Edisi.,Volume,. Bulan.. ISSN : 2089-9033

ANALISIS METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION UNTUK PENGENALAN SEL KANKER OTAK

Novita Handayani

Teknik Informatika - Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112-116, Bandung 402123
E-mail handayani.novita1991@gmail.com

ABSTRAK

Berdasarkan hasil studi literatur penggunaan metode *backpropagation* untuk identifikasi penyakit melalui pengenalan pola citra medis, sebagai contoh penelitian yang dilakukan untuk identifikasi penyakit TBC (*Tuberculosis*) tingkat akurasi yang dihasilkan pada penelitian ini sebesar 77.5%. Penelitian lain yang menggunakan metode *backpropagation* adalah penelitian untuk mengklasifikasikan kanker payudara didapatkan akurasi sebesar 97%. Penelitian untuk mengidentifkasi penyakit tumor otak menggunakan jaringan syaraf tiruan *radial basic function* didapatkan tingkat akurasi sebesar 80% dalam mengidentifikasi tumor otak. Belum ditemukan penelitian untuk mengenali pola sel kanker otak menggunkan metode *backpropagation*, oleh sebab itu, perlu dilakukan penelitian untuk membuktikan apakah metode *backpropagation* cocok untuk mengenali pola kanker otak.

Metode yang digunakan adalah jaringan syaraf tiruan- *backpropagation*. Metode *backpropagation*. Metode pendekatan yang digunakan adalah terstruktur dan metode pengembangannya menggunakan model *waterfall*.
Simulator dibangun menggunakan Microsoft Visual Studio 2010 dengan menggunakan bahasa C#.

Berdasarkan hasil pengujian dengan menggunakan tiga parameter pembelajaran yang berbeda, maka diperoleh kesimpulan tingkat akurasi yang dihasilkan pada penelitian ini masih kurang baik dalam mengenali sel kanker otak dengan tingkat akurasi rata-rata 57% dengan rata-rata waktu pengenalan 29.5 *milisecond dengan* menggunakan 10 data uji dengan 5 data normal dan 5 data kanker.

Kata Kunci : Pengenalan Sel Kanker Otak, Jaringan Syaraf Tiruan *Backpropagation*, Citra Digital.

1. PENDAHULUAN

Metode jaringan syaraf tiruan *backpropagation* merupakan metode yang banyak digunakan untuk diaplikasikan pada penyelesaian suatu masalah berkaitan dengan identifikasi, prediksi, dan pengenalan pola. Penelitian yang menggunakan metode *backpropagation*, salah satunya untuk mengidentifikasi suatu penyakit melalui pengenalan pola pada citra medis.

Berdasarkan hasil studi literatur penggunaan metode *backpropagation* untuk identifikasi penyakit melalui pengenalan pola citra medis contohnya penelitian yang dilakukan untuk identifikasi penyakit TBC (*Tuberculosis*). Langkah-langkah pengenalan pola penyakit TBC adalah *preprocessing* yang terdiri dari *grayscale* dan *threshold*, lalu pembelajaran menggunakan metode *backpropagation*. Tingkat akurasi yang dihasilkan pada penelitian ini sebesar 77.5% dalam pengenalan pola penyakit TBC menggunakan metode *backpropagation* [1]. Penelitian lain yang menggunakan metode *backpropagation* adalah penelitian untuk mengklasifikasikan kanker payudara. Penelitian ini bertujuan untuk mencari optimasi parameter pembelajaran yang paling baik dengan menggunakan algoritma genetika, optimasi parameter pembelajaran bertujuan untuk meningkatkan akurasi. Langkah-langkah untuk mengklasifikasikan kanker payudara adalah preprocessing dataset, pembangkitan populasi awal, pembelajaran backpropagation, lalu identifikasi, pada penelitian ini didapatkan akurasi sebesar 97% dalam mendeteksi kanker payudara [2]. Penelitian untuk mengidentifkasi penyakit tumor otak pernah dilakukan dengan menggunakan jaringan syaraf tiruan *radial basic function*, langkah-langkah proses pengidentifikasian terdiri dari preprocessing, ekstraksi cirri statistik, lalu pembelajaran menggunakan metode *radial basis function*, pada penelitian ini didapatkan tingkat akurasi sebesar 80% dalam mengidentifikasi tumor otak [3].

