

Penerapan *Backpropagation* untuk Meningkatkan Efektivitas Waktu dan Akurasi pada Data *Wall-Following Robot Navigation*

Aprilia Tri Wahyu Utami dan Brodjol Sutijo Supri Ulama

Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: brodjol_su@statistika.its.ac.id

Abstrak— *Mobile robot* adalah sebuah mesin otomatis yang mampu bergerak dalam suatu kondisi tertentu sehingga dibutuhkan sistem navigasi yang baik. Untuk bernavigasi dengan baik, diperlukan adaptasi yang cepat dan respon yang sesuai terhadap kejadian baru yang terjadi di sekitarnya. Dalam navigasi robot, terdapat indikasi bahwa pola klasifikasinya bersifat nonlinier. Karena itu, navigasi robot dapat menerapkan algoritma *backpropagation* untuk menginterpretasikan input sensor yang dimiliki robot. Penelitian ini akan mencari pendekatan yang dapat membuat penerapan *backpropagation* dalam data *wall-following robot navigation* lebih efektif. Proses pelatihan ini dapat dikatakan lebih efektif apabila dalam pelatihan tersebut dibutuhkan waktu yang lebih singkat namun tetap mampu mempertahankan atau bahkan dapat meningkatkan akurasi. Dalam penelitian ini, akan digunakan lima variasi pelatihan *backpropagation* antara lain yaitu *gradient descent backpropagation*, *gradient descent with momentum backpropagation*, *gradient descent with adaptive learning rate backpropagation*, *gradient descent with momentum and adaptive learning rate backpropagation*, dan *Levenberg-Marquardt backpropagation*. Berdasarkan kecepatan pelatihan, pelatihan *Levenberg-Marquardt backpropagation* mempunyai MSE yang lebih cepat konvergen daripada pelatihan lainnya. Sedangkan berdasarkan akurasi, pelatihan *gradient descent with momentum and adaptive learning rate* memiliki ketepatan klasifikasi yang paling baik untuk data *testing*. Dalam navigasi robot, lebih diutamakan pemilihan bobot dan bias yang dapat menghasilkan akurasi yang paling baik untuk *testing*. Oleh karena itu, dapat disimpulkan bahwa *gradient descent with momentum and adaptive learning rate* merupakan pelatihan yang terbaik untuk digunakan dalam data *wall-following robot navigation*.

Kata Kunci— *Wall-Following Robot Navigation*, *Backpropagation*, *Gradient Descent*, *Momentum*, *Adaptive Learning Rate*, *Levenberg-Marquardt*

I. PENDAHULUAN

Arificial Neural Network (ANN) merupakan suatu sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan saraf biologi [1]. Salah satu metode dalam *neural network* yang populer adalah *backpropagation*. Metode ini pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai dalam *neural network*. Sekitar 90% bahkan lebih, *backpropagation* telah berhasil diaplikasikan di berbagai bidang diantaranya pengenalan pola tulisan tangan, pengenalan pola suara, pengolahan citra medika, sistem kendali dan lainnya. Dewasa ini, perkembangan robotika berjalan dengan pesat di berbagai

negara. *Mobile robot* adalah sebuah mesin otomatis yang mampu bergerak dalam suatu kondisi tertentu sehingga dibutuhkan sistem navigasi yang baik. Robot membutuhkan adaptasi yang cepat dan respon yang sesuai terhadap kejadian baru yang terjadi di sekitarnya. Dalam navigasi robot, terdapat indikasi bahwa pola klasifikasinya bersifat nonlinier. Oleh karena itu, navigasi robot dapat menggunakan pelatihan *backpropagation* untuk menginterpretasikan input sensor dari robot. Penelitian ini akan mencari pendekatan yang dapat membuat penerapan *backpropagation* dalam data *wall-following robot navigation* lebih efektif. Proses pelatihan ini dapat dikatakan lebih efektif apabila dalam pelatihan tersebut membutuhkan waktu yang lebih singkat namun tetap mampu mempertahankan atau bahkan dapat meningkatkan akurasi.

Dalam penelitian ini, akan digunakan lima variasi pelatihan yaitu *gradient descent backpropagation*, *gradient descent with momentum backpropagation*, *gradient descent with adaptive learning rate backpropagation*, *gradient descent with momentum and adaptive learning rate backpropagation*, dan *Levenberg-Marquardt backpropagation*. Kelima variasi pelatihan tersebut akan diukur kinerja pelatihannya beserta bobot yang dihasilkan. Kemudian akan dibandingkan berdasarkan akurasi dan waktu yang telah diperoleh dari pelatihan sehingga dapat ditentukan variasi pelatihan *backpropagation* yang paling efektif untuk data *wall-following robot navigation*.

II. TINJAUAN PUSTAKA

A. Statistika Deskriptif

Statistika deskriptif adalah statistika yang meliputi kegiatan-kegiatan pengumpulan, penyajian, penyederhanaan atau penganalisisan dan penentuan ukuran-ukuran khusus dari suatu data tanpa penarikan kesimpulan [2]. Statistika deskriptif dilakukan tanpa ada tujuan membuat kesimpulan untuk generalisasi. Dalam penelitian ini, ukuran statistik yang digunakan untuk variabel prediktor adalah rata-rata, deviasi standar, nilai minimum dan nilai maksimum. Sedangkan variabel respon akan disajikan dalam diagram lingkaran (*pie chart*) disertai perhitungan persentase untuk tiap kelas.

