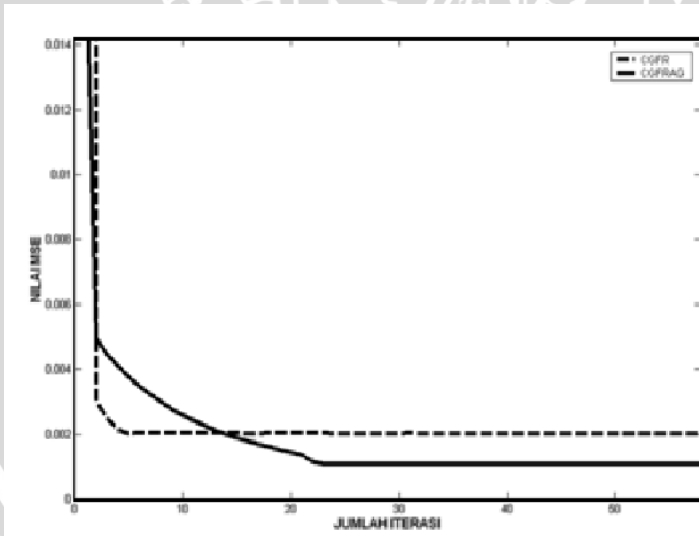
Tabel 3.1. Perbandingan hasil pembelajaran *Backpropagation* dengan metode *CGFR* dan *CGFRAG*.

	Laju Pembelajaran (η)	MSE (10^{-3})		Waktu (Detik 10^3)	
		<i>CGFR</i>	<i>CGFRAG</i>	<i>CGFR</i>	<i>CGFRAG</i>
8-3-1	0,000035	1,7352	1,1252	1,4598	0,6695620
	0,00005	1,7908	1,095	1,8658	0,690870
	0,00008	1,782	1,1322	0,6389370	0,6809690
	0,0001	1,895	1,1927	0,6060310	0,6908280
8-5-1	0,000035	2,4	1,9156	1,1509	1,0724
	0,00005	1,5418	1,2974	1,4783	1,0607
	0,00008	1,78	1,72	1,2790	1,4187
	0,0001	2,036	1,965	1,0051	1,0906
8-8-1	0,000035	2,561	1,535	2,3064	1,6734
	0,00005	2,22	1,9	1,5962	1,9873
	0,00008	2,2	1,565	1,5184	1,6936
	0,0001	2,08	1,1868	1,4847	1,6763
8-12-1	0,000035	2,2509	2,40	2,3243	1,6891
	0,00005	1,514	1,6897	2,4618	2,4953
	0,00008	1,4574	2,5724	2,2930	1,6608
	0,0001	1,4527	-	2,2126	-

Tabel 3.1 menunjukkan MSE dan waktu proses pembelajaran masing-masing metode. Metode *GCFR* menghasilkan MSE terkecil $1,4527e-003$ pada model arsitektur 8-12-1 dengan laju pembelajaran 0,0001, sedangkan metode *GCFRAG* menghasilkan MSE terkecil sebesar $1,095e-003$ pada model arsitektur 8-3-1 dengan laju pembelajaran 0,00005. Arsitektur jaringan dan laju pembelajaran yang berbeda mempengaruhi MSE yang dihasilkan. Seluruh pembelajaran menggunakan metode *GCFRAG* menghasilkan nilai MSE yang lebih kecil dibandingkan metode *GCFR*, kecuali arsitektur 8-12-1. MSE metode *GCFRAG* pada arsitektur tersebut relatif lebih besar, bahkan gagal melakukan pembelajaran pada laju pembelajaran 0,0001.

Hasil pembelajaran *Backpropagation* menggunakan metode *CGFR* dan *CGFRAG* menghasilkan nilai MSE yang berbeda. Gambar 3.1 sampai dengan Gambar 3.4 menunjukkan perbandingan dan perilaku grafik masing-masing arsitektur kedua metode tersebut. Perbandingan menggunakan arsitektur JST dengan nilai MSE terkecil sesuai Tabel 3.1.

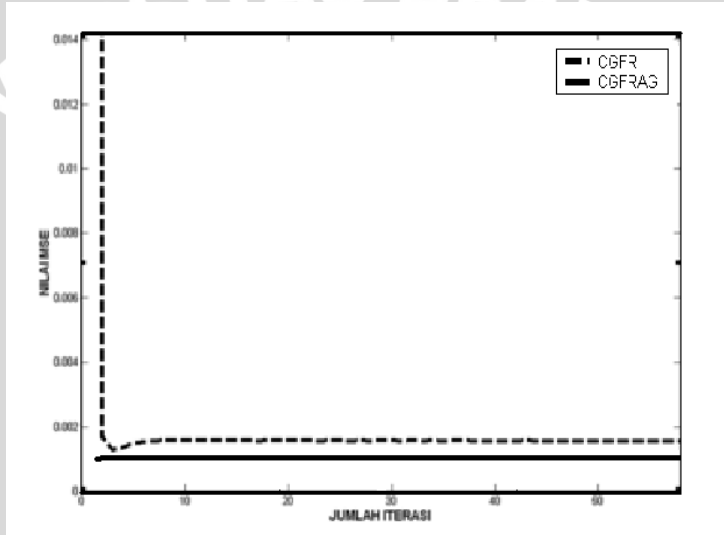
1. Model arsitektur 8-3-1



Gambar 3.1 Perbandingan hasil pembelajaran model 8-3-1 dengan $\eta=0,00005$

Metode *CGFR* pada arsitektur 8-3-1 menghasilkan MSE sekitar $0,002 \times 10^{-3}$ pada 10 iterasi pertama, kemudian nilainya tidak berubah sampai iterasi ke-1000 walaupun grafiknya tidak stabil. Metode *CGFRAG* menghasilkan MSE 5×10^{-3} pada iterasi pertama, kemudian menghasilkan MSE sekitar 1×10^{-3} pada iterasi selanjutnya. Hal ini menunjukkan metode *CGFRAG* dapat menghasilkan MSE yang kecil pada 20 iterasi pertama.

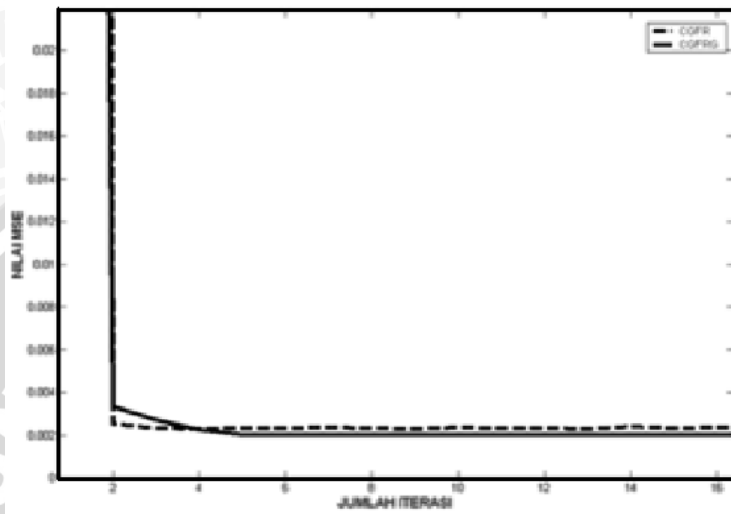
2. Model arsitektur 8-5-1



Gambar 3.2 Hasil pembelajaran model 8-5-1 dengan $\eta=0,00005$

Metode *CGFR* menghasilkan grafik MSE yang tidak stabil. Gambar 3.2 menunjukkan metode tersebut menghasilkan MSE sebesar $0,02 \times 10^{-3}$ pada iterasi pertama, kemudian turun menjadi $1,3 \times 10^{-3}$ dan naik menjadi $1,52 \times 10^{-3}$ sampai dengan iterasi terakhir. Metode *CGFRAG* menghasilkan nilai MSE relatif stabil sekitar $1,2 \times 10^{-3}$ pada awal iterasi sampai dengan iterasi ke-1000.

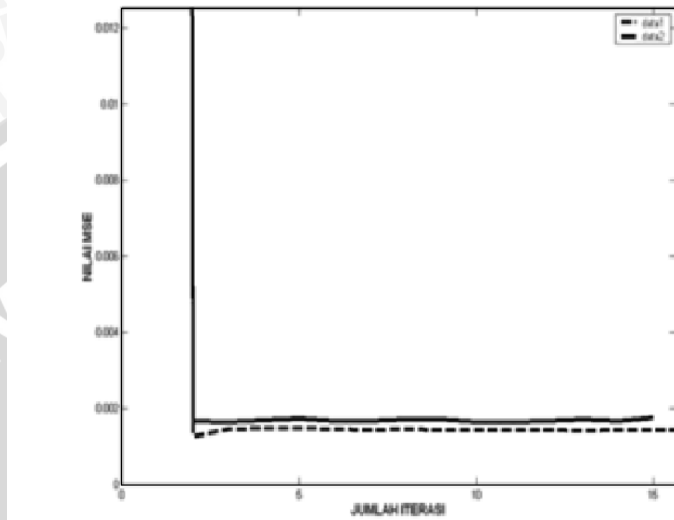
3. Model arsitektur 8-8-1



Gambar 3.3 Hasil pembelajaran model 8-8-1 dengan $\eta=0,0001$

Gambar 3.3 menunjukkan grafik yang dihasilkan metode *CGFRAG* dan *CGFR* tidak stabil, tetapi metode *CGFRAG* menghasilkan MSE yang lebih kecil pada keseluruhan iterasi.