Metode *backpropagation* merupakan salah satu metode pembelajaran dari Jaringan Syaraf Tiruan (JST). Algoritma ini akan menghasilkan kinerja yang lebih baik karena latihan yang berulang–ulang. Metode *backpropagation* dipilih karena kemampuannya untuk belajar dan menangani nilai tersembunyi yang berada di dataset. Dataset

adalah keseluruhan data *training*, dengan kelebihan tersebut dapat mewujudkan sistem yang konsisten bekerja dengan baik.

Penelitian-penelitian yang pernah dilakukan sebelumnya dengan menggunakan metode jaringan syaraf tiruan untuk identifikasi suatu penyakit melalui pola citra medis telah berhasil dilakukan, tetapi belum ditemukan penelitian untuk mengenali pola sel kanker otak menggunkan metode *backpropagation*, oleh sebab itu, perlu dilakukan penelitian untuk membuktikan apakah metode *backpropagation* cocok untuk mengenali pola kanker otak.

Tujuan dari penelitian yang dilakukan dalam menganalisis metode jaringan syaraf tiruan *backpropagation* dalam pengenalan sel kanker otak untuk mengetahui performansi dan akurasi pada pengenalan citra kanker otak menggunakan metode jaringan syaraf tiruan *backpropagation.*

2. ISI PENELITIAN

2.1 Metode yang digunakan

Pada penelitian ini metode yang digunakan meliputi preprocessing dan metode jaringan syaraf tiruan *backpropagation.*

2.1.1 Preprocessing

Preprocessing terdiri dari *scaling, grayscale, dan thresholding.*

1. Scaling

Scaling atau Penskalaan citra adalah sebuah operasi geometri yang memberikan efek memperbesar atau memperkecil ukuran citra *input* sesuai dengan variabel penskalaan citranya. *Scaling* digunakan untuk memperbesar (*zoom-in*) atau memperkecil (*zoom-out*) citra [5]. Rumus yang digunakan untuk proses *scaling* terlihat seperti persamaan 1.

$$x' = Sh\ x \quad y' = Sv\ y \qquad (1)$$

Keterangan :
Sh : faktor skala horizontal
Sv : faktor skala vertikal

2. Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian *RED* = *GREEN* = *BLUE*. [5].
Rumus yang digunakan untuk grayscale terlihat seperti persamaan 2.

$$Grayscale = (R + G + B)/3 \qquad (2)$$

3. Tresholding

Tresholding atau pengambangan membuat citra memiliki dua tingkat keabuan yaitu hitam dan putih, proses pengambangan akan menghasilkan citra biner [6]. Proses tresholding mengikuti aturan dari persamaan 3.

$$g(x,y) = \begin{cases} 1 \ if \ f(x,y) \geq T \\ 0 \ if \ f(x,y) < T \end{cases} \qquad (3)$$

2.1.2. Metode Jaringan Syaraf Tiruan *Backpropagation*

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceprton dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid, seperti terlihat pada rumus 4. merupakan fungsi aktivasi sigmoid biner, yaitu [4] :

$$f(x) = \frac{1}{1+e^{-x}} \qquad (4)$$

Algoritma *backpropagation* [4]:
Inisialisasi bobot (ambil bobot awal dengan nilai random).
Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai FALSE:

1. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Feedforward:

a. Tiap-tiap unit input (X_i, i=1,2,3,...,n) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

b. Tiap-tiap unit tersembunyi (Z_i, j=1,2,3,...,p) menjumlahkan sinyal-sinyal input terbobot:

$$z_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \qquad (5)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z_in_j) \qquad (6)$$

kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

c. Tiap-tiap unit output (Y_k, k=1,2,3,...,m) menjumlahkan sinyal-sinyal input terbobot.