B. Nonlinieritas dalam Data Klasifikasi

Data yang bersifat nonlinier dalam kasus klasifikasi adalah data dimana pola penyebaran kelas variabel respon tidak dapat dipisahkan secara garis linier sehingga dibutuhkan garis

pemisah nonlinier yang lebih mampu membedakan antar kelas. Adanya sifat nonlinier dapat dideteksi secara visual melalui *scatter plot* penyebaran kelas variabel respon berdasarkan antar variabel prediktor. Untuk data yang tidak mengikuti distribusi normal, fungsi klasifikasi linier tidak akan sesuai untuk digunakan [3].

C. Pembagian Data Training dan Data Testing

Data *training* merupakan data yang digunakan untuk membentuk model klasifikasi. Sedangkan data *testing* adalah data yang digunakan untuk mengestimasi galat dari metode yang digunakan serta dari model klasifikasi yang telah terbentuk dalam data *training*. Pembagian data *training-testing* yang umum digunakan adalah dengan metode *holdout* yang membagi sebagian data untuk data *testing* dan sisanya untuk data *training*. Setiap *dataset* harus dipilih secara independen. Namun, pembagian ini tidak menjamin bahwa data *testing* atau data *training* memiliki representatif dari setiap kelas yang ada. Oleh karena itu, diperlukan sebuah prosedur dimana data *training* dan data *testing* harus memiliki representatif untuk semua kelas. Prosedur ini disebut stratifikasi sehingga gabungan metode ini disebut *stratified holdout* [4].

D. Artificial Neural Network (ANN)

Artificial neural network (ANN) atau yang sering diterjemahkan dengan jaringan saraf tiruan merupakan suatu sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan saraf biologi [1]. *Neural network* ditentukan oleh tiga hal yaitu pola hubungan antar unit (arsitektur jaringan), metode untuk menentukan bobot dalam *link* koneksi (metode pelatihan atau algoritma) dan fungsi aktivasi [5]. Berdasarkan cara modifikasi bobotnya, ada dua macam pelatihan yang dikenal yaitu pelatihan supervisi (*supervised*) dan pelatihan tanpa supervisi (*unsupervised*). Dalam pelatihan supervisi, terdapat sejumlah pasangan data (input-target) yang digunakan untuk melatih jaringan sedangkan pelatihan tanpa supervisi (*unsupervised*) tidak ada informasi target yang digunakan. Salah satu keunggulan dari *neural network* adalah unit dalam jaringan dapat menjadi linier atau nonlinier [6]. Sebuah *neural network* yang dibuat dari interkoneksi unit yang nonlinier, maka *neural network* tersebut bersifat nonlinier. *Neural network* sangat berguna apabila digunakan untuk data yang bersifat nonlinier. Namun juga tidak membatasi data linier untuk menggunakan metode *neural network*.

E. Backpropagation

Algoritma pelatihan *backpropagation* atau yang sering disebut dengan propagasi balik termasuk metode pelatihan supervisi dimana terdapat pasangan input-target serta didesain untuk operasi jaringan *feed forward* multilapis. Dalam penelitian ini, notasi yang akan digunakan dalam algoritma pelatihan *backpropagation* adalah sebagai berikut.

- n : Banyak variabel prediktor.
- m : Banyak kategori variabel respon.
- \mathbf{x} : Vektor input, $\mathbf{x} = (x_1, \dots, x_i, \dots, x_n)$.
- \mathbf{t} : Vektor target, $\mathbf{t} = (t_1, \dots, t_k, \dots, t_m)$.
- δ_k : Informasi galat untuk unit output Y_k .
- δ_j : Informasi galat untuk unit tersembunyi Z_j .
- α : Learning rate (tingkat pembelajaran).

- X_i : Unit input ke- i .
- v_{0j} : Bias untuk unit tersembunyi ke- j .
- v_{ij} : Bobot dari unit input ke- i ke unit tersembunyi ke- j .
- Z_j : Unit tersembunyi ke- j .
- w_{0k} : Bias untuk unit output ke- k .
- w_{jk} : Bobot dari unit tersembunyi ke- j ke unit output ke- k .
- Y_k : Unit output ke- k .

Berikut merupakan algoritma pelatihan *backpropagation*.

0. Inisialisasi bobot dan parameter.
1. Ketika kondisi berhenti tidak terpenuhi, dilakukan langkah 2 sampai langkah 9.
2. Untuk setiap pasang data, lakukan langkah 3 sampai langkah 8.
3. Setiap unit input ($X_i, i=1, \dots, n$) menerima sinyal input x_i .
4. Setiap unit tersembunyi ($Z_j, j=1, \dots, p$) menghitung nilai $z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$ dan $z_j = f(z_in_j)$.
5. Setiap unit output ($Y_k, k=1, \dots, m$) menghitung nilai $y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$ dan $y_k = f(y_in_k)$.
6. Setiap unit output ($Y_k, k=1, \dots, m$) menghitung nilai $\delta_k = (t_k - y_k) f'(y_in_k)$ kemudian $\Delta w_{jk} = \alpha \delta_k z_j$ dan $\Delta w_{0k} = \alpha \delta_k$.
7. Setiap unit tersembunyi ($Z_j, j=1, \dots, p$) menghitung nilai $\delta_in_j = \sum_{k=1}^m \delta_k w_{jk}$ dan $\delta_j = \delta_in_j f'(z_in_j)$. Kemudian menghitung $\Delta v_{ij} = \alpha \delta_j x_i$ dan $\Delta v_{0j} = \alpha \delta_j$.
8. Setiap unit output ($Y_k, k=1, \dots, m$) mengupdate bias dan bobot dengan cara $w_{0k}^{(l+1)} = w_{0k}^{(l)} + \Delta w_{0k}$ dan $w_{jk}^{(l+1)} = w_{jk}^{(l)} + \Delta w_{jk}$.
Setiap unit tersembunyi ($Z_j, j=1, \dots, p$) mengupdate bias dan bobot dengan cara $v_{0j}^{(l+1)} = v_{0j}^{(l)} + \Delta v_{0j}$ dan $v_{ij}^{(l+1)} = v_{ij}^{(l)} + \Delta v_{ij}$.
9. Uji kondisi berhenti.