4. Model arsitektur 8-12-1



Gambar 3.4 Hasil pembelajaran model 8-12-1 dengan $\eta=0,0001$

Gambar 3.4 merupakan perbandingan metode *CGFR* dan *CGFRAG* menggunakan laju pembelajaran 0,0001. Metode *CGFR* menghasilkan MSE yang lebih kecil daripada *CGFRAG* pada seluruh pembelajaran arsitektur 8-12-1. Gambar 3.4 menunjukkan metode *CGFRAG* gagal melakukan pembelajaran dan berhenti pada iterasi ke-15. Hal ini dikarenakan laju pembelajaran dan *neuron hidden layer* yang digunakan terlalu besar, sehingga menyebabkan nilai *output* akan menjadi besar dan divergen. Metode *CGFR* menghasilkan MSE 1,4527 tetapi grafiknya tidak stabil.

3.6 Analisa Hasil Pembelajaran

Berdasarkan hasil pembelajaran pada Tabel 3.1 didapatkan perbandingan dari kedua metode sebagai berikut.

1. MSE

Metode *CGFRAG* menghasilkan MSE yang lebih kecil dibandingkan metode *CGFR*. MSE terkecil metode *CGFRAG* sebesar $1,095 \times 10^{-3}$, sedangkan metode *CGFR* bernilai $1,4527 \times 10^{-3}$. Pembelajaran metode *CGFRAG* menggunakan arsitektur 8-12-1 dan laju pembelajaran 0,0001 gagal menghasilkan nilai MSE karena arsitektur dan laju pembelajaran yang digunakan terlalu besar.

2. Laju Pembelajaran

Pemilihan laju pembelajaran pada arsitektur jaringan sangat mempengaruhi laju konvergensi masing-masing metode. Nilai laju pembelajaran yang terlalu besar menyebabkan proses pembelajaran menjadi tidak stabil, bahkan pada metode *CGFRAG* menyebabkan jaringan gagal melakukan pembelajaran. Semakin besar laju pembelajaran yang digunakan metode *CGFRAG* maka semakin besar pula waktu yang dibutuhkan. Kecuali pada laju pembelajaran 0,0001 waktu yang dibutuhkan akan kecil. MSE terkecil metode *CGFR* dan metode *CGFRAG* dihasilkan pada arsitektur 8-3-1 dengan laju pembelajaran 0,00005.

3. Arsitektur Jaringan

Arsitektur jaringan dipengaruhi oleh banyaknya *neuron* lapisan *hidden* yang digunakan. MSE metode *CGFRAG* lebih kecil daripada metode *CGFR* kecuali pada arsitektur 8-12-1. Hal ini menunjukkan penambahan *Adaptive Gain* dapat menghasilkan MSE yang kecil pada arsitektur sederhana, tetapi tidak efektif jika digunakan pada arsitektur jaringan yang besar.

4. Waktu Komputasi

Metode *CGFR* memiliki rata-rata laju konvergensi yang lebih cepat dibandingkan metode *CGFRAG*, karena perhitungan algoritma *CGFR* lebih sederhana dibandingkan algoritma *CGFRAG*. Total waktu pembelajaran metode *CGFR* $25,6813 \times 103$ detik dan metode *CGFRAG* selama $20,2917 \times 103$ detik dengan 1 arsitektur gagal melakukan pembelajaran.

3.7 Pemilihan Model Arsitektur Terbaik

Pemilihan model arsitektur terbaik bertujuan untuk menentukan model arsitektur yang digunakan dalam proses simulasi. Kriteria pemilihan arsitektur terbaik adalah sebagai berikut.

3.7.1 Nilai MSE

Kriteria pemilihan arsitektur dengan nilai MSE terkecil berpengaruh terhadap proses simulasi, sehingga diperoleh arsitektur masing-masing metode sebagai berikut.

1) Metode *CGFR*.

MSE terkecil dari proses pembelajaran metode *CGFR* yaitu $1,4527 \times 10^{-3}$ menggunakan arsitektur 8-12-1 dengan laju pembelajaran 0,0001.

2) Metode *CGFRAG*

MSE terkecil yang dihasilkan metode *CGFRAG* pada saat proses pembelajaran sebesar $1,095 \times 10^{-3}$ terdapat pada arsitektur 8-3-1 dengan laju pembelajaran 0,00005.

3.7.2 Waktu Komputasi

Pemilihan waktu komputasi terbaik bertujuan untuk mendapatkan metode dengan arsitektur tercepat untuk mencapai kekonvergenan. Analisa hasil pembelajaran adalah sebagai berikut.

1) Metode *CGFR*

Hasil pembelajaran Tabel 3.1 menunjukkan semakin besar model arsitektur jaringan maka semakin lama waktu yang dibutuhkan. Waktu komputasi tercepat sebesar $0,6060310 \times 10^3$ detik terdapat pada model 8-3-1 dengan laju pembelajaran 0,0001, sedangkan waktu komputasi terlama $2,4618 \times 10^3$ detik terdapat pada model 8-12-1 dengan laju pembelajaran 0,00005.

2) Metode *CGFRAG*

Metode *CGFRAG* membutuhkan waktu yang berbeda-beda tiap arsitektur jaringan yang digunakan. Waktu komputasi tercepat sebesar 669,5620 detik terdapat pada model 8-3-1 dengan laju pembelajaran 0,000035, sedangkan waktu komputasi terlama sebesar $2,4953 \times 10^3$ detik pada model 8-12-1 dengan laju pembelajaran 0,00005.

Berdasarkan MSE dan total waktu yang didapatkan, arsitektur terbaik metode *CGFR* adalah model 8-12-1 dengan laju pembelajaran 0,0001, sedangkan arsitektur terbaik metode *CGFRAG* adalah model 8-3-1 dengan laju pembelajaran 0,00005. Arsitektur tersebut digunakan dalam proses simulasi.

3.8 Proses Simulasi

Tahap selanjutnya yaitu menguji kemampuan JST untuk mengklasifikasikan data penderita diabetes pada suku Pima Indian sebanyak 268 data. Data tersebut dinormalisasikan terlebih dahulu, kemudian dilakukan proses simulasi. Proses simulasi menggunakan bobot awal yang diperoleh dari proses pembelajaran (Lampiran 1 dan Lampiran 2).

Hasil simulasi (Lampiran 3) menunjukkan selisih antara nilai target dan *output* proses simulasi yang disebut *error*, dan perbandingan nilai *error* antara metode *CGFR* dan *CGFRAG*.

Metode *CGFR* menghasilkan *error* terbesar yaitu 0,1155 pada data ke-207, dan total MSE sebesar 9,1952. Metode *CGFRAG* menghasilkan *error* terbesar yaitu 0,0996 pada data ke-120. Total MSE yang dihasilkan sebesar 8,6563.

Kedua metode tersebut dapat mengklasifikasikan data penderita diabetes pada suku Pima Indian dan memiliki tingkat akurasi yang hampir sama, tetapi metode *CGFRAG* memiliki tingkat ketelitian yang lebih baik. Hal ini dapat dilihat dari *error* yang dihasilkan metode *CGFRAG* yang lebih kecil daripada metode *CGFR* pada keseluruhan hasil simulasi.

UNIVERSITAS BRAWIJAYA



BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil dan pembahasan dapat diambil kesimpulan berikut.

1. Metode *CGFR* dan *CGFRAG* dapat diterapkan dalam model *Backpropagation* untuk mencari perubahan bobot dan bias. Penambahan *Adaptive Gain* pada fungsi aktivasi menghasilkan nilai MSE yang lebih kecil, namun tidak efektif jika digunakan pada arsitektur dan laju pembelajaran yang terlalu besar.
2. Pengklasifikasian penderita diabetes pada suku Pima Indian menggunakan kedua metode tersebut memberikan hasil yang berbeda. Proses pembelajaran menggunakan metode *CGFRAG* menghasilkan MSE lebih kecil daripada metode *CGFR*, namun membutuhkan waktu lebih lama. Kedua metode tersebut dapat digunakan untuk mengklasifikasikan apakah seseorang terkena diabetes atau tidak, tetapi metode *CGFRAG* menghasilkan tingkat kesalahan yang lebih kecil.

4.2 Saran

β *Fletcher-Reeves* menghasilkan nilai yang mendekati 0 pada iterasi ke- $(k+1)$. Untuk mendapatkan hasil yang lebih optimal dari $d_{(k+1)}$ dapat digunakan β *Polak-Ribere*, β *Al Assady-Al Bayati*, β *Hestene-Steifel* atau yang lain .

