

Klasifikasi Status Gizi Balita Menggunakan Metode Backpropagation Dengan Algoritma Levenberg-Marquardt dan Inisialisasi Nguyen Widrow

Wildan Azka Adzani*¹⁾, Priyo Sidik Sasongko *²⁾

*Departemen Ilmu Komputer/Informatika, Fakultas Sains Matematika, Universitas Diponegoro

¹⁾wildan.azka.adzani@if,undip.ac.id, ²⁾priyoss_undip@yahoo.co.id

Abstrak

Banyaknya Kasus Gizi Buruk pada anak-anak usia di bawah lima tahun umumnya ditemukan akibat ketidaktahuan terhadap permasalahan gizi pada anak. Klasifikasi Status Gizi Balita merupakan upaya yang dilakukan untuk mengetahui status gizi pada balita. Penelitian ini bertujuan untuk membangun sebuah sistem Klasifikasi status gizi balita berbasis web menggunakan jaringan syaraf tiruan Backpropagation dengan Algoritma Levenberg-Marquardt dan inisialisasi Nguyen-Widrow. Variabel yang digunakan dalam penelitian ini merupakan data antropometri sebanyak 4 variabel. Seluruh data penelitian diambil dari POSYANDU RW 08 Kelurahan Sambiroto Kecamatan Tembalang, Semarang, Jawa Tengah. Data yang diambil sebanyak 100 data dengan pembagian data latih dan data uji menggunakan K-Fold Cross Validation. Hasil penelitian menunjukkan arsitektur terbaik untuk melakukan Klasifikasi didapat pada kombinasi parameter hidden neuron 12, parameter Levenberg-Marquardt (μ) 0.01, maksimum epoch 1000 dan target error 0.001 yang menghasilkan MSE 0.000064

Kata kunci : Sistem Klasifikasi Status Gizi Balita, K-Fold Cross Validation, Jaringan Syaraf Tiruan, Backpropagation, Levenberg-Marquardt, Nguyen-Widrow

Abstract

The number of malnutrition cases in children under the age of five is generally found due to ignorance of nutritional problems in children. Classification of Nutritional Status of Toddlers is an effort made to determine the nutritional status of infants. This research aims to develop an Classification system of Toddler Nutrition Status using artificial neural network Backpropagation with Levenberg-Marquardt algorithm and Nguyen Widrow Inilization. The variables used in this research is the anthropometry data as many as 4 variables. All data were taken from POSYANDU RW 08 Kelurahan Sambiroto Kecamatan Tembalang, Semarang, Jawa Tengah. The data are taken as much as 100 data with the distribution of training data and test data used K -fold Cross Validation. The results showed the best architecture for the detection obtained in combination parameter 12 hidden neuron, Levenberg-Marquardt parameter (μ) of 0.01, the maximum epoch of 1000 , the target error of 0.001 which resulted in MSE 0.000064.

Keywords : Classification System of Toddler Nutrition Status, K-Fold Cross Validation, Artificial Neural Network, Backpropagation, Levenberg-Marquardt Algorithm, Nguyen-Widrow

1 PENDAHULUAN

Kesehatan merupakan salah satu bidang yang sangat penting dalam kehidupan manusia. Salah satu cabang dari bidang kesehatan adalah gizi. Gizi merupakan komponen kimia yang terdapat dalam zat makanan yang sangat dibutuhkan oleh tubuh untuk perkembangan dan pertumbuhan. Status gizi dapat menjadi ukuran keberhasilan dalam pemenuhan nutrisi untuk anak yang diindikasikan oleh berat badan dan tinggi badan anak. Status gizi juga didefinisikan sebagai status kesehatan [1]. Status gizi balita menjadi suatu indikator yang mencerminkan tingkat kesejahteraan suatu masyarakat.

Untuk mendapatkan suatu hasil klasifikasi status gizi balita yang cepat dan akurat, diperlukan suatu metode yang tepat. Sekarang ini sudah terdapat beberapa penelitian tentang status gizi dengan menggunakan metode tertentu, diantaranya yaitu menggunakan metode Perceptron dengan nilai ketepatan sebesar 82,609% [2]. Namun metode Perceptron ini masih kurang dalam hal ketepatan karena terdapat penelitian tentang status gizi dengan menggunakan Backpropagation yang memiliki nilai ketepatan lebih tinggi yaitu sebesar 95% [3].

Backpropagation terbukti memiliki tingkat akurasi yang sangat baik dalam melakukan klasifikasi. Namun, masalah utama yang dihadapi dalam Backpropagation adalah lamanya iterasi yang harus dilakukan [4]. Permasalahan tersebut bisa diatasi dengan menggunakan algoritma Levenberg Marquardt yang merupakan pengembangan dari algoritma Backpropagation. Hal ini dibuktikan dengan adanya penelitian tentang perbandingan algoritma Backpropagation dengan Levenberg Marquardt. Penelitian tersebut

menghasilkan tingkat error terkecil sebesar 1×10^{-4} pada iterasi ke 480 untuk algoritma Backpropagation. Sedangkan dengan menggunakan algoritma Levenberg Marquardt, dapat menghasilkan error yang lebih kecil dan iterasi lebih cepat yaitu sebesar 1×10^{-6} pada iterasi ke 35 [5]. Penelitian lain yang menggunakan algoritma Levenberg Marquardt adalah klasifikasi harga emas [6] dengan tingkat akurasi mencapai 99,7604% dan peramalan awal musim hujan [7] dengan tingkat akurasi sebesar 99%. Untuk mengurangi waktu pelatihan dapat diatasi dengan algoritma Nguyen Widrow [8]. Algoritma Nguyen Widrow ini merupakan sebuah algoritma yang digunakan untuk inialisasi bobot pada jaringan syaraf tiruan [4].

Artikel ini akan melakukan studi analisis klasifikasi status gizi balita menggunakan metode Backpropagation dengan algoritma Levenberg- Marquardt dan inialisasi Nguyen Widrow. Metode tersebut telah digunakan dalam beberapa penelitian dan dapat menghasilkan akurasi yang tinggi serta waktu komputasi yang cukup cepat.