Dalam penelitian ini, fungsi aktivasi yang digunakan dalam unit tersembunyi dan unit output adalah fungsi aktivasi *sigmoid* bipolar. Fungsi ini memiliki range (-1,1) [7]. Berikut persamaan untuk fungsi *sigmoid* bipolar dan turunannya.

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

$$f'(x) = (1+f(x))(1-f(x)). \quad (2)$$

Selain itu, dalam penelitian ini, digunakan metode *batch updating* dimana dalam metode ini bobot dan bias akan diupdate dengan mengakumulasi galat yang didapat dari seluruh data *training* [1]. Kriteria yang digunakan untuk menghentikan suatu pelatihan terdapat dua macam cara yaitu menggunakan tingkat galat yang diinginkan dan menentukan jumlah maksimum *epoch* yang ingin dilakukan. Tinjauan pustaka mengenai variasi-variasi *backpropagation* yang akan digunakan adalah sebagai berikut.

1. Gradient Descent

Algoritma *gradient descent* mencari derivasi dari E terhadap vektor bobot \mathbf{w} untuk melakukan update bobot, dimana E merupakan energi dari perbedaan antara output dengan target yang diinginkan pada titik data ke- l dan nilai E bergantung pada nilai bobot yang diperoleh [8]. Setelah turunan parsial dari E terhadap w_{jk} dijabarkan dan disederhanakan, maka diperoleh turunan parsial dari E .

$$\frac{\partial E}{\partial w_{jk}} = -(t_k - y_k) f'(y_in_k) z_j = -\delta_k z_j, \quad (3)$$

sehingga nilai koreksi bobot w_{jk} dapat diringkas menjadi seperti berikut.

$$\Delta w_{jk} = -\alpha \frac{\partial E}{\partial w_{jk}}, \quad (4)$$

$$\Delta w_{jk} = \alpha \delta_k z_j. \quad (5)$$

2. Gradient Descent dengan Momentum

Dalam *backpropagation* yang standar, perubahan bobot didasarkan atas gradien yang terjadi untuk pola terakhir. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas kombinasi arah dari gradien pola terakhir dengan gradien pola sebelumnya [5]. Inilah yang disebut momentum. Penambahan parameter momentum dalam mengupdate bobot seringkali bisa mempercepat proses pelatihan [7]. Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang *outlier*. Namun, apabila data terakhir memiliki pola yang berbeda dengan lainnya, perubahan dilakukan secara lambat. Persamaan untuk mengupdate bobot dalam *backpropagation* dengan momentum adalah sebagai berikut.

$$w_{jk}^{(l+1)} = w_{jk}^{(l)} + \alpha \delta_k z_j + \rho [w_{jk}^{(l)} - w_{jk}^{(l-1)}], \quad (6)$$

dan,

$$v_{ij}^{(l+1)} = v_{ij}^{(l)} + \alpha \delta_j x_i + \rho [v_{ij}^{(l)} - v_{ij}^{(l+1)}], \quad (7)$$

dimana nilai parameter momentum ρ dibatasi antara 0 sampai 1 dengan nilai terbaik adalah di dekat titik terakhir [1].

3. Gradient Descent dengan Adaptive Learning Rate

Dalam standar *backpropagation*, *learning rate* merupakan suatu konstanta yang dipakai dalam seluruh *epoch*-nya. Perubahan dapat dilakukan dengan memberikan *learning rate* yang berbeda-beda untuk setiap *epoch*-nya. Apabila E meningkat, bobot baru yang terbentuk tidak akan digunakan dan *learning rate* akan turun sebesar β kali. Hal ini terjadi apabila peningkatkan E tersebut melebihi suatu nilai yang telah ditentukan sebelumnya sebagai maksimum peningkatan E . Sebaliknya apabila nilai E menurun, *learning rate* ini akan meningkat sebesar γ kali dan bobot baru yang telah terbentuk tetap digunakan [9]. Parameter β dan γ merupakan rasio yang digunakan untuk memperkecil dan meningkatkan *learning rate*. Berikut adalah perubahan bobot dan bias dengan *adaptive learning rate*.