$$y_in_k = w_{0k} + \sum_{i=1}^{p} z_i w_{jk} \qquad (.7)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_in_k) \qquad (8)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Backpropagation

d. Tiap-tiap unit output (Y_k, k=1,2,3,...,m) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya:

$$\delta_k = (t_k - y_k)\, f'(y_in_k) \qquad (9)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

$$\Delta w_{jk} = \alpha\, \delta_k\, z_j \qquad (10)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}):

$$\Delta w_{0k} = \alpha\, \delta_k \qquad (11)$$

kirimkan δ_k ini ke unit-unit yang ada di lapisan bawahnya.

e. Tiap-tiap unit tersembunyi (Z_j, j=1,2,3,...,p) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \qquad (.12)$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$$\delta_j = \delta_in_j\, f'(z_in_j) \qquad (13)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{jk} = \alpha\, \delta_j\, x_i \qquad (.14)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}):

$$\Delta v_{0j} = \alpha\, \delta_j \qquad (15)$$

f. Tiap-tiap unit output (Y_k, k=1,2,3,...,m) memperbaiki bias dan bobotnya (j=0,1,2,...,p):

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \qquad (16)$$

Tiap-tiap unit tersembunyi (Z_j, j=1,2,3,...,p) memperbaiki bias dan bobotnya (i=0,1,2,...,n):

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \qquad (17)$$

2. Tes kondisi berhenti

2.1.3 Analisis Algoritma Asimptotik

Kompleksitas algoritma terdiri dari waktu dan ruang. Kompleksitas waktu asimptotik merupakan waktu yang dibutuhkan suatu Algoritma menyelesaikan tiap langkahnya. Setiap Algoritma memiliki kompleksitas waktu yang berbeda-beda. Komplesitas waktu asimptotik diperlukan untuk menghitung performansi suatu Algoritma. Untuk menghitung kompleksitas waktu asimptotik suatu Algoritma digunakanlah notasi "O-Besar" (*Big-O*) yang merupakan notasi kompleksitas waktu asimptotik. Definisi dari Big-O atau O(g(n)) adalah kumpulan semua fungsi yang *order of growth*-nya lebih kecil atau sama dengan g(n), sedangkan definisi dari *order of growth* adalah istilah yang dapat digunakan untuk pola varian jumlah input dalam suatu pengujian algoritma [6]

Berikut ini adalah pengelompokan algoritma berdasarkan notasi *O*-besar dapat dilihat pada Tabel1 [6].

Tabel 1. Pengelompokan Algoritma Berdasarkan Notasi *O*-Besar [6]

Kelompok Algoritma	Nama
$O(1)$	konstan
$O(\log n)$	logaritmik
$O(n)$	lanjar
$O(n \log n)$	n log n
$O(n^2)$	kuadratik
$O(n^3)$	kubik
$O(2^n)$	eksponensial
$O(n!)$	faktorial

Urutan spektrum kompleksitas waktu algoritma adalah :

$O(1)<O(\log n) < O(n)< O(n \log n) < O(n^2)\ O(n^3)<...<O(2^n)<O(n!)$

algoritma polinomial — algortima eksponensial

Penjelasan masing-masing kelompok algoritma dapat dilihat pada Tabel 2.

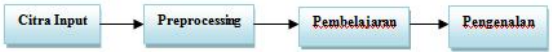
Tabel 2 Kelompok Algoritma dan Penjelasannya[6]

Kelompok Algortima	Penjelasan
O(1)	Kompleksitas *O*(1) berarti waktu pelaksanaan algoritma adalah tetap, tidak bergantung pada ukuran masukan.
$O(n)$	Algoritma yang waktu pelaksanaannya lanjar umumnya terdapat pada kasus yang setiap elemen masukannya dikenai proses yang sama, misalnya algoritma pencarian_beruntun. Bila *n* dijadikan dua kali semula, maka waktu pelaksanaan algoritma juga dua kali semula.
$O(n^2)$	Algoritma yang waktu pelaksanaannya kuadratik hanya praktis digunakan untuk persoalana yang berukuran

	kecil. Umumnya algoritma yang termasuk kelompok ini memproses setiap masukan dalam dua buah kalang bersarang, misalnya pada algoritma urut_maks. Bila *n* = 1000, maka waktu pelaksanaan algoritma adalah 1.000.000. Bila *n* dinaikkan menjadi dua kali semula, maka waktu pelaksanaan algoritma meningkat menjadi empat kali semula.
$O(2^n)$	Algoritma yang tergolong kelompok ini mencari solusi persoalan secara "*brute force*", misalnya pada algoritma mencari sirkuit Hamilton. Bila *n* = 20, waktu pelaksanaan algoritma adalah 1.000.000. Bila *n* dijadikan dua kali semula, waktu pelaksanaan menjadi kuadrat kali semula!