2 TINJAUAN PUSTAKA

2.1 KLASIFIKASI STATUS GIZI BALITA

Dalam menentukan status gizi balita harus ada ukuran baku yang sering disebut *reference*. Pengukuran baku antropometri indeks ambang batas (z-score) yang sekarang digunakan di Indonesia mengacu pada Standar World Health Organization (WHO). Klasifikasi status gizi dapat dibedakan menjadi empat yaitu [9]:

1. Gizi lebih (*Over weight*)

Gizi lebih terjadi bila tubuh memperoleh zat-zat gizi dalam jumlah berlebihan sehingga menimbulkan efek toksis atau membahayakan [10]. Kelebihan berat badan pada balita terjadi karena ketidakmampuan antara energi yang masuk dengan keluar, terlalu banyak makan, terlalu sedikit olahraga atau keduanya. Kelebihan berat badan anak tidak boleh diturunkan, karena penyusutan berat akan sekaligus menghilangkan zat gizi yang diperlukan untuk pertumbuhan. Untuk penentuan status gizi lebih, masuk pada kategori indeks ambang batas (z-score) >2 SD.

2. Gizi baik (well nourished)

Status gizi baik atau status gizi optimal terjadi bila tubuh memperoleh cukup zat-zat gizi yang digunakan secara efisien sehingga memungkinkan pertumbuhan fisik, perkembangan otak, kemampuan kerja dan kesehatan secara umum pada tingkat setinggi mungkin [10]. Untuk penentuan status gizi baik, masuk pada kategori indeks ambang batas (z-score) -2 SD sampai dengan 2 SD.

3. Gizi kurang (under weight)

Status gizi kurang terjadi bila tubuh mengalami kekurangan satu atau lebih zat-zat esensial [10]. Untuk penentuan status gizi kurang, masuk pada kategori indeks ambang batas (z-score) -3 SD sampai dengan <-2 SD.

4. Gizi buruk (severe PCM)

Gizi buruk adalah suatu kondisi di mana seseorang dinyatakan kekurangan nutrisi, atau dengan ungkapan lain status nutrisinya berada di bawah standar rata-rata. Nutrisi yang dimaksud bisa berupa protein, karbohidrat, dan kalori. Di Indonesia, kasus KEP (Kurang Energi Protein) adalah salah satu masalah gizi utama yang banyak dijumpai pada balita [11]. Untuk penentuan status gizi buruk,

masuk pada kategori indeks ambang batas (z-score) < -3 SD.

2.2 METODE BACKPROPAGATION

Backpropagation merupakan salah satu model jaringan syaraf tiruan. Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh Perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya [12]. Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa tapi tidak sama dengan pola yang dipakai selama pelatihan [4].

2.3 ALGORITMA LEVENBERG-MARQUARDT

Algoritma Levenberg-Marquardt merupakan pengembangan dari algoritma error Backpropagation. Algoritma ini dibangun untuk mengatasi beberapa kekurangan yang ada pada algoritma error Backpropagation dengan memanfaatkan teknik optimasi numerik standar yaitu menggunakan pendekatan matriks Jacobian.

Tujuan dari Levenberg-Marquardt adalah meminimalkan total error. Berikut beberapa struktur algoritma Levenberg-Marquardt :

1. Inisialisasi bobot dan bias dengan bilangan acak, epoch maksimum, minimal goal (*performance error*)
2. Menentukan parameter yang dibutuhkan, antara lain:

- a. Parameter *Levenberg-Marquardt* yang nilainya harus lebih besar dari nol
 - b. Parameter faktor input dan bias yang digunakan sebagai parameter yang dikalikan atau dibagi dengan parameter *Levenberg-Marquardt*
3. Menghitung maju (feedforward) pada hidden dan output layer seperti langkah-langkah pada algoritma error backpropagation. Pada langkah (1)- (4).
 4. Menghitung nilai error
 5. Menghitung error dan total error jaringan

- a. Rumus untuk error :

$$e_r = t_r - y_r \quad (1)$$

r merupakan input ke-r

- b. Rumus untuk menghitung total error :

$$e = [e_1 \ e_2 \ e_3 \ \dots \ e_N]^T \quad (2)$$

e merupakan vektor kesalahan berukuran Nx1 yang terdiri dari er
r = 1,2,3...N dimana N menunjukkan indeks pola data.

6. Menghitung matriks Jacobian J(x)
x merupakan matriks yang berisi nilai bobot dan bias dari keseluruhan jaringan.

$$X = [v_{11}, v_{12}, \dots, v_{ij}; w_{01}, w_{02}, \dots; w_{11}, w_{12}, \dots, w_{jk}; w_{01}, w_{02} \dots w_{0k}]$$

Matriks Jacobian berisi turunan pertama error jaringan terhadap bobot dan bias jaringan. Rumus untuk mencari Jacobian Matriks antara lain

$$J(x) = [\partial e_r / \partial w] \quad (3)$$

7. Setelah didapatkan nilai J(x) maka dapat dihitung perubahan koreksi bobot dan biasnya dengan rumus berikut:

$$\Delta x = [J(x)^T J(x) + \mu I]^{-1} * Gradient Gradient:$$

$$J(x)^T e \quad (4)$$

8. Setelah didapatkan nilai Δx tahap selanjutnya adalah pengkoreksian bobot dengan rumus yang sama seperti pada algoritma error backpropagation. Pada langkah 12
9. Menghitung maju (feedforward) dengan bobot dan bias yang baru. Pada langkah (1-4)
10. Menghitung error jaringan dengan bobot dan bias yang baru. Kemudian tes kondisi berhenti
11. Jika epoch atau iterasi masih berlanjut maka akan terdapat 2 kemungkinan:
 - a. error naik
 - b. error turun
12. Kemudian lakukan kembali langkah 5 sampai dengan 8 (Kusumadewi, 2004).