$$w_{jk}^{(l+1)} = w_{jk}^{(l)} + \alpha^{(l+1)} \delta_k z_j, \quad (8)$$

dan,

$$v_{ij}^{(l+1)} = v_{ij}^{(l)} + \alpha^{(l+1)} \delta_j x_i. \quad (9)$$

4. Gradient Descent dengan Momentum dan Adaptive Learning Rate

Variasi *backpropagation* ini merupakan kombinasi dari dua variasi sebelumnya. Selain terdapat penambahan momentum, dalam pelatihan ini juga menerapkan *adaptive learning rate*. Oleh karena itu, perubahan bobot dan bias dengan adanya momentum dan *adaptive learning rate* adalah sebagai berikut.

$$w_{jk}^{(l+1)} = w_{jk}^{(l)} + \alpha^{(l+1)} \delta_k z_j + \rho [w_{jk}^{(l)} - w_{jk}^{(l-1)}], \quad (10)$$

dan,

$$v_{ij}^{(l+1)} = v_{ij}^{(l)} + \alpha^{(l+1)} \delta_j x_i + \rho [v_{ij}^{(l)} - v_{ij}^{(l+1)}]. \quad (11)$$

5. Levenberg-Marquardt

Metode ini merupakan kombinasi dari dua metode yaitu metode Newton dan *gradient descent* dimana metode Newton

cepat konvergen namun kurang stabil sedangkan *gradient descent* pasti konvergen namun sangat lambat [6]. Algoritma pelatihan ini menggunakan dasar metode Newton namun tidak menggunakan matriks Hessian untuk orde kedua. Matriks Hessian didekati dengan $\mathbf{H} = \mathbf{J}^T \mathbf{J}$ dan vektor gradien dengan $\mathbf{g} = \mathbf{J}^T \mathbf{e}$. Matriks \mathbf{J} adalah matriks Jacobian yang terdiri dari turunan pertama galat berdasarkan masing-masing bobot. Aturan update bobot untuk pelatihan *Levenberg-Marquardt backpropagation* adalah sebagai berikut.

$$\mathbf{w}^{(l+1)} = \mathbf{w}^{(l)} - [\mathbf{H} + \lambda \mathbf{I}]^{-1} \mathbf{g}. \quad (12)$$

Matriks \mathbf{I} adalah matriks identitas yang ukurannya bersesuaian dengan matriks \mathbf{H} . Nilai parameter λ sangat penting dalam pelatihan *Levenberg-Marquardt*. Apabila nilai λ bernilai nol, algoritma pelatihan ini akan menjadi seperti algoritma dalam metode Newton. Sedangkan apabila λ bernilai besar, pelatihan ini akan menjadi seperti algoritma *gradient descent* [10].

F. Apparent Error Rate (APER)

APER ditentukan dari hasil bagi antara observasi yang diklasifikasikan salah dengan jumlah sampel seluruhnya [3]. Nilai APER dapat dihitung dengan mudah dari matriks *confusion*. Perhitungan untuk nilai APER dan akurasi adalah sebagai berikut.

$$\text{APER} = \frac{n_{1M} + n_{2M}}{n_1 + n_1}, \quad (13)$$

$$\text{Akurasi} = (1 - \text{APER}) \times 100\%. \quad (14)$$

G. Wall-Following Robot

Robot *wall follower* merupakan robot otomatis yang gerakannya mengikuti dinding-dinding pembatas lintasan [11]. Cara kerjanya adalah dengan membaca dan mendeteksi suatu penghalang atau dinding terhadap robot dengan penggunaan sensor jarak. Dalam data penelitian ini, robot yang digunakan adalah SCITOS G5 yang dirancang untuk penggunaan *indoor* dan dilengkapi oleh 24 sensor ultrasonik yang terpasang melingkar di pinggang robot. Sensor ultrasonik tersebut akan mengukur jarak antara dinding penghalang dengan robot menggunakan gelombang ultrasonik seperti yang dilakukan oleh lumba-lumba dan kelelawar. Gelombang ultrasonik dipancarkan oleh *transmitter* dan pantulannya diterima oleh *receiver*. Prinsip kerja sensor jarak ini adalah *transmitter* mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya kembali pantulan dari dinding penghalang [12]. Lamanya waktu ini sebanding dengan dua kali jarak sensor ke dinding penghalang.

III. METODOLOGI PENELITIAN

A. Sumber Data

Data yang akan dianalisis merupakan data sekunder yang diperoleh dari UCI *machine learning repository* [13]. Data yang digunakan adalah data *wall-following robot navigation* yang merupakan data klasifikasi dengan empat kelas. Total data keseluruhan sebanyak 5456 dengan 24 variabel prediktor.

B. Variabel Penelitian

Variabel respon atau target (t) dalam penelitian ini adalah arah navigasi robot yang terdiri dari empat kelas seperti yang disajikan dalam Tabel 1.

Tabel 1. Kelas Variabel Respon

Kelas	Jumlah	Kode Biner
<i>Move-Forward</i>	2205	1 0 0 0
<i>Slight-Right-Turn</i>	826	0 1 0 0
<i>Sharp-Right-Turn</i>	2097	0 0 1 0
<i>Slight-Left-Turn</i>	328	0 0 0 1

Dalam Tabel 1 disertakan kode biner untuk menyatakan masing-masing kelas yang akan digunakan dalam analisis lebih lanjut. Sedangkan variabel prediktornya merupakan jarak antara dinding penghalang dengan robot yang merupakan pembacaan dari 24 sensor ultrasonik yang terpasang di berbagai sudut. Oleh karena itu, terdapat 24 variabel prediktor dan semuanya bertipe kontinu seperti dalam Tabel 2. Sudut dalam sensor ultrasonik hanya digunakan untuk penamaan sensor-sensor tersebut.