2.2 Analisis Algoritma

Proses yang digunakan pada penelitian ini terlihat seperti gambar 1



Gambar 1 Proses Pengenalan Kanker Otak

2.2.1 Citra Input

Citra input merupakan citra MRI otak yang terdiri dari citra MRI otak normal dan citra MRI kanker otak. Citra MRI pada awalnya belum berupa citra digital, jadi diperlukan proses akuisisi citra menggunakan kamera digital. Citra input diubah ekstensinya dari jpeg menjadi bmp menggunakan photoshop CS5.

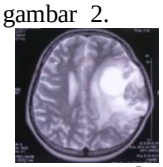
2.2.2 Preprocessing

1. Scaling

Scaling merupakan proses penskalaan pada gambar, pada proses penskalaan gambar, gambar dapat diperbesar atau diperkecil, pada penelitian ini gambar akan diperkecil. Gambar yang akan menjadi data latih dan data uji diperkecil menggunakan library aforge dengan ukuran lebar 45 pixel dan panjang 45 pixel.

2. Grayscale

Grayscale merupakan proses pengubahan citra menjadi citra keabuan. Sebagai contoh kasus, terdapat citra MRI kanker otak berukuran 45 x 45 pixel terlihat seperti gambar 2.



Gambar 2. Citra MRI Kanker Otak Ukuran 45 x 45 Pixel

Sebagai contoh perhitungan diambil matriks 3 x 3 pixel dari citra MRI kanker otak diambil pada koordinat ((25,25), (25,26), (25,27), (26,25), (26,26), (26,27), (27,25), (27,26), (27,27)), terlihat seperti gambar 3, matriks dari citra tersebut akan diubah menjadi citra keabuan.

x/y	25	26	27
25	(175,169,197)	(174,168,196)	(166,157,188)
26	(200,194,222)	(200,194,222)	(195,186,217)
27	(195,189,215)	(200,194,222)	(202,193,222)

Gambar 3 Matriks Citra MRI Kanker Otak Ukuran 3x3 Pixel

Langkah-langkah perhitungan *grayscale* adalah sebagai berikut:

a. Setiap pixel dihitung nilai *grayscale* nya

$$\text{grayscale} = \frac{R + G + B}{3}$$

b. Nilai RGB setiap pixel diulang dengan nilai *grayscale* yang dihasilkan pada perhitungan langkah a. Hasil dari proses *grayscale* terlihat seperti gambar 4



Gambar 4 Citra Kanker Otak Hasil *Grayscale*

Matriks citra *grayscale* pada gambar 4, terlihat pada gambar 5, nilai RGB pada masing-masing pixel telah sama.

x/y	25	26	27
25	180	179	170
26	205	205	199
27	200	205	206

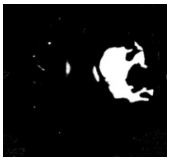
Gambar 5 Matriks Citra Kanker Otak Hasil *Grayscale*

3. Threshold

Threshold merupakan proses pengubahan citra menjadi citra biner atau citra hitam putih. Citra masukan yang akan diubah menjadi citra biner adalah citra kanker otak pada gambar 4. Proses *threshold* dilakukan dengan langkah-langkah sebagai berikut :

a. Tentukan nilai *threshold* (T) dengan rentang 0-255, dalam penelitian ini diambil nilai T=190, karena jika nilai T< 190 maka *background* dari otak terlalu banyak, sedangkan apabila T >190 maka pixel kanker terlalu banyak yang hilang
b. Jika nilai pixel lebih dari atau sama dengan 190 maka ubah nilai pixel pada citra menjadi 1, jika nilai pixel kurang dari 190 maka ubah nilai pixel menjadi 0.