Dimana:

er = error ke- r

tr = target ke- r

yr = output ke- r

r = input ke- r

J(x) = matriks Jacobian

2.4 ALGORITMA INISIASI NGUYEN WIDROW

Nguyen Widrow adalah sebuah algoritma yang digunakan untuk inisialisasi bobot pada jaringan saraf

tiruan untuk mengurangi waktu pelatihan. Algoritma inisialisasi Nguyen Widrow adalah sebagai berikut [4].

1. Tentukan:
 - n = jumlah unit *input*
 - p = jumlah unit tersembunyi
 - β = faktor skala = $0.7 (p)^{1/n}$ (5)
2. Untuk setiap unit tersembunyi ($j=1,2,\dots,p$), lakukan tahap 3 – 6
3. Inisialisasi semua bobot v_{ij} (lama)= bilangan acak dalam interval [-0.5 dan 0.5]

4. Hitung nilai:

$$\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + v_{j3}^2 + \dots + v_{jn}^2} \quad (6)$$

5. Bobot yang dipakai sebagai inisiasi

$$v_{ji} = \beta v_{ji}(\text{lama}) / \|v_j\| \quad (7)$$

6. Bias yang dipakai sebagai inisialisasi:

$$v_{j0} = \text{bilangan acak antara } -\beta \text{ dan } \beta \quad (8)$$

Metode Nguyen Widrow akan menginisialisasi bobot-bobot lapisan dengan nilai antara -0.5 sampai 0.5. Sedangkan bobot-bobot dari lapisan input ke lapisan tersembunyi di rancang sedemikian rupa sehingga dapat meningkatkan kemampuan lapisan tersembunyi dalam melakukan proses pembelajaran [12]

2.5 K-FOLD CROSS VALIDATION

K-Fold Cross Validation merupakan salah satu metode yang umumnya digunakan untuk mengevaluasi kinerja classifier. Metode *K-Fold Cross Validation* dilakukan dengan membagi dataset secara acak menjadi k himpunan bagian (subset) [13].

3 METODE PENELITIAN

3.1 PENGUMPULAN DATA

Pengumpulan data yang digunakan dalam penelitian, dilakukan pada tanggal 28 Juli 2016 sampai 30 Juli 2016 di POSYANDU RW 08 Kelurahan Sambiroto Kecamatan Tembalang, Semarang, Jawa Tengah. Data yang diperoleh merupakan data antropometri dan status gizi balita pada tahun 2015 dengan mengambil 100 data. Contoh data hasil pengolahan antropometri balita pada tahap pengumpulan data ditunjukkan pada Tabel 1

Tabel 1 Data yang Dikumpulkan

Id	Jenis Kelamin	Umur (Bulan)	Berat Badan (kg)	Tinggi Badan (cm)	Status Gizi
1	L	14	9.00	74.00	Baik
2	L	51	20.00	105.00	Baik
3	P	42	14.50	98.00	Baik
4	L	16	9.00	77.00	Baik
5	P	55	15.00	105.00	Baik
...
96	L	2	7.20	57.00	Lebih
97	P	39	11.00	92.00	Kurang
98	P	48	12.00	99.00	Kurang
99	P	40	8.20	67.00	Buruk
100	L	52	11.00	95.00	Buruk

3.2 NORMALISASI DATA

Normalisasi data bertujuan untuk mengubah nilai data gejala ke dalam interval [0,1] sehingga interval data input sesuai dengan interval fungsi aktivasi yang digunakan yaitu fungsi aktivasi sigmoid biner. Pada penelitian ini, data numerik ditransformasi ke dalam range 0 sampai 1 sesuai dengan fungsi aktivasi yang digunakan yaitu sigmoid biner. Normalisasi untuk data numerik menggunakan persamaan sebagai berikut [14].

$$data' = \frac{data - \min}{\max - \min} * (a - b) + b \quad (9)$$

data' : Nilai normalisasi

data : Nilai data

min : Nilai minimum data

max : Nilai maksimum data

a : Batas atas yang ditentukan (1)

b : Batas bawah yang ditentukan (0)

Proses normalisasi untuk variabel x1, x2, x3, x4, dan t dilakukan dari data ke- 1 sampai data ke-100. Data yang telah dinormalisasi selanjutnya digunakan pada strategi pembentukan data latih dan data uji. Tabel 2 merupakan data yang sudah dinormalisasi.

Tabel 2 Data Anthropometri Normal

Id	x	x2	x3	x4	t
1	0	0.2476	0.2469	0.3666	0.7
2	0	0.7030	0.6959	0.8017	0.7
3	1	0.5923	0.4714	0.7035	0.7
4	0	0.2723	0.2469	0.4087	0.7
5	1	0.7523	0.4918	0.8017	0.7
...
96	0	0.1000	0.1734	0.1280	1
97	1	0.5553	0.3285	0.6193	0.2
98	1	0.6661	0.3693	0.7175	0.2
99	1	0.5676	0.2142	0.2684	0
10	0	0.7153	0.3285	0.6614	0

Tabel 3. Data Latih

Id	x1	x2	x3	x4	t
6	1	0.22	0.32	0.39	0.75
7	0	0.22	0.25	0.34	0.75
8	0	0.52	0.45	0.66	0.75
...
50	0	0.53	0.19	0.24	0.75
56	0	0.30	0.29	0.48	0.75
57	0	0.20	0.23	0.37	0.75
...
98	1	0.67	0.37	0.72	0.25
99	1	0.57	0.21	0.27	0
100	0	0.72	0.33	0.66	0

Tabel 4. Data Uji

Id	x1	x2	x3	x4	t
1	0	0.25	0.25	0.37	0.75
2	0	0.70	0.70	0.80	0.75
3	1	0.59	0.47	0.70	0.75
4	0	0.27	0.25	0.41	0.75
5	1	0.75	0.49	0.80	0.75
51	0	0.67	0.70	0.84	0.75
52	1	0.65	0.70	0.80	0.75
53	1	0.33	0.29	0.44	0.75
54	0	0.36	0.37	0.52	0.75
55	0	0.32	0.37	0.44	0.75

3.3 PEMBAGIAN DATA LATIH DAN DATA UJI

Tahapan ini dilakukan untuk membagi dataset yang telah dinormalisasi menjadi data latih dan data uji. Penelitian ini menggunakan K-Fold Cross Validation dengan nilai K=10 untuk membagi data latih dan data uji. Dataset akan dibagi menjadi 10 partisi dengan tiap partisi terdiri dari jumlah data yang sama untuk setiap kelasnya. Dataset berjumlah 100 data dibagi menjadi 10 partisi sehingga tiap partisi berisi 10 data.

