

Identifikasi Posisi Robot *Quadpod* pada Arena Pertandingan Menggunakan Jaringan Syaraf Tiruan - Algoritma *Backpropagation*

Dimas Hutomo Daud Saputro
 Program Studi Teknik Elektro
 Universitas Hang Tuah
 Surabaya, Indonesia
 s.dimas@hangtuah.ac.id

Joko Subur
 Program Studi Teknik Elektro
 Universitas Hang Tuah
 Surabaya, Indonesia
 joko.subur@hangtuah.ac.id

M Taufiqurrohman
 Program Studi Teknik Elektro
 Universitas Hang Tuah
 Surabaya, Indonesia
 taufiqurrohman@hangtuah.ac.id

Abstract— The Indonesian Robot Contest (KRI) is an event to develop creativity and enrich science and technology in the field of robotics. In KRI, there are several branch categories, one of which is the KRPAI (Indonesian Fire Fighting Robot Contest) category. In the KRPAI category competition, the robot aims to find a room where there is a fire, then the robot will extinguish the fire in the room, and return to the starting point. This requires the robot to know its own actual position so that it can return to the starting point. This can be solved by adding artificial intelligence (Artificial Intelligence) artificial neural networks. One of the neural network models used for the learning process is backpropagation. It should be noted that in artificial intelligence systems, artificial neural networks in order to work optimally in identifying the position of the robot, it is necessary to carry out a learning process (learning) first in order to obtain optimal weight values. The learning system is carried out by entering 280 position data samples that present 28 positions. Learning data is obtained by reading the distance between the robot and the maze wall and the direction facing the robot. To test the artificial neural network system, a running process is carried out to determine the position of the robot. From 140 trial data conducted, the artificial neural network was able to recognize the position of the robot correctly with an accuracy of 92.85%. The simulation results are expected to be applied to the real quadpod robot.

Keywords— Robot Berkaki, Jaringan Syaraf Tiruan, Posisi Robot, Pembelajaran, Bobot Optimal

I. PENDAHULUAN

Dunia elektronika yang semakin berkembang pesat tidak terlepas dari sistem kendali otomatis, salah satunya adalah sistem robot. Robot tidak hanya dapat dikendalikan secara konvensional tetapi telah dikembangkan dengan kecerdasan buatan[1]. Kontes Robot Indonesia (KRI) merupakan salah satu ajang untuk mengembangkan kreatifitas dan memperkaya ilmu pengetahuan dan teknologi dalam bidang robotika[2]. Dalam KRI terdapat beberapa cabang kategori di mana salah satunya adalah kategori KRPAI (Kontes Robot Pemadam Api Indonesia). Dalam kompetisi kategori KRPAI tersebut robot bertujuan mencari ruangan yang terdapat api, lalu robot akan

memadamkan api yang ada pada ruangan tersebut, dan kembali ke titik awal.

Dalam pertandingan ini lintasan robot pemadam api berbentuk simulasi interior rumah dengan 4 ruangan. Untuk mengatasi ketidak mampuan robot untuk kembali ke titik awal maka robot harus dapat mengetahui kondisi ruangan dengan jelas[3]. Hal ini diharuskan robot harus mengetahui posisi actual dirinya sendiri sehingga dapat kembali ke titik awal. Hal itu dapat diselesaikan dengan menambahkan kecerdasan buatan (*Artificial Inteligent*) jaringan syaraf tiruan.

Dengan proses pembelajaran, jaringan syaraf tiruan akan terus memperbaiki diri agar menghasilkan nilai yang diinginkan. Data pembelajaran diperoleh dengan membaca jarak antara robot dan dinding labirin. Jarak yang diamati adalah jarak di sisi depan, kiri dan kanan yang sesuai dengan penempatan sensor ultrasonik. Sensor kompas akan membaca arah robot.