Hasil proses *threshold* terlihat pada gambar 6



Gambar 6 Citra Kanker Otak Hasil *Threshold*

Matriks citra kanker otak hasil *threshold* terlihat seperti gambar 7

x/y	25	26	27
25	0	0	0
26	1	1	1
27	1	1	1

Gambar 7 Matriks Kanker Otak Hasil *Threshold*

2.2.3 Pembelajaran

Metode pembelajaran yang digunakan pada penelitian ini adalah jaringan syaraf tiruan *backpropagation* dengan menggunakan dua perambatan yaitu perambatan maju dan perambatan mundur. Masukan pada proses pembelajaran merupakan matriks hasil *threshold* citra kanker otak pada gambar 6. Selanjutnya, matriks tersebut diubah menjadi vektor terlihat seperti gambar 8

0
0
0
1
1
1
1
1
1

Gambar 8 Vektor Citra Kanker Otak

Ditambahkan satu citra otak normal dengan matriks 3x3, diketahui vektor dari citra tersebut seperti terlihat pada gambar 9

1
0
0
0
0
0
0
0
0

Gambar 9 Vektor Citra Otak Normal

Vektor dari citra kanker otak dan citra otak normal kemudian disatukan untuk kemudian dilatih. Terlihat seperti gambar 10

0	1
0	0
0	0
1	0
1	0
1	0
1	0
1	0
1	0

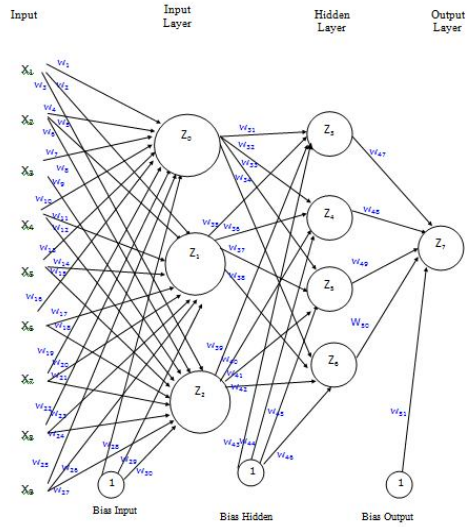
Gambar 10 Gabungan Vektor Kanker Otak dan Otak Normal

Dengan target seperti gambar 11, 1 merupakan klasifikasi kanker dan 0 merupakan klasifikasi normal

1	0

Gambar 11 Target Output

Menurut buku yang berjudul *machine learning* karangan Tom M. Mitchell [7], pada buku tersebut dijelaskan mengenai pembangunan sebuah mesin pembelajaran untuk pengenalan wajah, arsitektur yang digunakan pada pembelajarannya menggunakan 2 layer dan jumlah unit sebanyak input yang akan direduksi, sedangkan pada penelitian ini jumlah layer yang digunakan yaitu 3 layer dan 8 unit *perceptron*. Arsitektur *backpropagation* terlihat seperti gambar 3.12, terdapat 9 input (x_1, x_2,x_3, x_4 ,x_5 ,x_6 ,x_7 ,x_8 ,x_9,), 3 input layer (z_0, z_1, z_2), 4 hidden layer (z_3, z_4, z_5, z_6), dan satu output layer (z_7). Z adalah penamaan untuk *perceptron*. Arsitektur backpropagation yang digunakan pada penelitian ini terlihat seperti gambar 12.



Gambar 12. Arsitektur *Backpropagation* Yang Digunakan

Langkah-langkah proses pembelajaran adalah sebagai berikut :

1. Tentukan nilai maksimum perulangan, rasio pembelajaran, dan error minimum. Pada contoh perhitungan ini digunakan batasan sebagai berikut:
   Maksimum perulangan = 2
   Rasio pembelajaran = 0.5, dengan *range* 0-1, maka dalam contoh kasus ini dipilih nilai 0.5 sebagai rasio pembelajaran
   Error minimum = 0.01
2. Inisialisasikan bobot awal dengan nilai random dengan interval 0-1
3. Perulangan dilakukan selama nilai perulangan lebih kecil dari maksimal perulangan dan kuadrat error lebih besar dari nilai error minimum. Setiap data dilakukan perulangan perhitungan langkah 4 dan langkah 5.
4. Perambatan maju *(feedforward)*
   perulangan ke-1
   Data ke-1
5. Perambatan Mundur *(backpropagation)*

Pada data kedua dilakukan operasi-operasi yang sama dengan menggunakan bobot-bobot akhir hasil pengolahan data pertama ini sebagai bobot-bobot awalnya. Proses ini dilakukan secara berulang sampai pada maksimum perulangan atau kuadrat error < target error.