3.4 PROSES BACKPROPAGATION DENGAN ALGORITMA LEVENBERG-MARQUARDT

Perhitungan Backpropagation dengan algoritma Levenberg-Marquardt yaitu dengan jumlah neuron pada hidden layer sebanyak 3 pada fold pertama, dengan nilai μ (μ) 0.001, error target 0.001 dan epoch maksimum 1000. Karena dilakukan untuk fold pertama maka data pertama sampai dengan kelima dan data ke-51 sampai data ke-55 akan dijadikan data pengujian sedangkan yang lainnya dijadikan data pelatihan. Data latih pada Tabel 3 dan Tabel 4 merupakan data uji

Langkah 0 : Memberikan inisialisasi bobot-bobot awal dengan menggunakan variasi *Nguyen Widrow*

1. Inisialisasi bobot menggunakan *Nguyen Widrow* Dengan n jumlah unit masukan adalah 4, p jumlah unit tersembunyi adalah 3, dapat dihitung factor skala β adalah

$$\beta = 0.7(p)^{1/n} = 0.7(3)^{1/4} = 0.92$$

Untuk semua unit masukan, inisialisasi bobot acak dalam interval [-0.5, 0.5]

Tabel 5. Inisiasi Bobot

	z1	z2	z3
x1	0.12	0.33	0.24
x2	0.35	-0.27	0.17
x3	0.21	0.43	-0.11
x4	0.37	0.17	0.23

2. Hitung

$$\begin{aligned} \|v1\| &= \sqrt{v112+v122+v132+v142} \\ &= \sqrt{0.122+0.352+0.212+0.372} \\ &= \sqrt{0.01+0.12+0.04+0.14} \\ &= \sqrt{0.31} \\ &= 0.56 \\ \|v2\| &= 0.63 \\ \|v3\| &= 0.39 \end{aligned}$$

3. Menghitung bobot yang dipakai sebagai inisiasi

$$\begin{aligned} v_{ji} &= \beta v_{ji}(lama)/\|v_j\| \\ v_{11} &= \beta v_{11}(lama)/\|v_1\| \\ &= 0.92 \times 0.12/\|0.56\| \\ &= 0.110.56=0.20 \end{aligned}$$

Berdasarkan contoh perhitungan bobot yang dipakai diatas untuk hasil keseluruhan dari bobot yang dipakai sebagai inisialisasi dapat ditunjukkan pada Tabel 6 yang merupakan hasil perhitungan bobot inisialisasi dengan menggunakan Nguyen Widrow.

Tabel 6. Bobot baru dengan nguyen widrow

	z1	z2	z3
x1	0.20	0.48	0.57
x2	0.58	-0.39	0.40
x3	0.35	0.63	-0.26
x4	0.61	0.25	0.54

4. Mengambil bilangan bias yang dipakai sebagai inisialisasi = v_{j0} = bilangan acak antara $-\beta$ dan β . $\beta = 0.92$, sehingga inisialisasi bilangan bias secara acak dalam range -0.92 sampai dengan 0.92 dapat disajikan pada Tabel 7.

Tabel 7. Inisialisasi bilangan bias dengan Nguyen Widrow

	z1	z2	z3
x0	-0.27	0.13	0.41

5. Mengambil bilangan acak kecil secara random (range [-1, 1]) untuk menentukan bobot dari layar tersembunyi ke layar keluaran = w_{kj}

Tabel 8 Tabel Bobot dari Unit Tersembunyi

	y
z0	-0.10
z1	0.21
z2	-0.11
z3	-0.19

Tentukan :

n = jumlah input = 4

p = jumlah neuron pada hidden layer = 3

B =faktor pembagi dan pengali parameter

μ = 10

μ = parameter Levenberg-Marquardt = 0.001

error target = 0.001

maxepoch = 1000

Langkah 1 : Jika kondisi penghentian (epoch \geq max epoch dan MSE < error target), lakukan langkah selanjutnya.

Langkah 2 : Untuk setiap data latih lakukan langkah selanjutnya.

Langkah 3 : Tiap masukan menerima sinyal dan meneruskannya ke unit tersembunyi.

Langkah 4 : Hitung semua keluaran jaringan di unit tersembunyi z_j ($j=1,2,\dots,p$).

Sebelum menghitung keluaran jaringan di unit tersembunyi z_j , hitung z_{netj}

Untuk data dengan id data 6 (data latih pertama) didapatkan nilai z_{netj} sebagai berikut

$$\begin{aligned} z_{net1} &= v_{10} + \sum_{i=1}^n x_{iv1i} \\ &= -0.27 + (1*0.20) + (0.22*0.58) + (0.32*0.35) + (0.39*0.61) \end{aligned}$$

$$= 0.41$$

$$z_{net2} = v_{20} + \sum_{10} x_{iv2i}$$

$$= 0.13 + (1*0.48) + (0.22*-0.39) + (0.32*0.63) + (0.39*0.25)$$

$$= 0.82$$

$$z_{net3} = v_{30} + \sum_{10} x_{iv3i}$$

$$= 0.41 + (1*0.57) + (0.22*0.40) + (0.32*-0.26) + (0.39*0.54)$$

$$= 1.20$$

Perhitungan z_{netj} dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$Z_{net} =$

	z_{net1}	z_{net2}	z_{net3}
data6	0.41	0.82	1.20
data7	0.19	1.14	1.67
.	.	.	.
data50	0.12	-0.01	0.44
data56	0.86	1.43	0.14
.	.	.	.
data99	-0.56	-0.34	-0.84
data100	1.15	0.70	-0.42