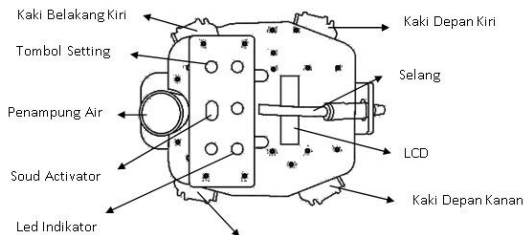
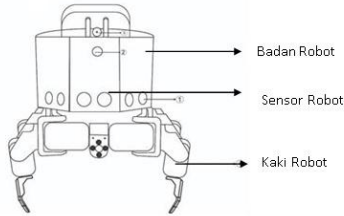
Data dari sensor ultrasonik dan sensor kompas akan menjadi inputan bagi jaringan syaraf tiruan. Pada hidden layer memiliki 2 layer dan 15 neuron dengan harapan dengan adanya hidden layer ini akan membantu proses learning dalam mencapai target. Pada layer output digunakan 5 neuron, 5 neuron tersebut merupakan representasi dari 5 data output/5 bit dengan asumsi nanti ditentukan 28 posisi atau lebih, dimana nanti masing-masing posisi tersebut akan di representasikan dengan hasil output kombinasi dengan 5 bit tersebut. Alasan dipilihnya jaringan syaraf tiruan ini karena jaringan syaraf tiruan merupakan suatu metode yang dapat digunakan untuk mengenali pola, peramalan dan klasifikasi. Diharapkan dengan penggunaan Jaringan Syaraf Tiruan dengan algoritma *backpropagation* hasil penelitian ini robot berkaki dapat mengidentifikasi Posisi robot berada.

II. TINJAUAN PUSTAKA

A. Robot Berkaki

Robot berkaki adalah robot yang bermanuver dengan kaki-kaki buatan, disebut dengan robot *humanoid*, berkaki tiga (*tripod*), berkaki empat (*quadpod*), berkaki enam (*hexapod*). Dari berbagai

litelatur robot dapat didefinisikan sebagai sebuah alat mekanik yang dapat diprogram berdasarkan informasi dari lingkungan (melalui sensor) sehingga dapat melaksanakan beberapa tugas tertentu baik secara otomatis ataupun tidak sesuai program yang dimasukkan berdasarkan logika[4]. Pada penelitian ini robot yang di bahas adalah mengenai robot *quadpod*. Robot adalah alat mekanik yang dapat melakukan tugas fisik baik melalui kontrol manusia maupun secara otomatis. Salah satu contoh yaitu robot *quadruped*[5]. Robot *quadruped* merupakan robot yang menirukan anatomi dari laba-laba dalam proses gerakannya.



Gambar 1. Robot Berkaki

B. Jaringan Syaraf Tiruan

Neural Networks (Jaringan Syaraf Tiruan) merupakan sistem prosesor paralel terhubung satu sama lain dalam bentuk grafik diarahkan. menurut bagan setiap neuron dari jaringan direpresentasikan sebagai *node*[6].

Jaringan syaraf tiruan terdiri dari 3 lapisan, yaitu *Input Layer, Hidden Layer dan Output Layer*, Setiap lapisan bertanggung jawab untuk melakukan fungsi yang sama melengkapi sistem[7]. *Back Propagation Neural Network* (BP-NN) adalah sistem komputasi di mana model memiliki beberapa kesamaan dengan prinsip sistem saraf manusia. Saraf manusia terdiri dari beberapa bagian seperti soma, dendrit, dan akson. Agar jaringan ini dapat belajar, ia harus dilatih dengan beberapa data pelatihan sesuai dengan topologinya[8].

Algoritma pelatihan *Backpropogation* diuraikan dalam langkah-langkah atau alur prosedur sebagai berikut :

1. Inisialisasi nilai penimbang, biasanya nilainya kecil dan random
2. Proses *looping*/pengulangan dihentikan jika kondisi false, lakukan langkah 3-10
3. Untuk setiap pasang data pelatihan, lakukan langkah 4-9
4. Masing-masing unit masukan ($X_i, i = 1,2,...n$) menerima sinyal masukan x_i .