2.2.4 Pengenalan

Pengenalan merupakan proses yang dilakukan untuk mengenali citra uji. Sebagai contoh kasus terdapat citra kanker otak untuk citra uji berukuran 3 x 3 pixel yang telah di-*preprocessing* terlebih dahulu, dan diubah dari matriks menjadi array satu dimensi terlihat seperti gambar 13

0
0
0
0
1
1
1
1
1

Gambar 13 Array Citra Uji

Langkah –langkah proses pengenalan adalah sebagai berikut :

1. Lakukan proses pembelajaran terlebih dahulu menggunakan perambatan maju dan perambatan mundur, proses pembelajaran sama dengan penjelasan pada sub-bab 3.2.3. Proses pembelajaran telah dilakukan dan didapatkan bobot-bobot hasil pembelajaran.
2. Setelah didapatkan bobot hasil pembelajaran, lakukan proses pengenalan dengan menggunakan perambatan maju (*Feedforward*)
3. Nilai z_7 hasil perhitungan pada proses *feedforward* menjadi hasil pengenalan. Diketahui batas ambang (*threshold*) *Threshold*=0.5, artinya jika nilai y >= 0.5 maka dijadikan 1. Target 1 merupakan target untuk klasifikasi kanker , dan sebaliknya jika nilai y < 0.5 maka dijadikan 0. Target 0 merupakan target untuk klasifikasi normal. Nilai y={0.9379}, maka hasil z_7={1}, artinya data uji merupakan citra dengan klasifikasi kanker.

2.2.5 Analisis Kompleksitas Waktu Asimptotik Metode *Backpropagation*

Perhitungan kompleksitas waktu asimptotik metode *backpropagation* dapat dihitung dengan menelusuri setiap langkah metode *backpropagation* pada *pseudocode*-nya, sebelum dilakukan proses pembelajaran menggunakan metode *backpropagation* dilakukan proses *preprocessing*. Berikut adalah algoritma *preprocessing* sebelum dilakukan proses pembelajaran menggunakan metode *backpropagation.*

Berdasarkan hasil perhitungan Big-O dari *grayscale* adalah

O (n) . O (n) + O(1) + O(1) + O(1) + O(1) + O(1) = O(n^2)

Hasil dari perhitungan notasi Big-O dari *graysale* adalah O(n^2) dan termasuk kedalam kelompok kuadratik, waktu pelaksanaan algoritmanya menjadi 4 kali semula.

Berdasarkan hasil perhitungan Big-O dari *threshold* adalah

O (n) . O (n) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) = O (n^2)

Hasil dari perhitungan notasi Big-O dari *threshold* adalah O(n^2) dan termasuk kedalam kelompok kuadratik, waktu pelaksanaan algoritmanya menjadi 4 kali semula.

Setelah dilakukan *preprocessing* maka dilakukan perhitungan menggunakan metode *backpropagation* yang terdiri dari proses pembelajaran dan pengenalan. *Pseudocode* pembelajaran *backpropagation* dan perhitungan nilai Big-O terlihat pada tabel 3.5

Best Case = 100n { 1 + n (n (1 + 1 + n . 1) + 1 + 1 + n . 1 + n . 1 + 1 + 1 + 1 + 1 + 1 + n . 1 + n . n . 1 + n . 1 + n . 1 + n. 1 + n.n.1 + n.n.1 + n.n.1+1} = 100n + n^{20}

Worst Case = 1000n { 1 + n (n (1 + 1 + n . 1) + 1 + 1 + n . 1 + n . 1 + 1 + 1 + 1 + 1 + 1 + n . 1 + n . n . 1 + n . 1 + n . 1 + n. 1 + n.n.1 + n.n.1 + n.n.1+1}=1000n+n^{20}

Didapatkan kompleksitas waktu asimptotik dari proses pembelajaran yaitu:

O (1000) + O (n^{20}) = O (n^{20})

O (n^{20}) hampir mirip dengan O (n^2) termasuk kedalam kelompok kuadratik, waktu pelaksanaan algoritmanya menjadi 4 kali semula.
Berdasarkan hasil perhitungan Big-O dari proses pengenalan adalah

n . 1 + 1 + n . 1 + 1 + n. 1 + n . 1 + 1 = 4n

Karena $n \geq 1$ maka termasuk O (n). Nilai dari kompleksitas pengenalan metode *backpropagation* yang dihasilkan dari perhitungan Big-O adalah O(n) atau disebut lanjar. Algoritma yang waktu pelaksanaannya lanjar maka waktu pelaksanaan algoritmanya dua kali semula. Lanjar termasuk dalam urutan spektrum kompleksitas waktu algoritma polinomial yaitu algoritma yang performansinya baik.

2.3 Hasil Penelitian

Rencana pengujian performansi yang akan dijalankan pada simulator ini merupakan pengujian dengan menggunakan berbagai kombinasi parameter pembelajaran, terlihat pada tabel 3

Tabel 3 Rencana Pengujian Performansi

Kombinasi	Parameter	Nilai
I	Maksimal Perulangan	50
	Rasio Pembelajaran	0.5
	Minimal Error	0.1
II	Maksimal Perulangan	100
	Rasio Pembelajaran	1
	Minimal Error	0.01
III	Maksimal Perulangan	500
	Rasio Pembelajaran	1
	Minimal Error	0.001

Pengujian untuk kombinasi parameter I terlihat pada tabel 4.

Tabel 4. Hasil Pengujian Kombinasi Parameter I

No	Nama Gambar	Dikenali oleh sistem	Akurasi	Waktu (ms)
1	Kanker1	Kanker	60%	29
2	Kanker2	Kanker	58%	30
3	Kanker3	Kanker	62%	30
4	Kanker4	Kanker	62%	30
5	Kanker5	Kanker	62%	30
6	Normal1	Normal	62%	30
7	Normal2	Normal	62%	29
8	Normal3	Normal	38%	30
9	Normal4	Normal	45%	31
10	Normal5	Normal	64%	31
Rata -rata			57.5%	30

Pengujian untuk kombinasi parameter II terlihat pada tabel 5

Tabel 5 Hasil Pengujian Kombinasi Parameter II

No	Nama Gambar	Dikenali oleh sistem	Akurasi	Waktu (ms)
1	Kanker1	Kanker	62%	30
2	Kanker2	Kanker	62%	30
3	Kanker3	Kanker	62%	30
4	Kanker4	Kanker	62%	26
5	Kanker5	Kanker	62%	30
6	Normal1	Normal	62%	30
7	Normal2	Normal	60%	30
8	Normal3	Normal	40%	30
9	Normal4	Normal	50%	26
10	Normal5	Normal	62%	30
Rata-rata			58.4%	29.2

Pengujian untuk kombinasi parameter III terlihat pada tabel 6

Tabel 6 Hasil Pengujian Kombinasi Parameter III

No	Nama Gambar	Dikenali oleh sistem	Akurasi	Waktu (ms)
1	Kanker1	Kanker	60%	28
2	Kanker2	Kanker	40%	30
3	Kanker3	Kanker	50%	26
4	Kanker4	Kanker	62%	30
5	Kanker5	Kanker	62%	29
6	Normal1	Normal	62%	30
7	Normal2	Normal	38%	30
8	Normal3	Normal	45%	30
9	Normal4	Normal	64%	30
10	Normal5	Normal	70%	30
Rata-rata			55.3%	29.3

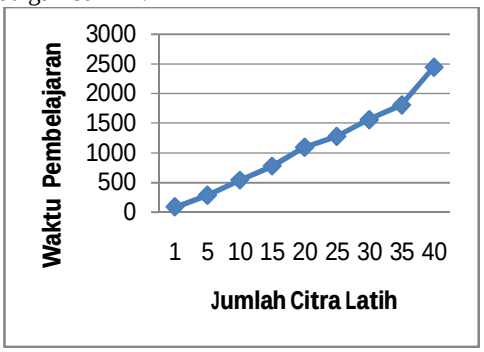
Kesimpulan hasil pengujian performansi didapatkan tingkat akurasi rata-rata dari ketiga kombinasi parameter sebesar 57% dan waktu pengenalan rata-rata sebesar 29.5 *milisecond.*