Setelah menghitung z_{netj} maka dilanjutkan menghitung semua keluaran jaringan di unit tersembunyi z_j

$$z_1 = 1/1 + e^{-z_{net1}} = 1/1 + e^{-(0.41)} = 0.6011$$

$$z_2 = 1/1 + e^{-z_{net2}} = 1/1 + e^{-(0.82)} = 0.6942$$

$$z_3 = 1/1 + e^{-z_{net3}} = 1/1 + e^{-(1.20)} = 0.7685$$

Perhitungan untuk menghitung keluaran jaringan di unit tersembunyi dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

	z_1	z_2	z_3
data6	0.6011	0.6942	0.7685
data7	0.5473	0.7577	0.8416
.	.	.	.
data50	0.5299	0.4975	0.6083
data56	0.3635	0.4158	0.3015
.	.	.	.

data99	0.8278	0.6682	0.3965
data100	0.7027	0.8069	0.5349

Langkah 5 : Menghitung semua keluaran jaringan di unit y

Sebelum menghitung keluaran jaringan di unit y, hitung nilai y_{net} untuk data dengan id data 6 (data latih pertama didapatkan y_{net} sebagai berikut

$$y_{net} = w_0 + \sum z_j w_j$$

$$= -0.10 + (0.6011*0.21) + (0.6942*-0.11) + (0.7685*-0.19)$$

$$= -0.1961$$

Perhitungan dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$y_{net} =$

data6	-0.1961
data7	-0.3281
.	.
data50	-0.0894
data56	-
	0.00049
.	.
data99	-0.3119
data100	-0.3157

Setelah menghitung y_{net} maka dilanjutkan menghitung semua keluaran jaringan di unit y sebagai berikut

$$y_1 = 1/1 + e^{-y_{net1}} = 1/1 + e^{-(-0.1961)} = 0.4511$$

Perhitungan dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$y =$

data6	0.4511
data7	0.4187
.	.
data50	0.4776
data56	0.4999
.	.
data99	0.4227
data100	0.4217

Langkah 6 : Menghitung nilai *Mean Square Error (MSE)*

$$MSE = 1/2a \sum (y_l - t_l)^2$$

$$= 0.12572$$

Langkah 7 : Menghitung matriks Jacobian

Langkah 8 : Menghitung perubahan bobot.

$$W_{k+1} = W_k - [J'J + \mu I]^{-1} J' e$$

Didapatkan hasil bobot baru sebagai berikut

$v_{ji} =$

	x_0	x_1	x_2	x_3	x_4
z_1	1.066	-10.473	-0.938	-0.298	2.049
z_2	2.298	-6.136	-1.329	-1.391	1.613
z_3	0.479	-12.611	0.115	-1.666	2.916

$w_j =$

	z_0	z_1	z_2	z_3
w_j	3.876	-10.872	-0.126	-6.496

Langkah 9 : Melakukan prosedur hitung maju menggunakan bobot baru. Menghitung semua keluaran jaringan di unit tersembunyi z_j ($j=1,2,\dots,p$). Sebelum menghitung keluaran jaringan di unit tersembunyi z_j , hitung z_{netj}

Untuk data dengan id data 6 (data latih pertama) didapatkan nilai keluaran jaringan di unit tersembunyi sebagai berikut

$$z_{net1} = -11.761$$

$$z_{net2} = -3.142$$

$$z_{net3} = -8.465$$

Perhitungan z_{netj} dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$z_{net} =$

	z_{net1}	z_{net2}	z_{net3}
data6	-11.761	-3.142	-8.465
data7	-12.350	-3.342	-9.029
.	.	.	.
data50	-12.943	-7.908	-13.759
data56	2.176	2.581	3.510
.	.	.	.
data99	1.539	3.817	4.620

data100	3.778	7.166	7.254
---------	-------	-------	-------

Setelah menghitung z_{netj} maka dilanjutkan menghitung semua keluaran jaringan di unit tersembunyi z_j

$$z_1 = 1/1 + e^{-z_{net1}} = 1/1 + e^{-(-11.761)} = 7.8 \times 10^{-6}$$

$$z_2 = 1/1 + e^{-z_{net2}} = 1/1 + e^{-(-0.8142)} = 0.0413$$

$$z_3 = 1/1 + e^{-z_{net3}} = 1/1 + e^{-(-8.465)} = 0.0002$$

Perhitungan untuk menghitung keluaran jaringan di unit tersembunyi dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

	z_1	z_2	z_3
data6	7.8×10^{-6}	0.0413	0.0002
data7	4.3×10^{-6}	0.0342	0.0001
.	.	.	.
data50	2.4×10^{-6}	0.0004	1.1×10^{-6}
data56	0.08980	0.9296	0.9709
.	.	.	.
data99	0.8233	0.9785	0.9902
data100	0.9777	0.9992	0.9993

Menghitung semua keluaran jaringan di unit y . Sebelum menghitung keluaran jaringan di unit y , hitung nilai y_{net}

Untuk data dengan id data 6 (data latih pertama) didapatkan y_{net} sebagai berikut

$$y_{net} = w_0 + \sum z_j w_j$$

$$= 4.1555$$

Perhitungan dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$y_{net} =$

data6	4.1555
data7	4.1138
.	.
data50	3.876
data56	-7.1972
.	.
data99	-6.1407
data100	-8.0446

Setelah menghitung y_{net} maka dilanjutkan menghitung semua keluaran jaringan di unit y sebagai berikut

$$y_i = 1/1 + e^{-y_{neti}} = 0.9846$$

Perhitungan dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$$y =$$

data6	0.9846
data7	0.9839
data50	0.9797
data56	0.0007
data99	0.0021
data100	0.0003