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Al Bayati, A.Y., Najmaddin A.S., dan Gulnar W.S. 2009. *A Modified Conjugate Gradient Formula for Backpropagation Neural Network Algorithm*. Journal of Computer Science 5, Science Publications.
- Al Shalabi, L., Shaaban,Z. dan Kasasbeh, B., 2006. *Data Mining: A Preprocessing Engine*. Journal of Computer Science 2, Science Publications.
- Anton, H. 1987. *Aljabar Linear Elementer*. Erlangga. Jakarta.
- Ayub, B.M dan R.H. McCuen. 1996. *Numerical Methods for Engineers*. Prentice-Hall,inc. New Jersey.
- Boyd, S. dan L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press. Neuroned Kingdom.
- Bronshtein, I.N., K.A. Semendyayev, G. Musiol, dan H. Muehlig. 2007. *Handbook of Mathematics*. Springer. Berlin.
- Chapra dan C. Steven. 2002. *Numerical methods for Engineers with Software and Programming Applications*, 4th ed. McGraw-Hill.Inc. New York.
- Freund, R. 2004. *The Steepest Descent Algorithm for Unconstrained Optimization*. Massachusetts Institute of Technology.
- Kusumadewi, S. dan Hartati S. 2006. *Integrasi Sistem Fuzzy dan Jaringan Saraf*. Graha Ilmu. Yogyakarta.
- Marthin, E. *Diabetes Mellitus (Penyakit Kadar Gula Tinggi)*. <http://WordPress.com/DiabetesMellitus>, tanggal akses: 11 Maret 2009.
- Mathews, J.H dan Kurtis D.F. 2004. *Numerical Methods using Matlab*, 4th ed. Pearson Education International. USA.
- Muis, S. 2006. *Teknik Jaringan Saraf Tiruan*. Graha Ilmu. Yogyakarta.
- Nawi, N.M., Rajesh, S.R. dan Meghana R.S. *An Improved Conjugate Gradient Based Learning Algorithm for Backpropagation Neural Networks*. International Journal of Computational Intelligence.
- Nocedal, J. dan Stephen J.W. 1999. *Numerical Optimization*. Springer-Verlag New York, Inc. USA.
- Patterson, D.W. 1996. *Artificial Neural Networks Theory and Applications*. Prentice Hall Simon and Schusler Pte Ltd. Singapore.

Sigillito, V. 1990. *National Institute of Diabetes and Digestive and Kidney Diseases*. <ftp://ftp.ics.uci.edu/pub/machine-learningdatabases/diabetes>, tanggal akses: 11 Maret 2009.

UNIVERSITAS BRAWIJAYA



LAMPIRAN

Lampiran 1:

Nilai bobot dan bias hasil pembelajaran metode *CGFR* dengan model 8-12-1.

Bobot dan bias antara *neuron input layer* dan *neuron hidden layer*(V_{ij})

V_{ij}		j											
		1	2	3	4	5	6	7	8	9	10	11	12
i	0	2,0490	0,8560	0,0522	0,7718	1,0078	0,9749	0,7744	0,4174	0,4650	0,2648	0,5465	0,2715
	1	1,0453	0,7257	0,4134	0,7446	0,2695	0,4391	0,9326	0,6823	0,2110	0,8415	0,6243	0,1315
	2	0,4595	0,6049	0,6353	0,3714	0,5807	0,4495	0,0421	0,0244	0,3083	0,0204	0,3702	0,6766
	3	0,2119	0,0341	0,6148	0,6088	0,0182	0,0153	0,1891	0,5855	0,0554	0,3709	0,6251	0,7144
	4	0,9996	0,0806	0,4594	0,4419	0,3576	0,1509	0,6730	0,6956	0,7219	0,4836	0,5405	0,1133
	5	0,5646	0,7142	0,8927	0,2721	0,2548	0,8644	0,2310	0,8035	0,9062	0,2311	0,2356	0,0473
	6	0,6804	0,6353	0,2046	0,8464	0,1875	0,1661	0,9899	0,4330	0,3293	0,3331	0,3318	0,3771
	7	0,6931	0,1188	0,0403	0,4590	0,8720	0,9335	0,2637	0,1592	0,8711	0,2407	0,6403	0,9643
	8	1,3072	0,8636	0,0217	0,1380	0,8311	0,4246	0,8850	0,7273	0,6755	0,3626	0,1323	0,1386



Bobot dan bias antara *neuron input layer* dan *neuron hidden layer*(W_{jk})

	W_{jk}	k
		1
j	0	-0,2981
	1	0,0149
	2	0,3995
	3	0,0984
	4	0,3598
	5	0,0064
	6	0,0177
	7	-0,0560
	8	-0,0960
	9	0,3939
	10	-0,4012
	11	-0,1529
	12	-0,3056

Lampiran 2:

Nilai bobot dan bias hasil pembelajaran metode *CGFRAG* dengan model 8-3-1.

Nilai Bobot dan bias antara *neuron input layer* dan *neuron hidden layer* (V_{ij})

V_{ij}		j		
		1	2	3
i	0	0,8379	0,4119	0,7446
	1	0,2771	0,4399	0,9334
	2	0,7142	0,2125	0,8393
	3	0,6352	0,1338	0,2071
	4	0,6283	0,6299	0,3705
	5	0,6042	0,4514	0,0439
	6	0,0906	0,3126	0,0130
	7	0,3963	0,6831	0,0929
8	0,0779	0,6124	0,6086	

Nilai Bobot dan bias antara *neuron input layer* dan *neuron hidden layer* (W_{jk})

W_{jk}		k
		1
j	0	-0,2181
	1	-0,1034
	2	0,0545
	3	0,4288

Nilai *Adaptive Gain* antara *neuron input layer* dan *neuron hidden layer* ($C1_{ak}$)

$C1_{ak}$		k		
		1	2	3
a	1	0,2563	0,1677	0,3341

Nilai *Adaptive Gain* antara *neuron input layer* dan *neuron hidden layer*($C2_{ta}$)

$C2_{ta}$		a
		1
t	1	0,1573

UNIVERSITAS BRAWIJAYA



Lampiran 3:Perbandingan hasil simulasi metode *GCFR* dengan metode *GCFRAG*

Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error Absolut	
		<i>CGFR</i>	<i>CGFRAG</i>	<i>CGFR</i>	<i>CGFRAG</i>
1.	0	-0,0026	0,0007	0,0026	0,0007
2.	0	-0,0005	0,0011	0,0005	0,0011
3.	0,1	0,0038	0,0015	0,0962	0,0985
4.	0	0,0113	0,0016	0,0113	0,0016
5.	0	0,0061	0,0017	0,0061	0,0017
6.	0	-0,0028	0,0012	0,0028	0,0012
7.	0,1	0,0136	0,0014	0,0864	0,0986
8.	0	0,0117	0,0007	0,0117	0,0007
9.	0	0,0011	0,0006	0,0011	0,0006
10.	0	0,0079	0,0022	0,0079	0,0022
11.	0,1	0,0045	0,0021	0,0955	0,0979
12.	0	0,0135	0,0006	0,0135	0,0006
13.	0	0,0046	0,0019	0,0046	0,0019
14.	0	-0,0115	0,0003	0,0115	0,0003
15.	0	0,0022	0,0007	0,0022	0,0007
16.	0,1	0,0098	0,0011	0,0902	0,0989
17.	0,1	0,0232	0,0025	0,0768	0,0975
18.	0	0,0067	0,0017	0,0067	0,0017
19.	0	-0,0005	0,0015	0,0005	0,0015
20.	0	0,0349	0,0020	0,0349	0,0020
21.	0	-0,0002	0,0008	0,0002	0,0008
22.	0	0,0114	0,0012	0,0114	0,0012
23.	0	-0,0179	0,0007	0,0179	0,0007
24.	0,1	0,0035	0,0016	0,0965	0,0984
25.	0	-0,0068	0,0006	0,0068	0,0006
26.	0	-0,0118	0,0006	0,0118	0,0006
27.	0	-0,0037	0,0005	0,0037	0,0005
28.	0	0,0035	0,0008	0,0035	0,0008
29.	0	0,015	0,0008	0,015	0,0008
30.	0	-0,007	0,0006	0,007	0,0006
31.	0	0,0051	0,0007	0,0051	0,0007
32.	0	-0,0028	0,0004	0,0028	0,0004
33.	0	0,0096	0,0014	0,0096	0,0014

Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
34.	0	-0,0022	0,0008	0,0022	0,0008
35.	0	0,0016	0,0008	0,0016	0,0008
36.	0,1	-0,0032	0,0006	0,1032	0,0994
37.	0	0,0012	0,0011	0,0012	0,0011
38.	0	0,0073	0,0017	0,0073	0,0017
39.	0	0,0189	0,0010	0,0189	0,0010
40.	0,1	0,017	0,0017	0,083	0,0983
41.	0,1	0,0266	0,0020	0,0734	0,0980
42.	0,1	0,0171	0,0011	0,0829	0,0989
43.	0,1	0,0094	0,0024	0,0906	0,0976
44.	0	0,0026	0,0009	0,0026	0,0009
45.	0	0,0054	0,0010	0,0054	0,0010
46.	0,1	0,0257	0,0020	0,0743	0,0980
47.	0,1	0,0313	0,0023	0,0687	0,0977
48.	0	0,0157	0,0011	0,0157	0,0011
49.	0	0,0174	0,0022	0,0174	0,0022
50.	0	0,0019	0,0019	0,0019	0,0019
51.	0	-0,007	0,0008	0,007	0,0008
52.	0	0,0067	0,0009	0,0067	0,0009
53.	0	0,0093	0,0021	0,0093	0,0021
54.	0	-0,0015	0,0007	0,0015	0,0007
55.	0	0,0107	0,0008	0,0107	0,0008
56.	0	0,022	0,0017	0,022	0,0017
57.	0	0,0025	0,0012	0,0025	0,0012
58.	0	0,0063	0,0019	0,0063	0,0019
59.	0	0,0101	0,0022	0,0101	0,0022
60.	0	-0,0048	0,0012	0,0048	0,0012
61.	0,1	0,0075	0,0018	0,0925	0,0982
62.	0,1	0,0312	0,0013	0,0688	0,0987
63.	0	0,0062	0,0009	0,0062	0,0009
64.	0	0,0025	0,0012	0,0025	0,0012
65.	0	-0,0075	0,0004	0,0075	0,0004
66.	0	0,0004	0,0006	0,0004	0,0006
67.	0	-0,0012	0,0008	0,0012	0,0008
68.	0	0,019	0,0018	0,019	0,0018

Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
69.	0	0,0161	0,0016	0,0161	0,0016
70.	0,1	0,0191	0,0012	0,0809	0,0988
71.	0	-0,0013	0,0009	0,0013	0,0009
72.	0	-0,0144	0,0005	0,0144	0,0005
73.	0	-0,0002	0,0009	0,0002	0,0009
74.	0	0,0078	0,0006	0,0078	0,0006
75.	0	0,0275	0,0010	0,0275	0,0010
76.	0	0,0085	0,0012	0,0085	0,0012
77.	0	0,0116	0,0013	0,0116	0,0013
78.	0,1	-0,0053	0,0004	0,1053	0,0996
79.	0	-0,0032	0,0014	0,0032	0,0014
80.	0,1	0,0196	0,0037	0,0804	0,0963
81.	0,1	0,001	0,0012	0,099	0,0988
82.	0	-0,0049	0,0012	0,0049	0,0012
83.	0	0,0094	0,0026	0,0094	0,0026
84.	0	0,0027	0,0014	0,0027	0,0014
85.	0,1	0,0559	0,0021	0,0441	0,0979
86.	0	-0,0115	0,0005	0,0115	0,0005
87.	0,1	0,003	0,0015	0,097	0,0985
88.	0	-0,0088	0,0008	0,0088	0,0008
89.	0,1	0,0236	0,0022	0,0764	0,0978
90.	0	-0,0099	0,0002	0,0099	0,0002
91.	0,1	0,0097	0,0023	0,0903	0,0977
92.	0	0,015	0,0013	0,015	0,0013
93.	0,1	0,0034	0,0013	0,0966	0,0987
94.	0	0,0054	0,0008	0,0054	0,0008
95.	0	0,0243	0,0018	0,0243	0,0018
96.	0,1	0,0151	0,0008	0,0849	0,0992
97.	0	0,0065	0,0010	0,0065	0,0010
98.	0	-0,004	0,0005	0,004	0,0005
99.	0,1	0,0037	0,0012	0,0963	0,0988
100.	0	0,0069	0,0007	0,0069	0,0007
101.	0	-0,0083	0,0007	0,0083	0,0007
102.	0	-0,0062	0,0007	0,0062	0,0007
103.	0	-0,001	0,0013	0,001	0,0013

Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
104	0,1	0,0223	0,0024	0,0777	0,0976
105	0,1	0,0038	0,0012	0,0962	0,0988
106	0	-0,0024	0,0009	0,0024	0,0009
107	0,1	0,0279	0,0013	0,0721	0,0987
108	0	-0,0049	0,0008	0,0049	0,0008
109	0	0,0292	0,0012	0,0292	0,0012
110	0	0,0107	0,0005	0,0107	0,0005
111	0	0,0122	0,0008	0,0122	0,0008
112	0,1	0,0233	0,0015	0,0767	0,0985
113	0,1	0,0346	0,0021	0,0654	0,0979
114	0	-0,0041	0,0012	0,0041	0,0012
115	0,1	0,0221	0,0023	0,0779	0,0977
116	0	-0,0093	0,0006	0,0093	0,0006
117	0	-0,0075	0,0009	0,0075	0,0009
118	0	-0,0115	0,0005	0,0115	0,0005
119	0,1	0,0044	0,0021	0,0956	0,0979
120	0,1	-0,0031	0,0004	0,1031	0,0996
121	0	0,0171	0,0013	0,0171	0,0013
122	0	-0,0079	0,0009	0,0079	0,0009
123	0	0,0061	0,0017	0,0061	0,0017
124	0	0,0104	0,0006	0,0104	0,0006
125	0	-0,0099	0,0004	0,0099	0,0004
126	0	0,0064	0,0014	0,0064	0,0014
127	0	-0,012	0,0003	0,012	0,0003
128	0	-0,0088	0,0004	0,0088	0,0004
129	0	0,0037	0,0014	0,0037	0,0014
130	0	-0,0098	0,0007	0,0098	0,0007
131	0,1	-0,0045	0,0011	0,1045	0,0989
132	0	0,0073	0,0010	0,0073	0,0010
133	0	-0,0104	0,0004	0,0104	0,0004
134	0	0,0117	0,0007	0,0117	0,0007
135	0	-0,0075	0,0010	0,0075	0,0010
136	0,1	-0,0023	0,0015	0,1023	0,0985
137	0	0,0023	0,0014	0,0023	0,0014
138	0	0,0001	0,0007	0,0001	0,0007

Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
139	0,1	0,0109	0,0015	0,0891	0,0985
140	0	-0,0048	0,0007	0,0048	0,0007
141	0	0,005	0,0007	0,005	0,0007
142	0	-0,0063	0,0006	0,0063	0,0006
143	0,1	0,0047	0,0017	0,0953	0,0983
144	0	-0,0029	0,0007	0,0029	0,0007
145	0	0,011	0,0011	0,011	0,0011
146	0	0,0441	0,0013	0,0441	0,0013
147	0,1	0,0129	0,0012	0,0871	0,0988
148	0,1	0,018	0,0011	0,082	0,0989
149	0,1	0,0162	0,0022	0,0838	0,0978
150	0	-0,0066	0,0007	0,0066	0,0007
151	0	0,0035	0,0007	0,0035	0,0007
152	0	0,0096	0,0009	0,0096	0,0009
153	0	0,0083	0,0014	0,0083	0,0014
154	0	-0,0072	0,0006	0,0072	0,0006
155	0	0,0102	0,0008	0,0102	0,0008
156	0,1	0,0514	0,0010	0,0486	0,0990
157	0	0,0019	0,0009	0,0019	0,0009
158	0	0,0249	0,0018	0,0249	0,0018
159	0	0,0057	0,0019	0,0057	0,0019
160	0,1	0,0033	0,0010	0,0967	0,0990
161	0	0,0057	0,0021	0,0057	0,0021
162	0,1	0,003	0,0014	0,097	0,0986
163	0,1	0,0286	0,0023	0,0714	0,0977
164	0,1	0,0191	0,0022	0,0809	0,0978
165	0,1	0,0063	0,0019	0,0937	0,0981
166	0	0,0122	0,0011	0,0122	0,0011
167	0,1	0,0159	0,0026	0,0841	0,0974
168	0,1	0,0018	0,0018	0,0982	0,0982
169	0	0,0242	0,0018	0,0242	0,0018
170	0	0,0162	0,0022	0,0162	0,0022
171	0	0,0248	0,0021	0,0248	0,0021
172	0	-0,0107	0,0004	0,0107	0,0004
173	0	0,0085	0,0019	0,0085	0,0019

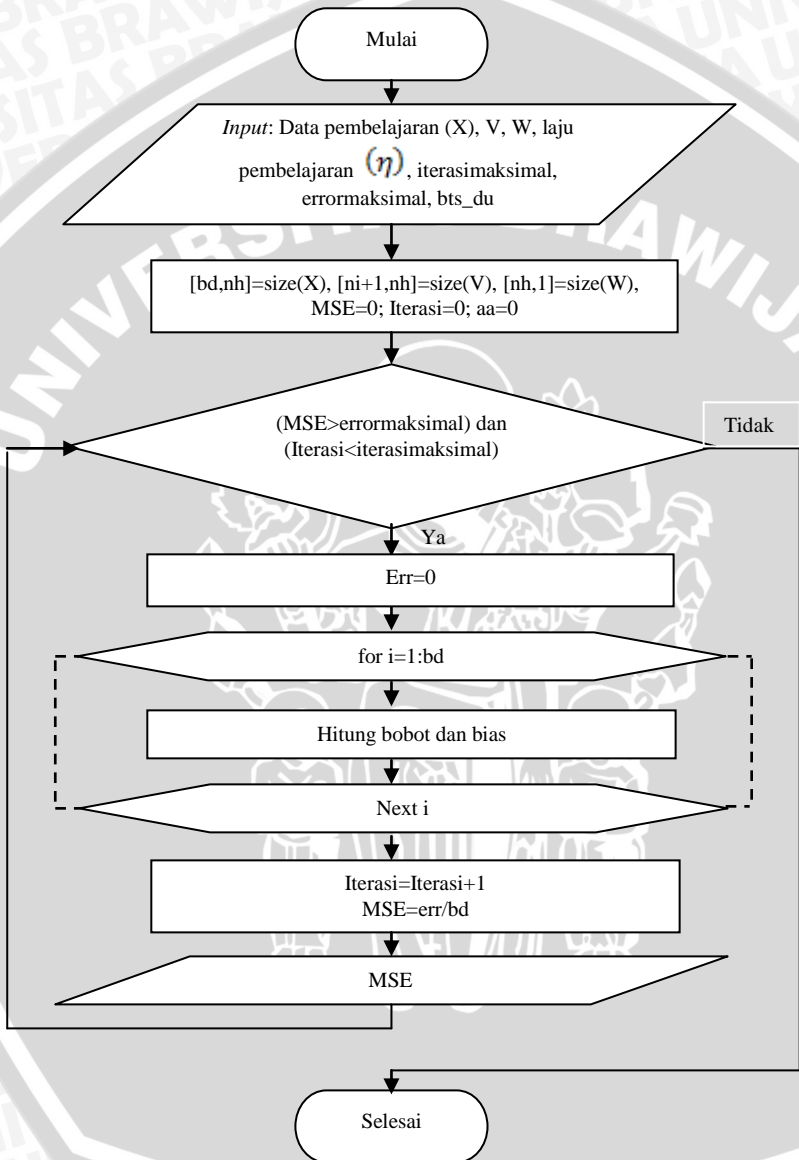
Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
174	0	0,0252	0,0011	0,0252	0,0011
175	0	0,0117	0,0022	0,0117	0,0022
176	0,1	-0,0014	0,0012	0,1014	0,0988
177	0,1	0,004	0,0020	0,096	0,0980
178	0	-0,006	0,0003	0,006	0,0003
179	0,1	-0,0042	0,0006	0,1042	0,0994
180	0	0,0191	0,0006	0,0191	0,0006
181	0	-0,0048	0,0006	0,0048	0,0006
182	0,1	0,0062	0,0012	0,0938	0,0988
183	0	0,0117	0,0009	0,0117	0,0009
184	0,1	-0,0063	0,0008	0,1063	0,0992
185	0	0,0009	0,0023	0,0009	0,0023
186	0	0,018	0,0010	0,018	0,0010
187	0	-0,0118	0,0005	0,0118	0,0005
188	0	-0,0029	0,0009	0,0029	0,0009
189	0	0,0117	0,0009	0,0117	0,0009
190	0,1	0,0289	0,0020	0,0711	0,0980
191	0	-0,007	0,0011	0,007	0,0011
192	0,1	0,004	0,0019	0,096	0,0981
193	0	0,0085	0,0010	0,0085	0,0010
194	0,1	0,0209	0,0022	0,0791	0,0978
195	0	-0,0115	0,0004	0,0115	0,0004
196	0,1	0,0467	0,0018	0,0533	0,0982
197	0,1	0,0127	0,0013	0,0873	0,0987
198	0	-0,0086	0,0002	0,0086	0,0002
199	0	0,0124	0,0009	0,0124	0,0009
200	0	-0,0018	0,0007	0,0018	0,0007
201	0	0,0187	0,0010	0,0187	0,0010
202	0,1	0,0063	0,0021	0,0937	0,0979
203	0,1	0,0107	0,0021	0,0893	0,0979
204	0	0,0082	0,0011	0,0082	0,0011
205	0	0,0055	0,0010	0,0055	0,0010
206	0	0,0002	0,0013	0,0002	0,0013
207	0,1	-0,0155	0,0010	0,1155	0,0990
208	0	0,0313	0,0007	0,0313	0,0007

Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
209	0,1	0,0041	0,0018	0,0959	0,0982
210	0,1	0,0141	0,0009	0,0859	0,0991
211	0	0,035	0,0008	0,035	0,0008
212	0	0,005	0,0017	0,005	0,0017
213	0,1	0,0074	0,0020	0,0926	0,0980
214	0	0,0235	0,0006	0,0235	0,0006
215	0	-0,0053	0,0008	0,0053	0,0008
216	0,1	0,0398	0,0018	0,0602	0,0982
217	0,1	0,0201	0,0016	0,0799	0,0984
218	0	0,0051	0,0022	0,0051	0,0022
219	0	0,0176	0,0011	0,0176	0,0011
220	0,1	0,0093	0,0020	0,0907	0,0980
221	0	-0,0032	0,0011	0,0032	0,0011
222	0	0,0186	0,0009	0,0186	0,0009
223	0,1	0,018	0,0016	0,082	0,0984
224	0	0,0168	0,0017	0,0168	0,0017
225	0	0,0019	0,0012	0,0019	0,0012
226	0	0,0065	0,0017	0,0065	0,0017
227	0	0,0163	0,0009	0,0163	0,0009
228	0	-0,0045	0,0009	0,0045	0,0009
229	0	-0,0111	0,0007	0,0111	0,0007
230	0	-0,0096	0,0003	0,0096	0,0003
231	0,1	0,0076	0,0013	0,0924	0,0987
232	0,1	-0,0114	0,0007	0,1114	0,0993
233	0,1	0,0145	0,0013	0,0855	0,0987
234	0	0,0117	0,0008	0,0117	0,0008
235	0	0,0025	0,0015	0,0025	0,0015
236	0	-0,0002	0,0012	0,0002	0,0012
237	0	0,0051	0,0008	0,0051	0,0008
238	0	0,0018	0,0016	0,0018	0,0016
239	0	0,012	0,0006	0,012	0,0006
240	0,1	0,0038	0,0010	0,0962	0,0990
241	0,1	0,023	0,0024	0,077	0,0976
242	0	0,0067	0,0009	0,0067	0,0009
243	0	0,0057	0,0006	0,0057	0,0006

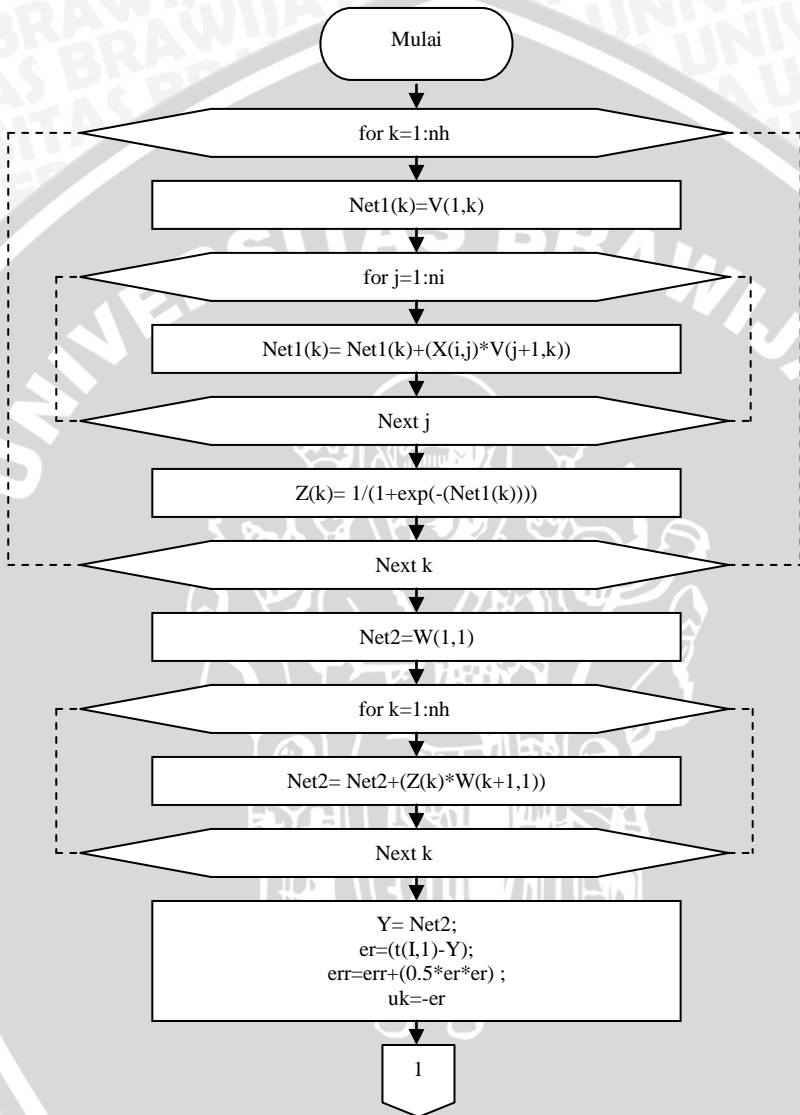
Data ke	Nilai Target (T)	Nilai Output JST (Y)		Error	
		CGFR	CGFRAG	CGFR	CGFRAG
244	0,1	0,0019	0,0017	0,0981	0,0983
245	0	0,0181	0,0023	0,0181	0,0023
246	0	0,0143	0,0022	0,0143	0,0022
247	0,1	0,0054	0,0014	0,0946	0,0986
248	0	0,01	0,0013	0,01	0,0013
249	0,1	0,0249	0,0016	0,0751	0,0984
250	0,1	0,0043	0,0018	0,0957	0,0982
251	0,1	-0,0084	0,0006	0,1084	0,0994
252	0	0,0096	0,0012	0,0096	0,0012
253	0	-0,006	0,0009	0,006	0,0009
254	0,1	0,0489	0,0013	0,0511	0,0987
255	0,1	0,0072	0,0022	0,0928	0,0978
256	0,1	0,0146	0,0016	0,0854	0,0984
257	0	0,0043	0,0020	0,0043	0,0020
258	0,1	0,0082	0,0014	0,0918	0,0986
259	0	-0,005	0,0005	0,005	0,0005
260	0,1	0,014	0,0024	0,086	0,0976
261	0	-0,0054	0,0007	0,0054	0,0007
262	0,1	0,0106	0,0022	0,0894	0,0978
263	0	-0,008	0,0010	0,008	0,0010
264	0	0,0296	0,0029	0,0296	0,0029
265	0	-0,0002	0,0011	0,0002	0,0011
266	0	0,0082	0,0012	0,0082	0,0012
267	0,1	0,0043	0,0013	0,0957	0,0987
268	0	-0,0054	0,0008	0,0054	0,0008
269		MSE		9,1952	8,6563