Tabel 2. Variabel Prediktor

Variabel	Keterangan
US1	Jarak hasil pembacaan sensor ultrasonik sudut 180°
US2	Jarak hasil pembacaan sensor ultrasonik sudut -165°
US3	Jarak hasil pembacaan sensor ultrasonik sudut -150°
US4	Jarak hasil pembacaan sensor ultrasonik sudut -135°
US5	Jarak hasil pembacaan sensor ultrasonik sudut -120°
US6	Jarak hasil pembacaan sensor ultrasonik sudut -105°
US7	Jarak hasil pembacaan sensor ultrasonik sudut -90°
US8	Jarak hasil pembacaan sensor ultrasonik sudut -75°
US9	Jarak hasil pembacaan sensor ultrasonik sudut -60°
US10	Jarak hasil pembacaan sensor ultrasonik sudut -45°
US11	Jarak hasil pembacaan sensor ultrasonik sudut -30°
US12	Jarak hasil pembacaan sensor ultrasonik sudut -15°
US13	Jarak hasil pembacaan sensor ultrasonik sudut 0°
US14	Jarak hasil pembacaan sensor ultrasonik sudut 15°
US15	Jarak hasil pembacaan sensor ultrasonik sudut 30°
US16	Jarak hasil pembacaan sensor ultrasonik sudut 45°
US17	Jarak hasil pembacaan sensor ultrasonik sudut 60°
US18	Jarak hasil pembacaan sensor ultrasonik sudut 75°
US19	Jarak hasil pembacaan sensor ultrasonik sudut 90°
US20	Jarak hasil pembacaan sensor ultrasonik sudut 105°
US21	Jarak hasil pembacaan sensor ultrasonik sudut 120°
US22	Jarak hasil pembacaan sensor ultrasonik sudut 135°
US23	Jarak hasil pembacaan sensor ultrasonik sudut 150°
US24	Jarak hasil pembacaan sensor ultrasonik sudut 165°

C. Langkah Analisis

Langkah-langkah yang digunakan sebagai pedoman analisis dalam penelitian ini adalah sebagai berikut .

1. Melakukan analisis statistika deskriptif untuk variabel respon dan variabel prediktor.
2. Mengetahui sifat nonlinieritas dalam data *wall-following robot navigation* dengan menggunakan *scatter plot* pola penyebaran kelas variabel respon berdasarkan antar variabel prediktor.
3. Melakukan pembagian data *training* dan data *testing* dengan berbagai proporsi.
4. Melakukan pelatihan *backpropagation* antara lain yaitu *gradient descent backpropagation*, *gradient descent with momentum backpropagation*, *gradient descent with adaptive learning rate backpropagation*, *gradient descent with momentum and adaptive learning rate backpropagation* dan yang terakhir adalah pelatihan *Levenberg-Marquardt backpropagation*. Setiap pelatihan tersebut akan diukur hasil kinerjanya.
5. Melakukan perbandingan dan pengambilan kesimpulan.

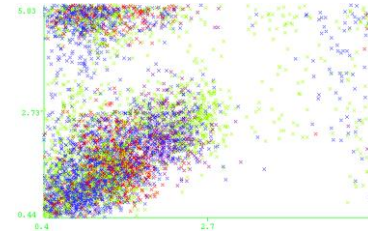
IV. ANALISIS DAN PEMBAHASAN

A. Statistika Deskriptif

Variabel respon dalam data *wall-following robot navigation* adalah arah navigasi robot yang terdiri dari empat kelas antara lain *move forward* dengan persentase 40%, *slight right turn* 15%, *sharp right turn* 39% dan *slight left turn* 7%. Perbedaan banyak data tiap kelas ini menyebabkan ketidakseimbangan data dalam klasifikasi (*unbalanced data classification*). Berdasarkan hasil analisis statistika deskriptif untuk variabel prediktor, US7 (sensor ultrasonik di sudut -90°) memiliki rata-rata tertinggi artinya sensor ini merupakan sensor ultrasonik yang paling sering memiliki jarak yang jauh dari dinding penghalang. Sedangkan sensor ultrasonik yang memiliki rata-rata terkecil adalah US18 (sensor ultrasonik di sudut 75°) artinya sensor ini paling sering memiliki jarak yang dekat dengan dinding penghalang. Jarak yang paling dekat dengan dinding penghalang yaitu sebesar 0,340 yang merupakan hasil pembacaan US19 (sensor ultrasonik di sudut 90°). Sedangkan US8 (sensor ultrasonik di sudut -75°) memiliki jarak yang paling jauh dari dinding penghalang yaitu sebesar 5,087.

B. Nonlinieritas dalam Data Klasifikasi

Berikut merupakan contoh *scatter plot* penyebaran kelas arah navigasi robot berdasarkan variabel US1 (sensor ultrasonik di sudut 180°) dan variabel US2 (sensor ultrasonik di sudut -165°).