5. Untuk masing-masing unit lapis tersembunyi ($Z_j, j = 1,2...p$) jumlah nilai penimbang sinyal masukan,

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \tag{1}$$

Kemudian dimasukan dalam *transfer function* untuk menghitung sinyal keluaran :

$$z_j = f(z_in_j) \tag{2}$$

6. Masing-masing unit keluaran ($Y_k, k = 1...m$) jumlahkan nilai penimbang sinyal masukan

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \tag{3}$$

Nilai ini dimasukan dalam keluaran fungsi pengaktif

$$y_k = f(y_in_k) \tag{4}$$

7. Unit keluaran ($Y_k, k = 1...m$) menerima *target pattern* yang berhubungan dengan *pattern* masukan data pelatihan.

$$\delta_k = (t_k - y_k) f'(y_in_k) \tag{5}$$

Menghitung kolerasi penimbang gunanya untuk memperbarui nilai penimbang

$$\Delta w_{jk} = \alpha \delta_k z_j \tag{6}$$

Menghitung kolerasi bias

$$\Delta w_{0k} = \alpha \delta_k \tag{7}$$

8. Unit lapis tersembunyi ($Z_j, j=1,2...p$) jumlahkan ke masukan delta

$$\delta_{in_j} = \sum_{k=1}^m \delta_{kw} z_{jk} \tag{8}$$

Menghitung galat

$$\delta_j = \delta'_{in_j} f'(z_in_j) \tag{9}$$

Menghitung kolerasi penimbang

$$\Delta v_{ij} = a \delta_j x_i \tag{10}$$

Menghitung kolerasi bias

$$\Delta v_{0j} = a \delta_j \tag{11}$$

9. Unit keluaran ($Y_k, k = 1...m$) melakukan perbaikan biasnya dan penimbangannya

$$w_{jk}(\text{baru}) = w_{jk}(\text{sekarang}) + \Delta w_{jk} \tag{12}$$

1 unit lapis tersembunyi ($Z_j, j = 1...p$) melakukan update/perbaikan bias dan penimbangannya

$$v_{ij}(\text{baru}) = v_{ij}(\text{sekarang}) + \Delta v_{ij} \tag{13}$$

10. Perbaharui nilai epoch

$$\text{epoch} = \text{epoch} + 1 \tag{14}$$

Kemudian, keluaran jaringan dibandingkan dengan target, hitung *error* E menggunakan :

$$E = t - y_k \tag{15}$$

Untuk menghitung *mean square error* (MSE), dirumuskan :

$$MSE = \frac{\sum_{k=1}^a (t_k - y_k)^2}{a} \tag{16}$$

III. METODE PENELITIAN

A. Rancangan Diagram Blok Sistem

Dari hasil studi literatur dan penelitian yang sudah dilakukan maka telah didapatkan sebuah rancangan blok sistem dari identifikasi posisi robot dengan

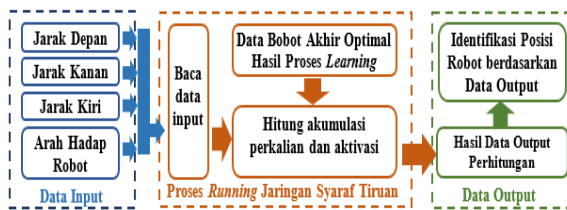
kecerdasan buatan jaringan syaraf tiruan. Pada sistem jaringan syaraf tiruan yang dibuat meliputi dua langkah tahapan, yaitu tahapan pembelajaran (*learning*) dan tahapan ujicoba (*running*).



Gambar 2. Diagram blok sistem tahapan pembelajaran (*learning*) Jaringan Syaraf Tiruan

Berdasarkan Gambar. 2 Menjelaskan proses *learning* Jaringan Syaraf Tiruan. Perlu diketahui bahwa pada sistem kecerdasan buatan jaringan syaraf tiruan supaya dapat berkerja secara optimal dalam meng-identifikasi posisi robot, maka perlu dilakukan proses pembelajaran (*learning*) terlebih dahulu guna untuk mendapatkan nilai bobot yang optimal. Dalam proses learning dilakukan dengan memasukkan data input yang meliputi data jarak depan, kiri, kanan dan data kompas beserta data output target Posisi robot.