Langkah 10 : Menghitung nilai *Mean Square Error* (MSE) trial.
 $MSE\ trial = 1/2a\Sigma(y_l - t_l)$
 $= 0.03225$

Nilai MSE trial lebih kecil dari MSE sehingga nilai $\mu=0.001/10$, dan akan dilanjutkan untuk *epoch* selanjutnya. Pelatihan akan berhenti ketika salah satu kondisi berhenti tercapai. Pelatihan ini berhenti dan menghasilkan bobot akhir pelatihan.

$$v_{ji} =$$

	x_0	x_1	x_2	x_3	x_4
z_1	2.205	-5.298	0.188	0.704	0.124
z_2	1.132	-6.655	-2.485	-2.111	1.248
z_3	0.183	2.716	0.595	2.165	-0.543

$$w_j =$$

	z_0	z_1	z_2	z_3
w_j	2.654	-8.494	0.501	1.714

3.5 PENGUJIAN DAN EVALUASI

Perhitungan dari proses pengujian dan evaluasi dengan jumlah *neuron* pada *hidden layer* sebanyak 5 pada *fold* pertama, dengan bobot hasil pelatihan diperoleh sebagai berikut

$$v_{ji} =$$

	x_0	x_1	x_2	x_3	x_4
z_1	2.205	-5.298	0.188	0.704	0.124
z_2	1.132	-6.655	-2.485	-2.111	1.248
z_3	0.183	2.716	0.595	2.165	-0.543

$$w_j =$$

	z_0	z_1	z_2	z_3
w_j	2.654	-8.494	0.501	1.714

Langkah 1 : Tiap masukan menerima sinyal dan meneruskannya ke unit tersembunyi.

Langkah 2 : Menghitung semua keluaran jaringan di unit tersembunyi z_j ($j=1,2,\dots,p$). Sebelum menghitung keluaran jaringan di unit tersembunyi z_j maka terlebih dahulu menghitung z_{netj}

Untuk data ke-1 didapatkan nilai keluaran jaringan di unit tersembunyi sebagai berikut

$$z_{net1} = -4.56$$

$$z_{net2} = -6.828$$

$$z_{net3} = 4.236$$

Perhitungan z_{netj} dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$$Z_{net} =$$

	z_{net1}	z_{net2}	z_{net3}
data1	-4.56	-6.828	4.238
data2	-2.4855	-14.8844	9.9850
data3	-1.8659	-12.3748	6.3217
data4	-3.3685	10.9811	7.6070
data5	2.8188	1.8164	0.8447
data51	2.9093	-0.9791	2.3490
data52	1.539	3.817	4.620
data53	3.778	7.166	7.254
data54	3.0977	-3.4649	2.9443
data55	2.3296	2.3812	-0.3592

Setelah menghitung z_{netj} maka dilanjutkan menghitung semua keluaran jaringan di unit tersembunyi z_j

$$z_1 = 1/1 + e^{-z_{net1}} = 0.0103$$

$$z_2 = 1/1 + e^{-z_{net2}} = 0.0011$$

$$z_3 = 1/1 + e^{-z_{net3}} = 0.9857$$

Perhitungan untuk menghitung keluaran jaringan di unit tersembunyi dilanjutkan sampai data terakhir dari data

latih sehingga didapatkan hasil sebagai berikut

$z =$

	z_{net1}	z_{net2}	z_{net3}
data1	0.0103	0.0011	0.9857
data2	0.0768	3.433×10^{-7}	0.9999
data3	0.1340	4.22×10^{-6}	0.9982
data4	0.0332	0.9999	0.9995
data5	0.9436	0.8601	0.6994
data51	0.9483	0.2730	0.9128
data52	0.9567	0.0303	0.9499
data53	0.9112	0.9153	0.4111
data54	0.75844	0.7422	0.4486
data55	0.6637	0.1160	0.6458

Langkah 3 : Menghitung semua keluaran jaringan di unit y. Sebelum menghitung keluaran jaringan di unit y, hitung nilai y_{net}

Untuk data ke-1 didapatkan y_{net} sebagai berikut

$$y_{net} = W_0 + \sum z_j w_j = 5.3428$$

Perhitungan dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$y_{net} =$

data1	5.3428
data2	4.7079
data3	4.2255
data4	5.7009
data5	-3.5501
data51	-3.1221
data52	-2.6890
data53	-4.1389
data54	-2.3142
data55	3.7484

Setelah menghitung y_{net} maka dilanjutkan menghitung semua keluaran jaringan di unit y

$$\begin{aligned} y1 &= 1/1 + e^{-y_{net1}} \\ &= 1/1 + e^{-(5.3428)} \\ &= 0.9952 \end{aligned}$$

Perhitungan dilanjutkan sampai data terakhir dari data latih sehingga didapatkan hasil sebagai berikut

$y =$

data1	0.9952
data2	0.9910
data3	0.9855
data4	0.9966
data5	0.0279
data51	0.0422
data52	0.0636
data53	0.0157
data54	0.0314
data55	0.0748

Menghitung nilai *Mean Square Error* (MSE).

$$\begin{aligned} MSE &= 1/2a \sum (y_l - t_l)^2 \\ &= 0.0000338 \end{aligned}$$

3.6 KEBUTUHAN FUNSIONAL DAN NON FUNSIONAL

Analisis kebutuhan perangkat lunak dari sistem klasifikasi Status Gizi Balita ini