Gambar 1. Scatter Plot Kelas Arah Navigasi Robot Berdasarkan Variabel US1 dan Variabel US2

Gambar 1 menunjukkan bahwa pola penyebaran kelas respon tersebut sangat acak dan tidak beraturan. Secara visual, antar kelas arah navigasi robot tidak dapat diklasifikasikan secara linier. Oleh sebab itu, dibutuhkan pemisah klasifikasi nonlinier yang lebih mampu membedakan antar kelas dibandingkan pemisah klasifikasi linier. Pola inilah yang biasa disebut dengan pola nonlinier dalam data klasifikasi.

C. Pembagian Data Training dan Data Testing

Penggunaan metode ini dikarenakan oleh penerapannya yang sangat sederhana dan sering digunakan serta data yang digunakan cukup besar sehingga data *training* dan data *testing* masih mempunyai data yang cukup. Dalam penelitian ini, akan digunakan beberapa proporsi pembagian data *training-testing* yaitu 90:10, 80:20, 70:30, 60:40, 50:50 dan 2:1. Dari proporsi-proporsi tersebut, akan dipilih proporsi pembagian data *training-testing* terbaik. Untuk memilih proporsi pembagian data *training-testing* yang terbaik, akan diterapkan salah satu variasi pelatihan yaitu *Levenberg-Marquardt backpropagation* untuk berbagai proporsi tersebut. Berikut adalah hasil evaluasi kinerja *Levenberg-Marquardt backpropagation* untuk masing-masing proporsi pembagian data *training-testing*.

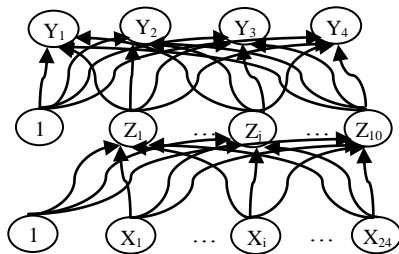
Tabel 3. Hasil Evaluasi Kinerja *Levenberg-Marquardt Backpropagation* untuk Berbagai Proporsi Pembagian Data *Training-Testing*

Proporsi Pembagian	Epoch	Training		Testing
		MSE	Akurasi (%)	Akurasi (%)
90:10	2064	0,0168	95,947	83,883
80:20	438	0,0048	98,579	85,066
70:30	1942	0,0166	95,941	88,149
60:40	77	0,0197	94,808	85,884
50:50	323	0,0061	98,094	87,238
2:1	826	0,0044	98,900	84,653

Seluruh pelatihan *Levenberg-Marquardt backpropagation* tersebut berhenti ketika gradien yang dihasilkan telah terlampaui kecil yaitu lebih kecil dari 1×10^{-5} . Hal ini dapat dikatakan bahwa MSE yang dihasilkan sudah mencapai kondisi yang konvergen. Dalam kasus klasifikasi, akurasi yang diperoleh dalam data *testing* lebih menjadi tolok ukur pertama daripada akurasi yang didapatkan dalam data *training*. Oleh karena itu, berdasarkan Tabel 3 proporsi pembagian data yang akan digunakan dalam pelatihan *backpropagation* adalah 70:30 dimana dengan proporsi tersebut memiliki akurasi data *testing* yang paling tinggi.

D. Variasi Pelatihan Backpropagation

Terdapat lima variasi pelatihan yang akan digunakan untuk data *wall-following robot navigation*. Jumlah unit lapisan tersembunyi yang digunakan adalah 10 unit. Arsitektur jaringan dapat dilihat dalam Gambar 2.



Gambar 2. Arsitektur *Backpropagation*

Fungsi aktivasi yang digunakan dalam unit tersembunyi dan unit output adalah *sigmoid bipolar (tansig)* serta menggunakan *learning rate* sebesar 0,01. Setiap pelatihan akan berhenti ketika MSE telah mencapai 0,003 atau kurang dari nilai tersebut. Namun, *epoch* dalam penelitian ini akan dibatasi sampai 10000 *epoch*.

1. Gradient Descent Backpropagation

Pelatihan *gradient descent backpropagation* berhenti saat *epoch* maksimum. Pada saat *epoch* ke-10000, MSE yang dihasilkan adalah sebesar 0,120. Nilai MSE yang didapatkan belum konvergen. Apabila jumlah *epoch* ditambahkan, akan diperoleh hasil yang lebih baik. Namun, pertambahan *epoch* ini hanya mengurangi sedikit demi sedikit MSE. Oleh karena itu, untuk mencapai target MSE akan memerlukan pertambahan *epoch* yang sangat banyak. Waktu pelatihan yang diperlukan selama 10000 *epoch* adalah 290,156 detik atau 4 menit 50,156 detik dengan rata-rata waktu per *epoch*-nya sekitar 0,029 detik. Sedang akurasi yang didapatkan yaitu sebesar 65,096%. Dengan menggunakan bobot yang sama, maka diperoleh akurasi data *testing* sebesar 66,402%. Sedangkan waktu yang diperlukan untuk pengujian data *testing* adalah 0,012 detik.

2. Gradient Descent with Momentum Backpropagation

Pada saat *epoch* maksimum pelatihan *gradient descent with momentum backpropagation* belum mencapai nilai MSE yang diinginkan. Nilai MSE yang dicapai adalah 0,120. Seperti halnya dalam pelatihan *gradient descent backpropagation*, MSE belum mencapai titik yang konvergen sehingga perbaikan dari bobot dan bias masih dapat dilakukan dengan penambahan *epoch* yang sangat banyak. Waktu yang diperlukan selama pelatihan 10000 *epoch* adalah 285,460 detik atau 4 menit 45,460 detik dengan rata-rata waktu pelatihan per *epoch* adalah sebesar 0,029 detik. Akurasi yang didapatkan dengan pelatihan ini adalah sebesar 65,096%. Sedangkan untuk pengujian data *testing* didapatkan nilai akurasi sebesar 66,402%. Sedangkan waktu yang diperlukan untuk merespon data *testing* tersebut adalah 0,012.