Setelah Jaringan Syaraf Tiruan membaca input maka akan dilakukan perhitungan akumulasi dan aktivasi sampai mendapatkan nilai minimal MSE, jika nilai MSE masih lebih besar daripada nilai error minimum maka akan dilakukan proses backward untuk mengupdate nilai bobot pada tiap neuron masing-masing layer. Setelah nilai *MSE* lebih kecil daripada nilai *error minimum* atau iterasi yang diatur sudah maksimal maka proses *learning* akan berhenti. Nilai bobot yang optimal akan disimpan guna untuk proses *running* pada sistem jaringan syaraf tiruan.



Gambar 3. Diagram blok sistem proses *Running* Jaringan Syaraf Tiruan untuk identifikasi posisi robot

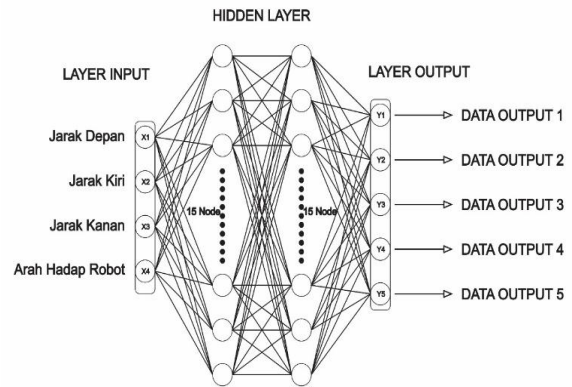
Untuk menguji sistem jaringan syaraf tiruan yang dibuat maka dilakukan proses *running* untuk mengetahui Posisi robot. Ketika sudah didapat nilai bobot optimal tiap neuron dalam setiap layer, nilai bobot akan digunakan untuk melakukan perhitungan pada proses *running* jaringan syaraf tiruan dalam melakukan pembacaan data dan pengambilan keputusan berdasarkan kondisi dari data yang masuk.

B. Pembuatan Algoritma Jaringan Syaraf Tiruan

Back propagation neural network (BP-NN) adalah sistem komputasi di mana model memiliki beberapa kesamaan dengan prinsip sistem saraf manusia. Saraf

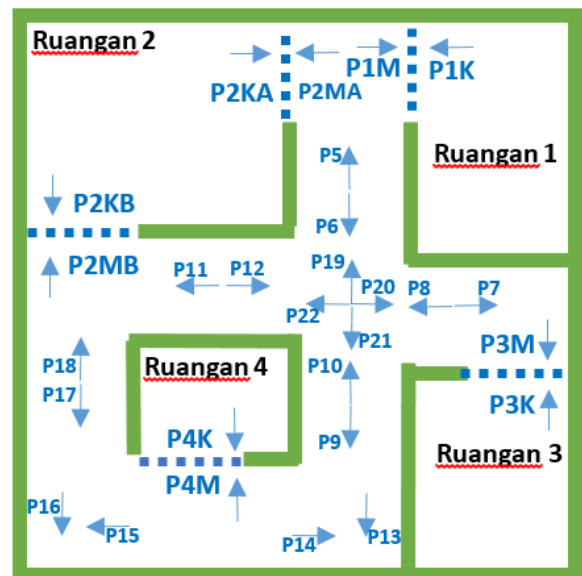
manusia terdiri dari beberapa bagian seperti soma, dendrit, dan akson. Agar jaringan ini dapat belajar, ia harus dilatih dengan beberapa data pelatihan sesuai dengan topologinya. JST dibuat dengan 8 input yang digunakan adalah jarak actual antara robot dengan dinding depan, kiri dan kanan. Orientasi robot dari pembacaan sensor kompas dimasukkan ke dalam JST (utara, barat, timur dan selatan).