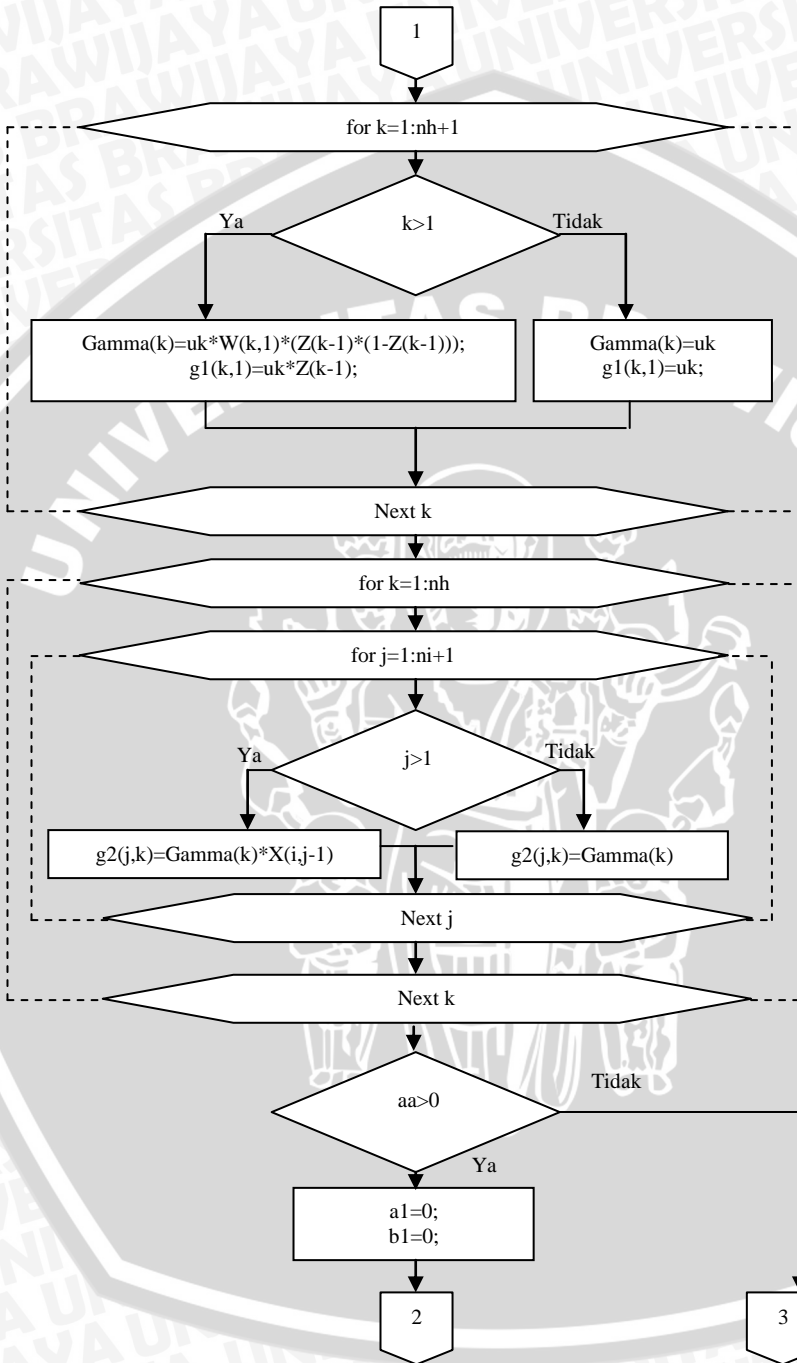
Lampiran 4:

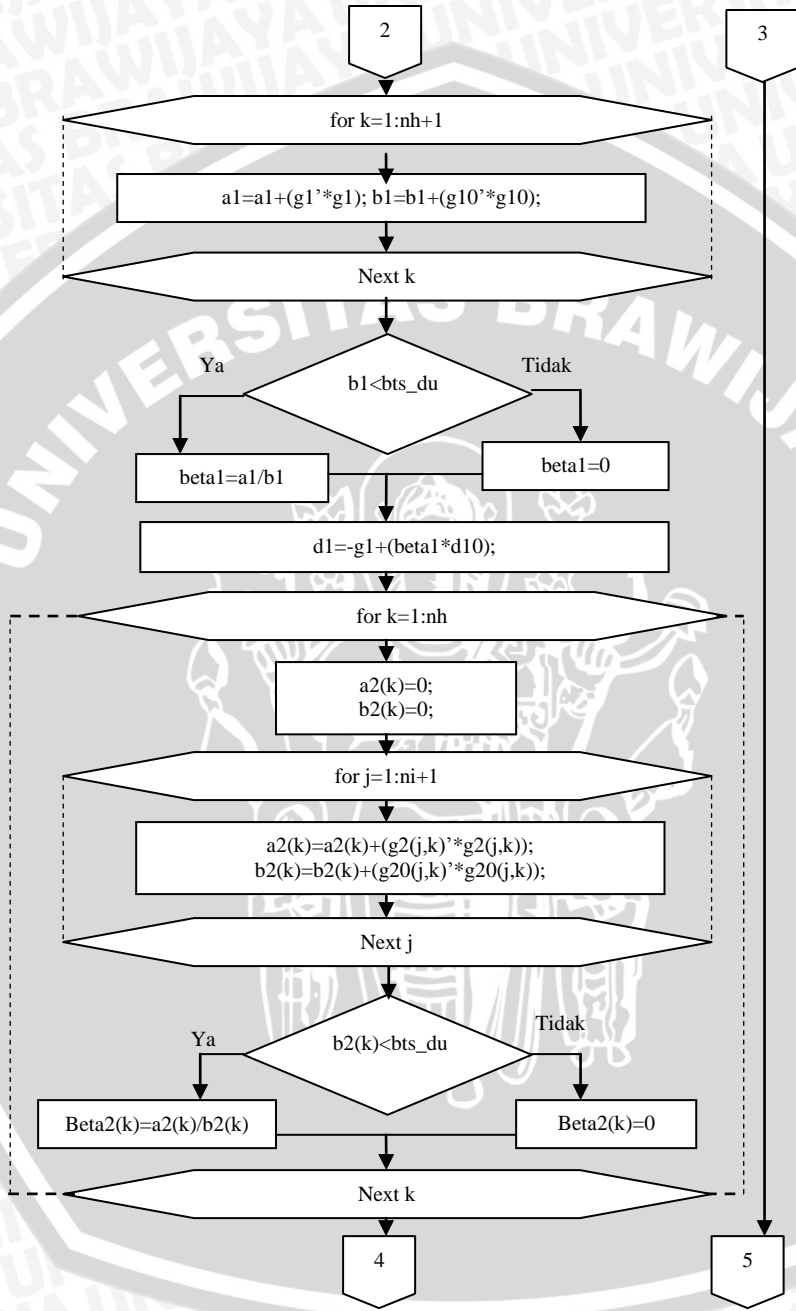
Diagram alir model *Backpropagation* dengan metode *GCFR*