3. Gradient Descent with Adaptive Learning Rate Backpropagation

Pelatihan *gradient descent with adaptive learning rate backpropagation* lebih baik apabila dibandingkan dengan kedua pelatihan sebelumnya. Hal ini dapat dilihat dari nilai akurasi yang cukup besar untuk data *training* yaitu sebesar 94,292%. Namun dalam pelatihan ini, target MSE masih belum tercapai. Pada saat *epoch* ke 10000, MSE yang dicapai adalah sebesar 0,028. MSE setiap *epoch* mengalami fluktuatif yang kecil namun tetap monoton turun. Apabila dibandingkan dengan kedua pelatihan sebelumnya, grafik MSE dalam pelatihan ini mengalami penurunan yang lebih tajam. Namun, MSE yang dihasilkan masih belum mencapai kondisi yang konvergen. Waktu yang dibutuhkan dalam pelatihan selama 10000 *epoch* adalah 293,748 detik atau 4 menit 53,748 detik dengan rata-rata 0,029 detik per *epoch*-nya. Akurasi untuk data *testing* juga cukup bagus dibandingkan kedua pelatihan sebelumnya yaitu sebesar 90,531% dengan waktu yang dibutuhkan untuk pengujian data *testing* adalah selama 0,012 detik.

4. Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation

Gabungan pelatihan tersebut mendapatkan akurasi yang lebih baik daripada pelatihan-pelatihan sebelumnya. Akurasi untuk data *training* yang didapatkan adalah sebesar 95,706%. Begitu juga dengan nilai MSE, dimana lebih kecil daripada ketiga pelatihan sebelumnya yaitu sebesar 0,021. Namun, masih belum memenuhi target MSE. Grafik MSE lebih sedikit mengalami fluktuasi dibandingkan dengan pelatihan *gradient descent with adaptive learning rate*. Penurunan MSE dalam pelatihan ini juga lebih tajam daripada ketiga pelatihan sebelumnya. Akan tetapi, MSE yang didapatkan masih belum mencapai kondisi yang konvergen. Pelatihan berjalan selama 302,890 detik atau 5 menit 2,890 detik dengan rata-rata waktu per *epoch* adalah 0,030 detik. Waktu yang dibutuhkan pada saat *testing* adalah selama 0,013 detik dan akurasi yang dihasilkan adalah sebesar 91,264%.

5. Levenberg-Marquardt Backpropagation

Kinerja pelatihan *Levenberg-Marquardt backpropagation* belum mencapai target MSE. Pemberhentian pelatihan ini dikarenakan gradien yang dihasilkan pada saat *epoch* 241 telah lebih kecil dari minimum gradien yang telah ditentukan

yaitu 1×10^{-5} . Nilai MSE untuk setiap *epoch* dari pelatihan *Levenberg-Marquardt backpropagation* turun tajam saat *epoch* awal. Pergerakan MSE ini sudah dalam keadaan yang konvergen namun belum mencapai target MSE. Hal ini dapat diartikan bahwa hasil kinerja pelatihan ini tidak dapat ditingkatkan lagi dengan penambahan *epoch*. Waktu pelatihan yang dibutuhkan selama 241 *epoch* adalah 133,825 detik atau 2 menit 13,825 detik dengan rata-rata 0,555 detik per *epoch*-nya. Sedangkan akurasi yang didapatkan yaitu 95,863%. Akurasi yang didapat untuk data *testing* adalah sebesar 88,027% dengan waktu pelatihan 0,014 detik.

E. Pemilihan Variasi Backpropagation Terbaik

Hasil kinerja kelima variasi pelatihan *backpropagation* akan dibandingkan satu sama lain berdasarkan kecepatan dan akurasi. Perbandingan kecepatan dan akurasi akan ditampilkan dalam Tabel 4 dan Tabel 5.

Tabel 4. Perbandingan Kecepatan Pelatihan

Fungsi Pelatihan	Training			Testing
	Banyak epoch	Waktu* (detik)	Waktu per epoch* (detik)	Waktu* (detik)
<i>traingd</i>	10000	290,156	0,029	0,012
<i>traingdm</i>	10000	285,460	0,029	0,012
<i>traingda</i>	10000	293,748	0,029	0,012
<i>traingdx</i>	10000	302,890	0,030	0,013
<i>trainlm</i>	241	133,825	0,555	0,014

*Nilai bisa berubah-ubah

Tabel 5. Perbandingan Akurasi Pelatihan

Fungsi Pelatihan	Training		Testing
	MSE	Akurasi (%)	Akurasi (%)
<i>traingd</i>	0,120	65,096	66,402
<i>traingdm</i>	0,120	65,096	66,402
<i>traingda</i>	0,028	94,292	90,531
<i>traingdx</i>	0,021	95,706	91,264
<i>trainlm</i>	0,016	95,863	88,027

Waktu pelatihan dapat berubah-ubah setiap kali *running*. Hal ini dikarenakan waktu *running* tersebut bergantung pada kondisi komputer pada saat *running*. Waktu pelatihan yang didapatkan juga akan berbeda apabila *running* dilakukan dengan komputer yang berbeda. Hal ini dikarenakan spesifikasi komputer sangat berpengaruh terhadap kinerja komputer, secara langsung juga akan berpengaruh terhadap lama waktu pelatihan. Dalam pelatihan ini, digunakan komputer dengan spesifikasi RAM 2.00 GB dan prosesor Intel® Core™ i3-2350M CPU @ 2.30 GHz.