Data pembelajaran diperoleh dengan membaca jarak antara robot dan dinding labirin. Jarak yang diamati adalah jarak di sisi depan, kiri, kanan, dan yang sesuai dengan penempatan sensor ultrasonik. Sensor kompas akan membaca arah robot.



Gambar 4. Topologi Jaringan Syaraf Tiruan yang digunakan.

Gambar 4 merupakan topologi dari Jaringan Syaraf Tiruan dengan memiliki 4 input 2 hidden layer dan 5 output. Data dari sensor ultrasonik dan sensor kompas akan menjadi inputan bagi jaringan syaraf tiruan. X1 merupakan data jarak depan, X2 jarak kiri, X3 jarak kanan dan X4 arah hadap robot.



Gambar 5. Posisi Robot

Pada hidden layer memiliki 15 neuron dengan harapan dengan adanya hidden layer ini akan membantu proses learning dalam mencapai target.

Pada layer output digunakan 5 neuron, 5 neuron tersebut merupakan representasi dari 5 data output/5 bit dengan asumsi nanti ditentukan 28 posisi atau lebih, dimana nanti masing-masing posisi tersebut akan di representasikan dengan hasil output kombinasi dengan 5 bit tersebut.

TABEL 1. KETERANGAN KODE DAN POSISI

No.	Kode Posisi	Keterangan Posisi	Target Data Output				
			5	4	3	2	1
0	-	Target tidak terdeteksi	0	0	0	0	0
1	P1M	Robot masuk ruangan 1	0	0	0	0	1
2	P1K	Robot keluar ruangan 1	0	0	0	1	0
3	P2MA	Robot masuk di ruangan 2 pintu A	0	0	0	1	1
4	P2KA	Robot keluar di ruangan 2 pintu A	0	0	1	0	0
5	P2MB	Robot masuk di ruangan 2 pintu B	0	0	1	0	1
6	P2KB	Robot keluar di ruangan 2 pintu B	0	0	1	1	0
7	P3M	Robot masuk ruangan 3	0	0	1	1	1
8	P3K	Robot keluar ruangan 3	0	1	0	0	0
9	P4M	Robot masuk ruangan 4	0	1	0	0	1
10	P4K	Robot keluar ruangan 4	0	1	0	1	0
11	P5	Robot berada di posisi 5	0	1	0	1	1
12	P6	Robot berada di posisi 6	0	1	1	0	0
13	P7	Robot berada di posisi 7	0	1	1	0	1
14	P8	Robot berada di posisi 8	0	1	1	1	0
15	P9	Robot berada di posisi 9	0	1	1	1	1
16	P10	Robot berada di posisi 10	1	0	0	0	0
17	P11	Robot berada di posisi 11	1	0	0	0	1
18	P12	Robot berada di posisi 12	1	0	0	1	0
19	P13	Robot berada di posisi 13	1	0	0	1	1
20	P14	Robot berada di posisi 14	1	0	1	0	0
21	P15	Robot berada di posisi 15	1	0	1	0	1
22	P16	Robot berada di posisi 16	1	0	1	1	0
23	P17	Robot berada di posisi 17	1	0	1	1	1
24	P18	Robot berada di posisi 18	1	1	0	0	0
25	P19	Robot berada di posisi 19	1	1	0	0	1
26	P20	Robot berada di posisi 20	1	1	0	1	0

27	P21	Robot berada di posisi 21	1	1	0	1	1
28	P22	Robot berada di posisi 22	1	1	1	0	0

Setelah proses pembelajaran selesai, bobot yang diperoleh disimpan dalam file teks. Nilai bobot dimasukkan ke dalam robot. Pemrograman robot dilakukan dengan menggunakan delphi. Bobot dijalankan oleh proses umpan maju untuk menentukan posisi robot di labirin. Hasil eksekusi program ditampilkan dalam program delphi yang menunjukkan posisi robot.