Pengujian waktu pembelajaran bertujuan untuk menyamakan hasil analisis kompleksitas waktu asimptotik dengan implementasinya, pada pengujian ini menggunakan parameter pembelajaran yaitu maksimal perulangan=100, rasio pembelajaran=0.5, dan minimal error=0.01 . Pengujian waktu pembelajaran terlihat seperti tabel 7

Tabel 7 Pengujian Waktu Pembelajaran Terhadap Jumlah Citra Latih

No	Jumlah Citra Latih	Waktu Pembelajaran (ms)
1	1 citra latih	88
2	5 citra latih	286
3	10 citra latih	541
4	15 citra latih	779
5	20 citra latih	1097
6	25 citra latih	1280
7	30 citra latih	1560
8	35 citra latih	1808
9	40 citra latih	2445

Grafik pengujian performansi berdasarkan waktu pembelajaran terhadap jumlah citra latih terlihat pada gambar 14.



Gambar 14. Grafik Waktu Pembelajaran Terhadap Jumlah Data Latih

Berdasarkan pengujian proses pembelajaran dapat ditarik sebuah kesimpulan semakin sedikit citra latih maka waktu yang digunakan juga sedikit, dan semakin banyak citra latih maka waktu yang digunakan juga semakin lama.

3. PENUTUP

Berdasarkan hasil yang didapat dalam penelitian dan penyusunan skripsi ini serta disesuaikan dengan tujuannya, maka diperoleh kesimpulan, tingkat akurasi rata-rata 57% artinya tingkat akurasi yang dihasilkan pada penelitian ini masih kurang baik, dan rata-rata waktu pengenalan 30 *milisecond* dengan menggunakan 10 data uji dengan 5 data normal dan 5 data kanker, hal tersebut terjadi karena arsitektur jaringan syaraf tiruan yang digunakan dan inisialisasi bobot-bobot awal yang digunakan belum cukup baik . Performansi yang dihasilkan pada penelitian ini cukup, tetapi ada ketidaksinkronan antara hasil akurasi dan performansi, hal tersebut terjadi akibat perbedaan arsitektur *compiler* pada saat implementasi, dan perbedaan implementasi antara struktur data dan arsitektur jaringan syaraf tiruan yang digunakan.

Pada masa yang akan datang pengenalan sel kanker otak dapat dikembangkan lebih lanjut, adapun saran untuk dimasa mendatang adalah dengan menggunakan arsitektur jaringan syaraf tiruan dan inisialisasi nilai bobot-bobot awal yang lebih baik. Pembuatan struktur data dan algoritma yang lebih baik agar hasil yang penelitian yang didapatkan agar lebih mangkus.

DAFTAR PUSTAKA

[1] Argisraha Satria C, 2012, *Indonesian Journal Of Tropical and Infectious Deseas, Digital Detection System Design Of Mycrobacterium Tuberculosis Trough Axtraction Of Sputum Image Using Neural Network* , No.1, Vol 3.

[2] Zamani Mizza A, 2012, Teknik Pomits, Implementasi Algortima Genetika pada Struktur *Backpropagation Neural Network* untuk Klasifikasi Kanker Payudara, No.1, Vol 1.

[3] Arif Riantini S, ITT Telkom, Deteksi Tumor Otak Berdasarkan Citra *Magnetic Resonance Imaging* (MRI) Berbasis Jaringan Syaraf Tiuan *Radial Basis Function* (RBF).

[4] Kusumadewi, 2003 , *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu : Yogyakarta.

[5] Putra Darma, 2010, *Pengolahan Citra Digital*, ANDI : Yogyakarta.

[6] Adam Mukharil Bachtiar, Diktat Perkuliahan Analisis Algoritma, Universitas Komputer Indonesia, Bandung.

[7] Tom M.Mitchell, 1997, *Machine Learning*, McGraw-Hill Science.