Berdasarkan kecepatan, variasi pelatihan *Levenberg-Marquardt backpropagation* lebih cepat konvergen dibandingkan lainnya. Sedangkan berdasarkan akurasi, maka *gradient descent with momentum and adaptive learning rate* memiliki ketepatan klasifikasi yang paling baik untuk data *testing*. Untuk data *wall-following robot navigation*, lebih diutamakan pemilihan bobot dan bias yang dapat menghasilkan akurasi untuk data *testing* yang paling baik. Secara keseluruhan, dapat disimpulkan bahwa pelatihan *gradient descent with momentum and adaptive learning rate backpropagation* merupakan pelatihan yang terbaik untuk digunakan. Dengan pemilihan metode ini, bobot dan bias masih dapat diupdate kembali dengan penambahan *epoch* agar hasil akurasi yang didapatkan bertambah baik.

Apabila diurutkan kinerja kelima variasi pelatihan *backpropagation* tersebut, peringkat variasi *backpropagation* dari pelatihan dengan hasil kinerja terbaik adalah: *gradient descent with momentum and adaptive learning rate backpropagation*, *Levenberg-Marquardt backpropagation*, *gradient descent with adaptive learning rate backpropagation*, *gradient descent with momentum backpropagation* dan yang terakhir adalah *gradient descent backpropagation*.

V. KESIMPULAN DAN SARAN

Berdasarkan kecepatan, pelatihan *Levenberg-Marquardt backpropagation* mempunyai MSE yang lebih cepat konvergen dibandingkan dengan variasi pelatihan *backpropagation* yang lain. Sedangkan berdasarkan akurasi, maka variasi pelatihan *gradient descent with momentum and adaptive learning rate backpropagation* memiliki ketepatan klasifikasi yang paling baik untuk data *testing*. Secara keseluruhan, dapat disimpulkan bahwa *gradient descent with momentum and adaptive learning rate* merupakan pelatihan yang terbaik untuk digunakan sebagai dasar sistem navigasi robot. Hal ini dikarenakan dalam sistem navigasi robot, lebih diutamakan pemilihan bobot dan bias yang dapat menghasilkan akurasi yang terbaik terutama pada saat *testing*.

Berdasarkan hasil analisis, dapat disarankan untuk sistem kendali robot dapat menggunakan pelatihan *gradient descent with momentum and adaptive learning rate backpropagation*. Dengan menggunakan pelatihan ini, akan diperoleh kesalahan bernavigasi robot yang sangat kecil. Selain itu, pelatihan ini juga dapat memberikan hasil kinerja yang baik untuk permasalahan di bidang lain terutama apabila datanya bersifat nonlinier. Saran untuk penelitian selanjutnya ialah dalam penentuan nilai parameter yang akan digunakan sebaiknya disesuaikan dengan permasalahan dan teori yang ada.

DAFTAR PUSTAKA

- [1] Fausett, L. (1994). *Fundamental of Neural Network Architectures, Algorithms, and Applications*. New Jersey: Prentice-Hall.
- [2] Nugroho, S. (2007). *Dasar-dasar Metode Statistika*. Jakarta: Grasindo.
- [3] Johnson, N. and Wichern, D. (2007). *Applied Multivariate Statistical Analysis, Sixth Edition*. New Jersey: Pearson Education, Inc.
- [4] Witten, I.H, Frank, E. dan Hall, M.A. (2011). *Data Mining: Pratical Machine Learning Tools and Techniques*. 3rd Ed. Massachusetts: Morgan Kaufmann.
- [5] Siang, J.J.S. (2009). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: Penerbit ANDI.
- [6] Haykin, S. (2009). *Neural Network and Learning Machines 3rd Edition*. New Jersey: Pearson International Edition.
- [7] Puspitaningrum, D. (2006). *Pengantar Jaringan Saraf Tiruan*. Yogyakarta: Penerbit ANDI.
- [8] Santosa, B. (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
- [9] Battiti, R. (1989). Accelerated Backpropagation Learning: Two Optimization Methods. *Complex System*. 3. p.331-342.
- [10] Yu, H. dan Wilamowski, B.M. (2010). *Levenberg-Marquadt Training*. K10149_C012.indd.
- [11] Winarno dan Arifianto, D. (2011). *Bikin Robot itu Gampang*. Jagakarsa: PT. Kawan Pustaka.
- [12] McComb, G. dan Predko, M. (2006). *The Robot Builder's Bonanza*. 3rd Ed. New York: McGraw-Hill Education.
- [13] UCI Machine Learning Repository. (2010). *Wall-Following Robot Navigation Data Data Set* [Online]. Diakses dari: <http://archive.ics.uci.edu/ml/datasets/WallFollowing+Robot+Navigation+Data>.