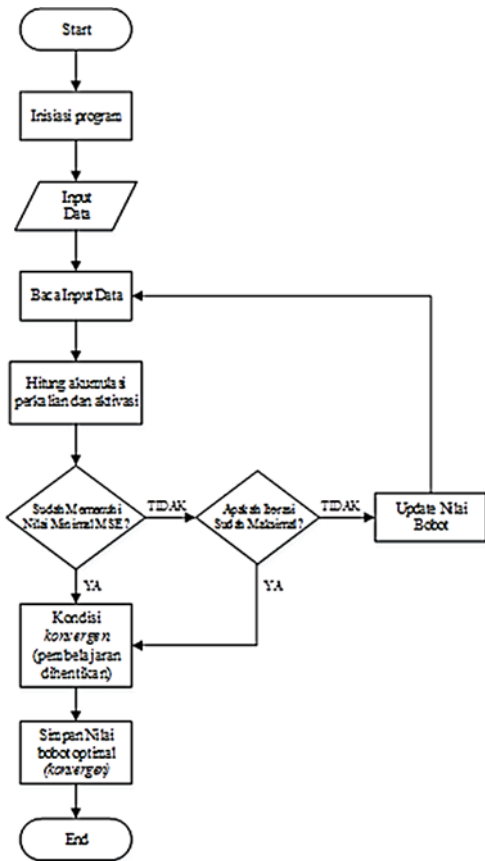
IV. HASIL DAN PEMBAHASAN

A. Proses Learning Jaringan Syaraf Tiruan

Dalam proses pembelajaran dilakukan dengan memasukkan data input yang telah didapat dari proses pembacaan sensor jarak dan kompas beserta data target posisi robot. Pembelajaran dilakukan terhadap 280 sampel pembacaan sensor dengan 10 sampel pada setiap posisi. Proses pembelajaran dilakukan terus-menerus sampai didapat nilai *Mean Square Error (MSE)* lebih kecil dari nilai *error minimum* yang telah ditetapkan yaitu 0.01 atau saat iterasi sudah mencapai maksimal. Ketika nilai *MSE* masih lebih besar dari nilai *error minimum* maka dilakukan proses mundur (*backward*) dari *layer output* ke *layer input* untuk merubah nilai bobot pada tiap *neuron* pada masing-masing layer.

Pada proses *learning* dilakukan 5 kali dengan perbedaan *learning rate* 0.1-0.5 dan iterasi maksimal 100000. Saat program dijalankan pertama sampel data input harus dimasukkan ke dalam program terlebih dahulu, setelah sampel data input dimasukkan bobot dan bias awal harus di inialisasi sebelum menjalankan proses *learning*.

Ketika nilai *MSE* sudah lebih kecil dari nilai *error minimum* atau sudah mencapai iterasi maksimum maka proses pelatihan akan berhenti. Setelah dilakukan proses *learning* didapat nilai bobot yang optimal (*konvergen*) untuk nantinya digunakan dalam proses *running* sistem jaringan syaraf tiruan.



Gambar 6. Flowchart Proses *learning* jaringan syaraf tiruan

Pada Gambar 6 terdapat sistem dari pemetaan ruangan oleh robot dapat jelaskan sebagai berikut:

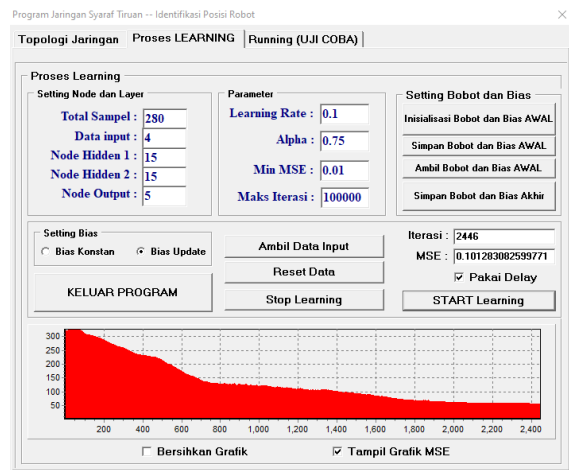
- Input data yang diperoleh dari sensor.
- Setelah data dimasukkan, data akan dibaca oleh JST.
- JST akan memasukkan bobot acak untuk proses *learning*.
- Ketika sudah mencapai nilai minimal *MSE* atau sudah mencapai iterasi maksimal maka akan dihentikan dan nilai bobot optimal (*konvergen*) akan di simpan.
- Nilai bobot yang optimal akan disimpan untuk proses *running* JST.