No. SRS – ID	Deskripsi
1 SRS- SKSGB-F-001	Memiliki otentikasi dan otorisasi untuk admin
2 SRS- SKSGB-F-002	Memiliki form ubah <i>password</i> admin
3 SRS-SKSGB-F-003	Memiliki form untuk tambah data penanganan
4 SRS-SKSGB-F-004	Memiliki fitur menampilkan data penanganan
5 SRS-SKSGB-F-005	Memiliki fitur hapus data penanganan
6 SRS-SKSGB-F-006	Memiliki fitur <i>import</i> dataset
7 SRS-SKSGB-F-007	Memiliki fitur hapus data antropometri
8 SRS-SKSGB-F-008	Memiliki fitur untuk menampilkan dataset
9 SRS-SKSGB-F-009	Memiliki fitur menampilkan data antropometri
10 SRS-SKSGB-F-010	Memiliki fitur pelatihan dan pengujian
11 SRS-SKSGB-F-011	Memiliki fitur pilih bobot final untuk klasifikasi
12 SRS-SKSGB-F-012	Memiliki fitur history pelatihan dan pengujian
13 SRS-SKSGB-F-013	Memiliki fitur hapus pelatihan dan pengujian
14 SRS-SKSGB-F-014	Memiliki fitur cek Status Gizi Balita

menghasilkan spesifikasi kebutuhan perangkat lunak atau yang disebut *Software Requirement Specification* (SRS) yang terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. Kode yang digunakan untuk merepresentasikan kebutuhan fungsional adalah SRS- SKSGB-F-xxx dan kode yang digunakan untuk merepresentasikan kebutuhan non fungsional adalah SRS- SKSGB -NF-xxx. Kebutuhan fungsional dapat dilihat pada Tabel 9 dan Kebutuhan non fungsional dapat dilihat pada Tabel 10.

Tabel 9 Kebutuhan Fungsional

Tabel 10 Kebutuhan Non Fungsional

No.SRS – ID	Deskripsi
1 SRS-SKSGB-NF- 001	Aplikasi dapat dijalankan menggunakan jaringan internet dan web browser yang mendukung HTML, javascript dan JQuery.

4 HASIL DAN PEMBAHASAN

4.1 IMPLEMENTASI ANTARMUKA

Implementasi antarmuka Sistem Klasifikasi Status Gizi Balita terdapat halaman antarmuka sebagai berikut.

4.1.1 HALAMAN UTAMA

Halaman utama merupakan halaman yang pertama ditampilkan ketika pengguna maupun admin memasuki sistem. Pengguna dapat memulai cek status gizi balita dengan cara memilih tombol “MULAI” pada halaman ini. Admin juga dapat langsung melakukan *Log in* untuk dapat memasuki halaman admin dengan cara memilih menu “*Log in*”. Tampilan halaman utama dapat dilihat pada Gambar 1.



Gambar 1. Antarmuka Halaman Utama

4.1.2 HALAMAN LIHAT DATA ANTROPOMETRI

Halaman lihat data antropometri menampilkan daftar antropometri yang telah dimasukkan. Halaman ini hanya dapat melihat seluruh data antropometri saja sehingga admin tidak dapat mengubah atau menghapus data antropometri. Halaman ini menyediakan menu untuk menghapus seluruh data antropometri yang telah tersimpan dalam DBMS MySQL. Tampilan Lihat Data Antropometri dapat dilihat pada Gambar 2.



Gambar 2. Antarmuka Lihat Data Antropometri

4.1.3 HALAMAN HISTORY PELATIHAN DAN PENGUJIAN

Halaman History Pelatihan dan Pengujian menampilkan daftar semua hasil pelatihan dan pengujian yang telah dilakukan. Pada tabel terdapat menu hapus data pelatihan dan pengujian baik per data maupun semua data. Admin juga dapat memilih pelatihan yang bobot hasilnya akan digunakan untuk proses klasifikasi.

Tampilan History Pelatihan dapat dilihat pada Gambar 3.

ID Pelatihan	hidden neuron	maxepoch	epoch	error target	mu	MSE Last	MSE Uj	Rata	Fold	Pakai
PL-005	6	1000	6	0.001	0.1	0.0009474203896677	0.007490312762009	100%	1	gagal
PL-006	6	1000	31	0.001	0.2	0.0009562034739768	0.007270629396503	100%	1	gagal
PL-007	6	1000	634	0.001	2	0.0009967980748564	0.0028484878148196	100%	1	gagal
PL-008	6	1000	624	0.001	3	0.0009874192477054	0.0041989167148916	100%	1	gagal
PL-009	6	1000	1000	0.001	3	0.0013000661379362	0.004272273198877	100%	1	gagal
PL-010	6	1000	940	0.001	3	0.0009994481222058	0.001209721439624	100%	1	gagal
PL-011	6	1000	306	0.001	1	0.000982019823631	0.00897421997145	100%	1	gagal
PL-012	6	1000	3	0.001	0.05	0.0008030093894848	0.0006000000000002	100%	1	gagal
PL-013	6	1000	1000	0.001	3	0.001102270591021	0.0009000000000000	100%	1	gagal
PL-016	6	1000	340	0.001	3	0.0009524670611902	0.0009154869371211	100%	4	gagal

Gambar 3. Antarmuka History Pelatihan dan Pengujian

4.2 PENGUJIAN

Pengujian Sistem Klasifikasi Status Gizi Balita dilakukan dengan metode *black box*, yaitu melakukan pengujian fungsionalitas dari perangkat lunak untuk menemukan kesalahan pada *requirement* dengan mengabaikan mekanisme internal atau komponen dari program tersebut. Pengujian *black box* diterima jika fitur-fitur dari perangkat lunak telah memenuhi kebutuhan sistem yang telah diidentifikasi sebelumnya. Berdasarkan hasil uji yang telah disajikan sebelumnya, dapat dilihat bahwa semua pengujian telah diterima.

Hasil yang diperoleh sesuai dengan kebutuhan fungsional sistem sehingga dapat disimpulkan bahwa Implementasi Jaringan Syaraf Tiruan *Backpropagation* dengan Algoritma Pelatihan *Levenberg-Marquardt* dan Inisialisasi Bobot awal *Nguyen Widrow* pada Sistem Klasifikasi Status Gizi Balita telah dibangun sesuai kebutuhan fungsional yang telah dibangun.