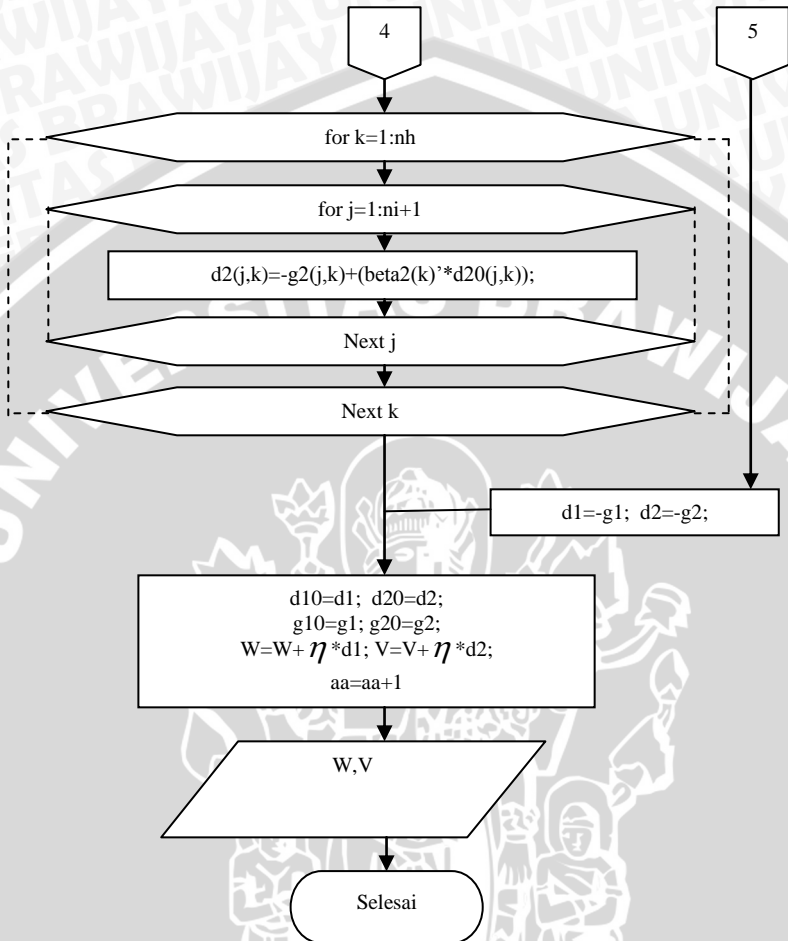


Prosedur Hitung Bobot dan Bias







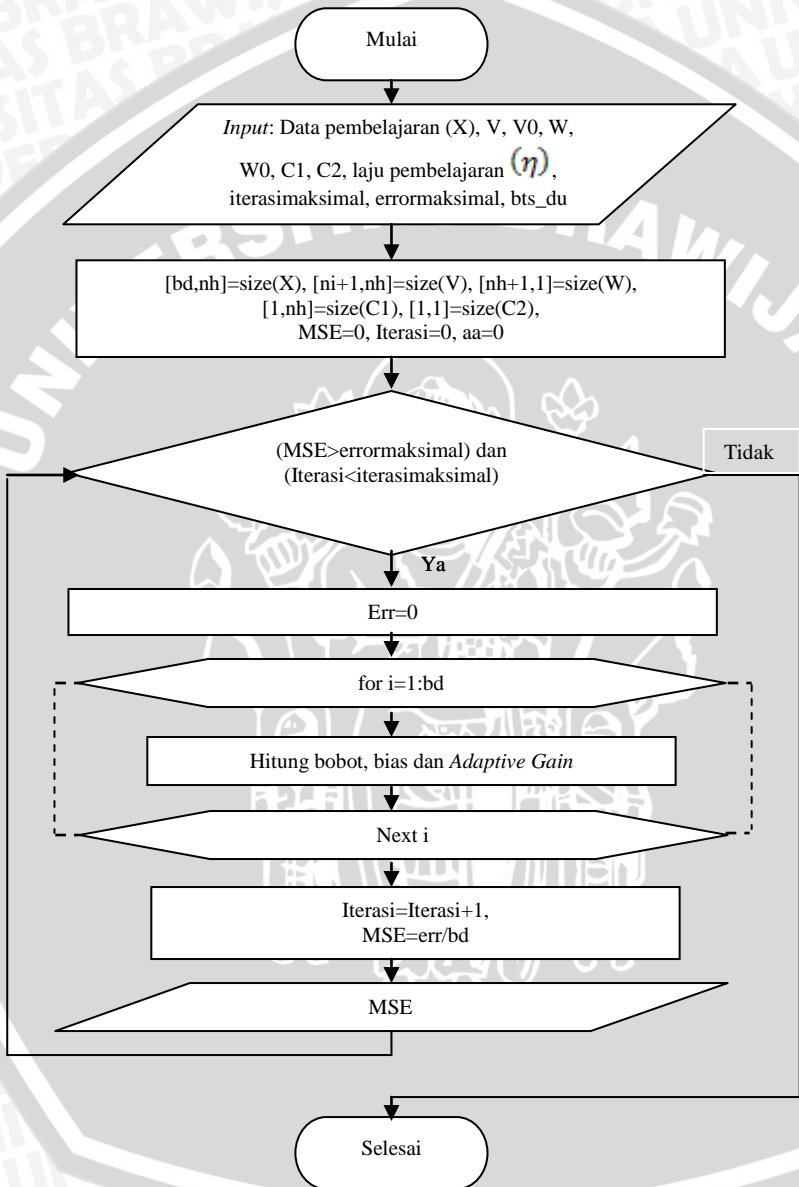


UNIVERSITAS BRAWIJAYA

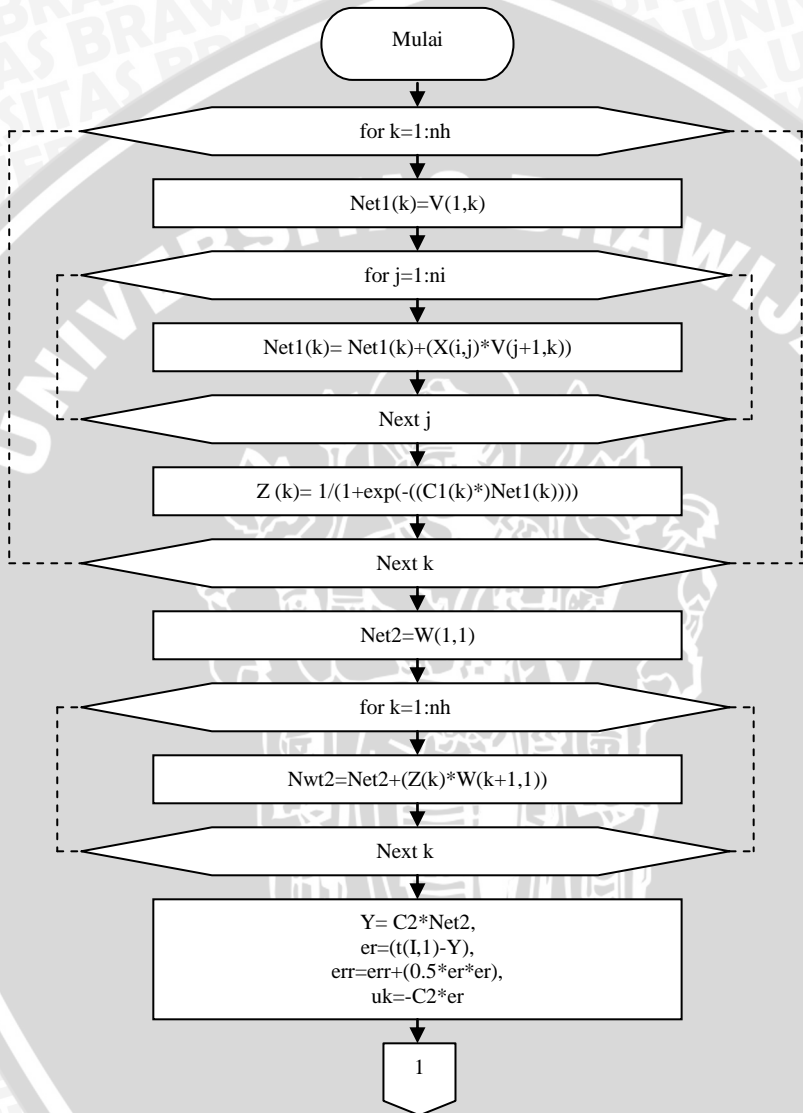


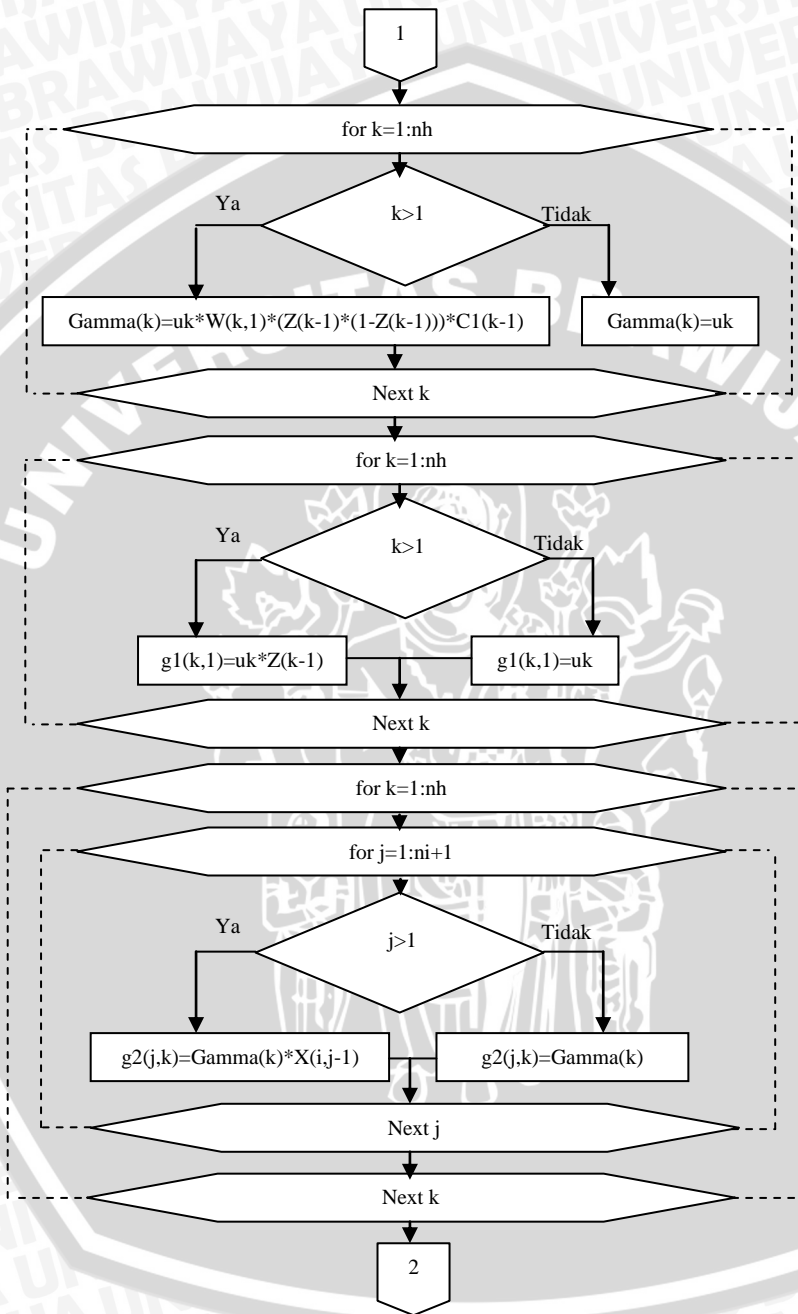
Lampiran 5:

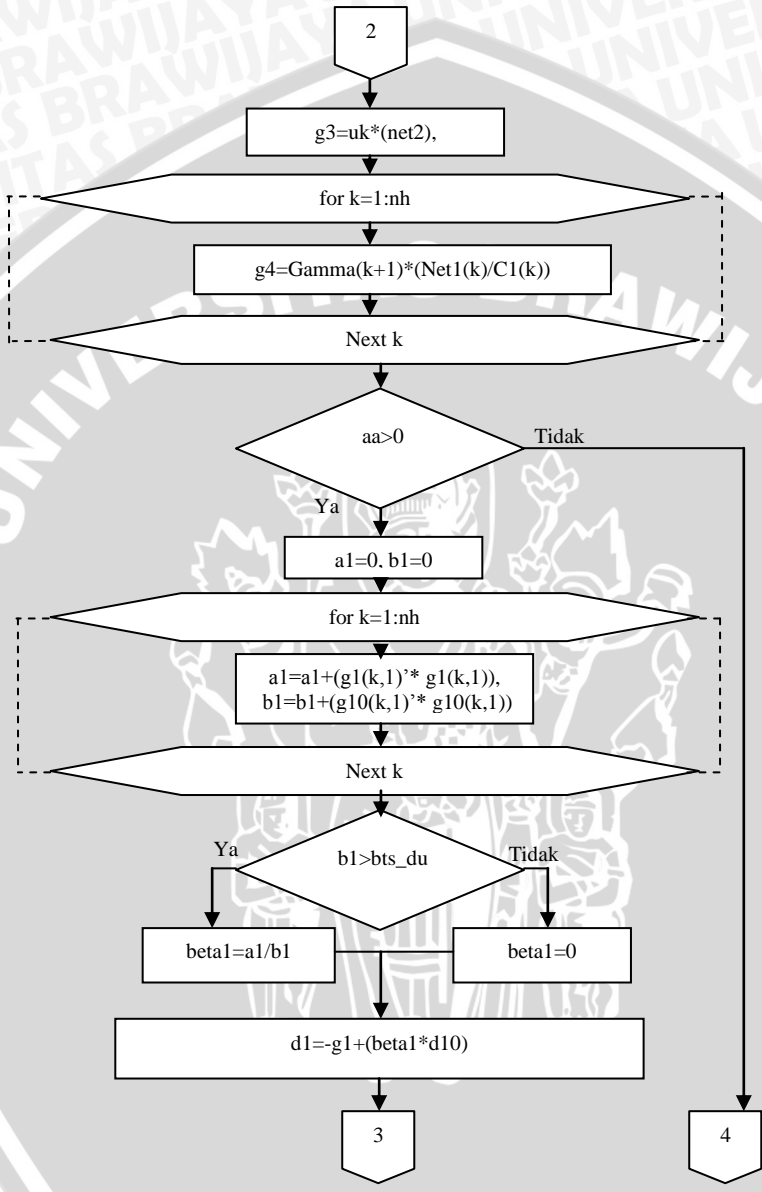
Diagram alir model *Backpropagation* dengan metode *GCFRAG*

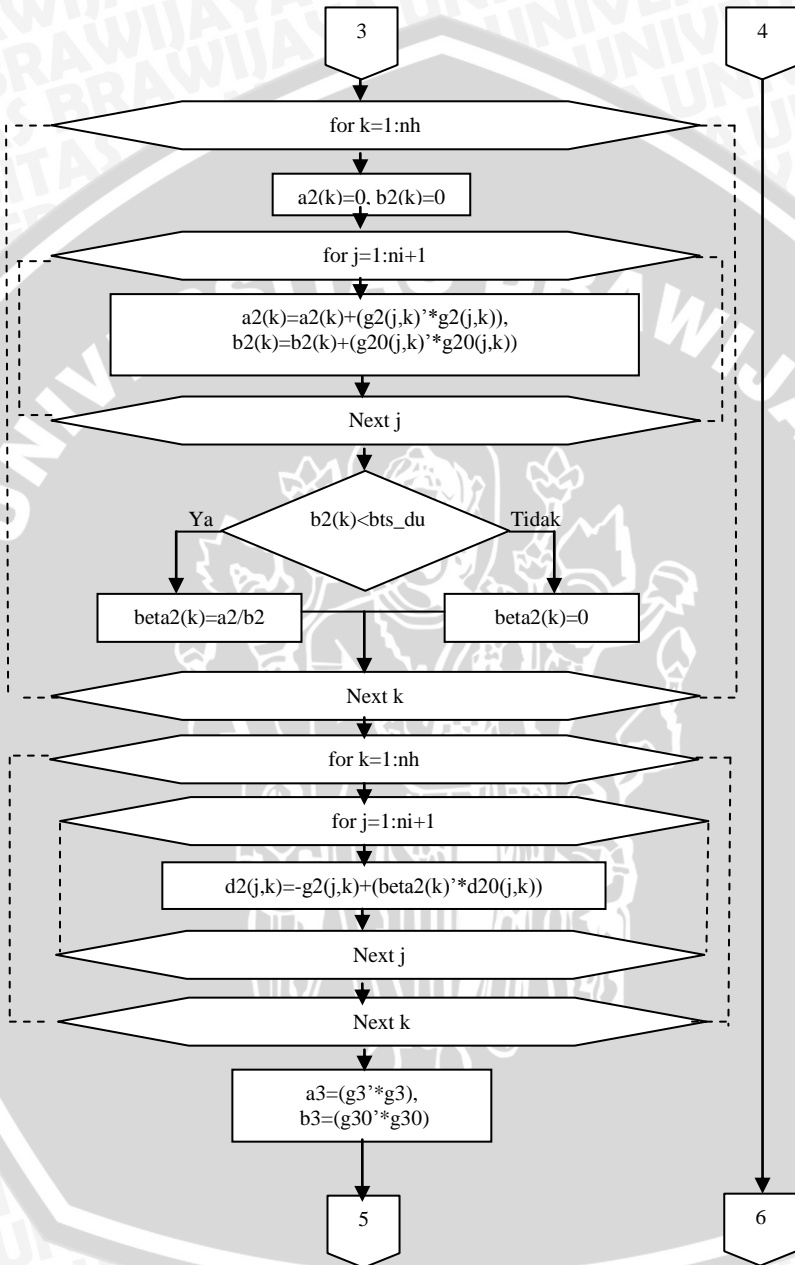


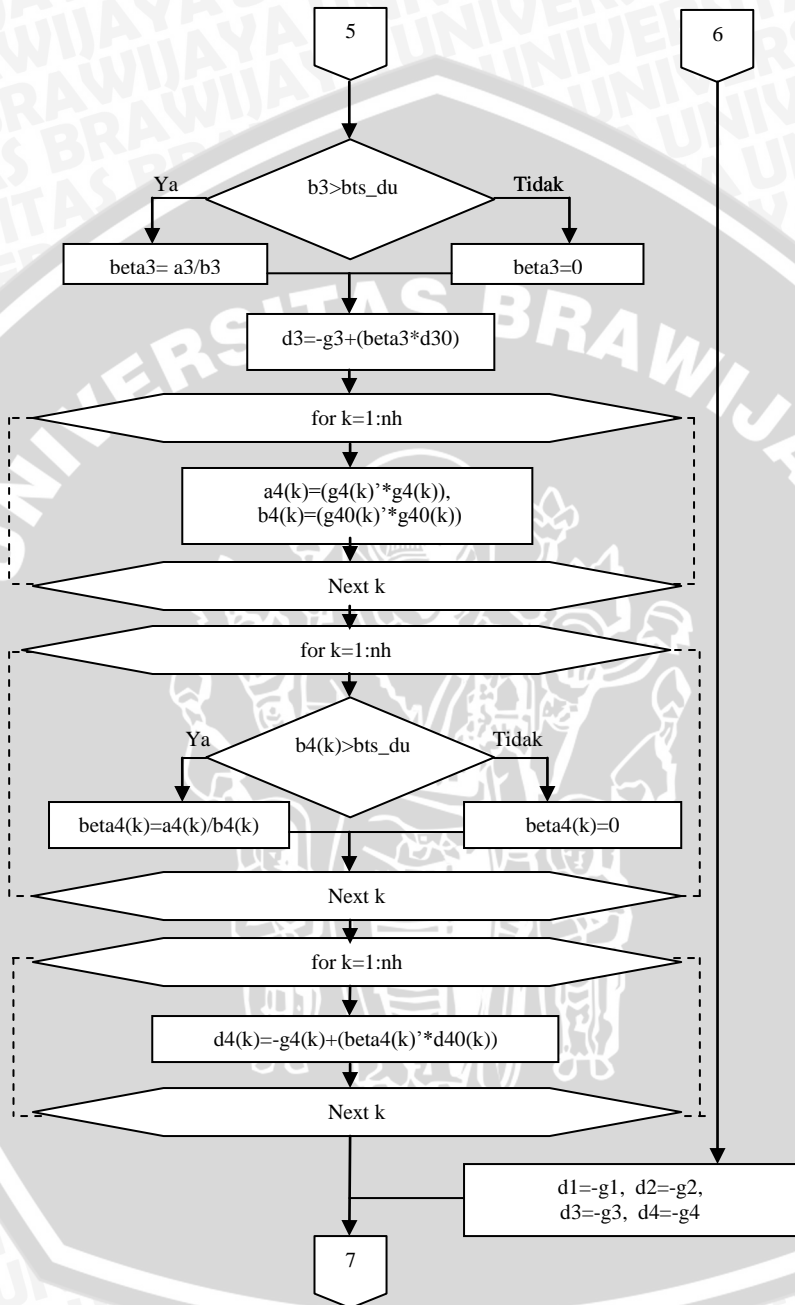
Prosedur Hitung Bobot, Bias dan *Adaptive Gain*











7

$d_{10}=d_1, d_{20}=d_2, d_{30}=d_3, d_{40}=d_4,$
 $g_{10}=g_1, g_{20}=g_2, g_{30}=g_3, g_{40}=g_4,$
 $W=W+\eta*d_1, V=V+\eta*d_2,$
 $C_2=C_2+\eta*d_3, C_3=C_3+\eta*d_4,$
 $aa=aa+1$

W,V,C2,C1

Selesai

UNIVERSITAS BRAWIJAYA