TABEL 2. *MSE* YANG TERCAPAI PADA PROSES *LEARNING* DENGAN PERBEDAAN *LEARNING RATE*

<i>Learnin g rate</i>	Minima l <i>MSE</i>	Iterasi	<i>MSE</i> tercapai
0.1	0.01	100.00 0	0.037532310391767 6
0.2	0.01	100.00 0	0.055372726216996 3
0.3	0.01	100.00 0	0.071437852524912 8
0.4	0.01	100.00 0	0.055365000334516
0.5	0.01	100.00 0	0.089291205271845 7

Berdasarkan tabel 2 diperoleh *MSE* terkecil adalah 0.037 pada *learning rate* 0.1 dan *MSE* terbesar adalah 0.089 pada *learning rate* 0.5. Terdapat beberapa faktor

yang memungkinkan *error MSE* masih besar dikarenakan data sampel yang terlalu banyak sehingga perlu pembelajaran lebih lama lagi. Kedua, adanya kemiripan pola data antara posisi satu dan lainnya.

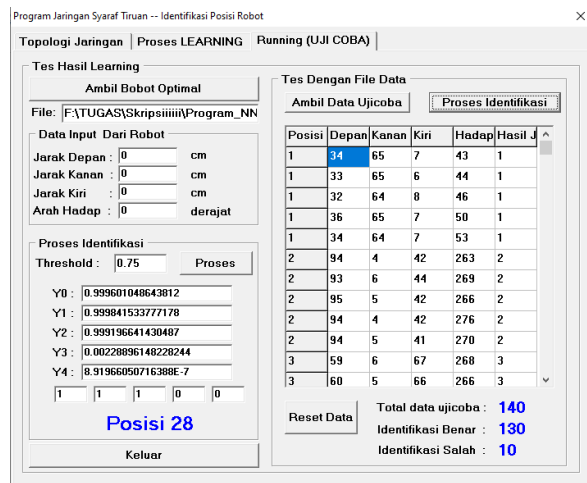


Gambar 7. Program *learning* jaringan syaraf tiruan menggunakan Delphi 7

B. Pengujian Jaringan Syaraf Tiruan

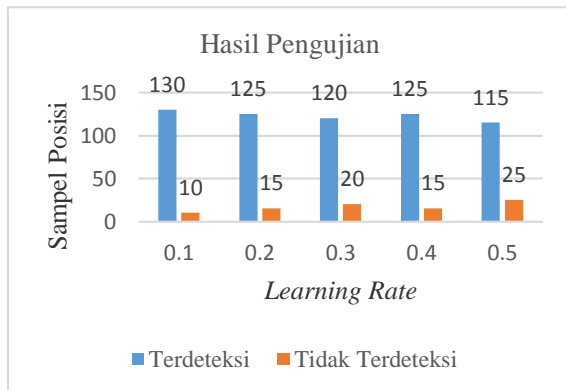
Pengujian jaringan syaraf tiruan akan dilakukan pada aplikasi program Delphi 7 yang telah dibuat sebelumnya, pengujian ini dilakukan untuk memastikan apakah nilai bobot dan bias akhir yang telah disimpan sesuai dengan pola penimbang.

Proses ini dilakukan sebanyak 5 kali pada masing-masing posisi dan dimasukkan input secara acak dari hasil baca sensor jarak dan kompas. Tiap percobaan akan dimasukkan *learning rate* dari 0.1 – 0.5 untuk mendapatkan perbandingan akurasi *learning rate*. Output dari proses jaringan syaraf tiruan berupa data biner 5 bit yang mempresentasikan 28 posisi. Output diberi sebuah *threshold* dimana saat nilai output diatas 0.75 maka dianggap sebagai nilai 1 dan sebaliknya.