Eksperimen bertujuan memberikan hasil berupa arsitektur jaringan terbaik untuk studi kasus Klasifikasi Status Gizi Balita. Eksperimen menggunakan perbandingan kombinasi nilai parameter *Backpropagation*, *Levenberg-Marquardt*,

dan *Levenberg-Marquardt* dengan Inisialisasi *Nguyen Widrow* menggunakan (μ) 0.01-0.09 dan jumlah *hidden neuron* bernilai 3, 8, 10 dan 12. Parameter lain yaitu maksimal *epoch* 1000 dan target error 0.001. Proses pelatihan dilakukan berkali-kali dengan melakukan uji coba berdasarkan kombinasi parameter yang pada rencana eksperimen. Setiap pelatihan menghasilkan bobot akhir. Bobot akhir tersebut kemudian digunakan untuk melakukan proses pengujian. Proses pengujian menghasilkan nilai MSE untuk setiap foldnya dan diambil nilai rata-rata MSE untuk setiap kombinasi parameter. Hasil dari eksperimen dapat dilihat pada Tabel 11.

Tabel 11 Perbandingan Hasil Eksperimen Terbaik

HN	μ	E	MSE	MSE LM	MSE LM & NW
10	0.03	6	0.000924	0.000273	0.000217
10	0.06	10	0.000813	0.000554	0.000468
12	0.01	3	0.002048	0.000728	0.000064

Dari tabel 11 dapat diperoleh nilai MSE terendah yaitu 0.000064 dengan nilai parameter pada *Levenberg-Marquardt* dan Inisialisasi *Nguyen Widrow* menggunakan (μ) 0.01 dan 12 *hidden neuron*.

5 PENUTUP

5.1 KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian Sistem Klasifikasi Status Gizi Balita Berbasis Web Menggunakan Jaringan Syaraf Tiruan *Backpropagation* dengan Algoritma *Levenberg-Marquardt* adalah:

1. Arsitektur jaringan syaraf tiruan terbaik yang terbentuk pada penelitian ini adalah 4-12-1, yaitu 4 *neuron* pada *input layer*, 12 *neuron* pada *hidden layer*, dan 1 *neuron* pada *output layer* dengan nilai parameter pada *Levenberg-Marquardt* dan Inisialisasi *Nguyen Widrow*

menggunakan (μ) terbaik adalah 0.01 serta menghasilkan MSE sebesar 0.000064.

2. Sistem Klasifikasi Status Gizi Balita berbasis web telah dibangun menggunakan jaringan syaraf tiruan *Backpropagation* dengan algoritma *Levenberg-Marquardt* dan Inisialisasi *Nguyen Widrow* dapat melakukan Klasifikasi Status Gizi Balita melalui data antropometri.

5.2 SARAN

Saran yang dapat diberikan untuk penelitian selanjutnya adalah dengan menambah lebih banyak jumlah data, dan menambah keragaman dari data antropometri atau data status gizi balita

DAFTAR PUSTAKA

- [1] Sulistijani D A, Herlianty M P. 2003. Menjaga Kesehatan Bayi dan Balita. Jakarta : Puspa Swara.
- [2] Fitri, Setyawati, O. & Didik, R, S., 2013. Aplikasi Jaringan Syaraf Tiruan Untuk Penentuan Status Gizi Balita Dan Rekomendasi Menu Makanan Yang Dibutuhkan. Jurnal EECIS Vol. 7, No. 2 Desember.
- [3] Utami, O., Irawan, B., Setyaningsih, F, A., 2014. Diagnosa Gizi Pada Anak Balita Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation*. Jurnal Coding Sistem Komputer Universitas Tanjungpura Vol. 2, No 1, hal 33-41 ISSN : 2338-493x.
- [4] Siang, J. J., 2005. Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab. Yogyakarta: Penerbit Andi.
- [5] Rahmat, Setiawan, R., Purnomo, M, H., 2006. Perbandingan Algoritma *Levenberg-Marquardt* Dengan Metoda *Backpropagation* Pada Proses Learning Jaringan Saraf Tiruan Untuk Pengenalan Pola Sinyal Elektrokardiograf. Seminar Nasional Aplikasi Teknologi Informasi Yogyakarta ISSN : 1907-5022.
- [6] Hidayat, R, N., Isnanto, R, R., Nurhayati, O, D., 2013. Implementasi Jaringan Syaraf Tiruan Perambatan Balik Untuk MemKlasifikasi Harga Logam Mulia Emas Menggunakan Algoritma *Levenberg Marquardt*. Jurnal Teknologi dan Sistem Komputer Vol. 1, No. 2 April.
- [7] Buono, A., Kurniawan, A., Faqih, A., 2012. Peramalan Awal Musim Hujan Menggunakan Jaringan Syaraf Tiruan *Backpropagation Levenberg-Marquardt*. Seminar Nasional Aplikasi Teknologi Informasi Yogyakarta ISSN : 1907-5022
- [8] Andrian, Y., Wayahdi, M, R., 2014. Analisis Algoritma Inisialisasi *Nguyen Widrow* Pada Proses Klasifikasi Curah Hujan Kota Medan Menggunakan Metode *Backpropagation Neural Network*. Jurnal Seminar Nasional Informatika 2014, hal 57-63.
- [9] Supariasa, I. D. dkk. 2001. Penilaian status gizi. Jakarta: EGC.

- [10] Almatsier, S. 2001, Prinsip dasar ilmu gizi, Jakarta: Gramedia Pustaka Utama.
- [11] Lusa. 2009. Gizi seimbang pada remaja dan dewasa. Diambil tanggal 15 Mei 2016 dari <http://www.lusa.web.id/gizi-seimbang-pada-remaja-dan-dewasa/>
- [12] Kusumadewi, S., 2004. Membangun Jaringan Syaraf Tiruan menggunakan Matlab dan Excel Link. 1st ed. Yogyakarta: Graha Ilmu.
- [13] Kohavi, R., 1995, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, Proceedings of IJCAI Conferences 1995 : Stanford University, 1137-1143
- [14] Han, J. & Kamber, M., 2006. Data Mining : Concepts and Techniques. San Fransisco: Morgan Kaufmann Publisher.