Gambar 8. Program pengujian jaringan syaraf tiruan menggunakan Delphi 7

Hasil pengujian JST ditunjukkan pada Gambar 9 dapat disimpulkan dari 5 percobaan dengan *learning rate* berbeda pada hasil pembelajaran yang memiliki akurasi terbaik yaitu pada *learning rate* 0.1 sebesar 92.85%, pada *learning rate* 0.1 dapat mendeteksi 130 dari 140 sampel posisi dan 10 posisi tidak terdeteksi.



Gambar 9. Hasil Pengujian Jaringan Syaraf Tiruan

Kesalahan terbesar terjadi saat membaca posisi 24. Hal ini bisa terjadi dikarenakan pembelajaran yang kurang lama ataupun ada kesamaan data posisi antara posisi 24 dan lainnya.

V. KESIMPULAN

Bisa disimpulkan proses pembelajaran nilai konstanta *learning* mempengaruhi hasil akhir *MSE* di iterasi maksimal jaringan syaraf tiruan. Dari 5 percobaan bisa disimpulkan bahwa bobot optimal dari *learning rate* 0.1 mendapatkan akurasi posisi sebanyak 92.85%, akurasi posisi ini dibidang mendapatkan akurasi terbaik dari 5 *learning rate* yang diujikan. Hasil dari simulasi ini diharapkan kedepannya dapat diterapkan kedalam robot *quadpod* sebenarnya

UCAPAN TERIMAKASIH

Mengucap syukur kepada Allah Swt. sehingga saya dapat menjalankan penelitian skripsi ini dengan lancar. Terimakasih kepada program studi Teknik Elektro Universitas Hang Tuah yang memfasilitasi pengerjaan penelitian ini di workshop robotika.

DAFTAR PUSTAKA

- [1] Yanti, Nur, and Taufik Nur. "Aplikasi Jaringan Syaraf Tiruan Pada Sistem Kontrol Robot Line Follower Dengan Algoritma Backpropagation." *PROSIDING SNITT POLTEKBA* 2.1 (2017): 270-276.
- [2] Siswanto, Agus. "Analisis Algoritma Untuk Mengidentifikasi Ruang pada Map Kontes Robot Pemadam Api Indonesia Menggunakan Logika Fuzzy." *Jurnal Processor* [Online], 14.1 (2019): 1-13.
- [3] Firmansyah, Riza Agung, and Tjahja Odianto. "Simulasi Pengenalan Pola Ruang Menggunakan Jaringan Syaraf Tiruan Pada Aplikasi Sistem Lokalisasi Robot Rescue."

Seminar Nasional Sains dan Teknologi Terapan V 2017 Institut Teknologi Adhi Tama Surabaya. 2017.

- [4] Walingkas, Irene S., Meicsy El Najoran, and Brave A. Sugiarto. "Perpaduan Sensor Ultrasonik Dengan Mini Computer Raspberry Pi Sebagai Pemandu Robot Beroda." *Jurnal Teknik Elektro dan Komputer* 8.3 (2019): 121-132.
- [5] Susanti, Endang, and Mr Hasbi. "Desain Sistem Gerak Robot Quadroped Berbasis Arduino Menggunakan Bluetooth HC-05." *SIGMA TEKNIKA* 2.1 (2019): 20-31.
- [6] Freeman, J. A., and D. M. Skapura. "Neural networks: Algorithms, applications and programming techniques." *JOURNAL-OPERATIONAL RESEARCH SOCIETY* 43 (1992): 1106-1106.
- [7] López, Vivian F., Santo L. Medina, and Juan F. de Paz. "Taranis: Neural networks and intelligent agents in the early warning against floods." *Expert Systems with Applications* 39.11 (2012): 10031-10037.
- [8] Kusumadewi, Sri. "Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link." *Yogyakarta: Graha Ilmu* (2004).

