Model Pembelajaran Off-Line Menggunakan Jaringan Syaraf Tiruan Untuk Pengemudian Otomatis pada Kendaraan Beroda Jurusan Teknik Elektronika PENS 2009

Arie Setya Wulandari[#1], Eru Puspita S.T., M.Kom[#2]

[#]*Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya*
Kampus PENS-ITS Sukolilo, Surabaya
[1]setya@eepis-its.edu
[2]eru@eepis-its.edu

***Abstrak*— Jaringan Syaraf Tiruan adalah pemrosesan suatu informasi yang terinspirasi oleh sistim sel syaraf biologi Jaringan Syaraf Tiruan, seperti manusia, belajar dari suatu contoh. Jaringan Syaraf Tiruan telah dikembangkan sebelum adanya suatu komputer konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun.**

Di jaman yang serba canggih seperti sekarang ini teknologi Jaringan Syaraf Tiruan banyak diterapkan untuk mengontrol pergerakan robot. Dalam Tugas Akhir ini kami membuat aplikasi Jaringan Syaraf Tiruan yang didasarkan pada proses belajar sendiri. Program akan melakukan proses pembelajaran tertentu bagaimana untuk bergerak maju, mundur, ke kiri, ke kanan atau kemungkinan lain berdasarkan pengalaman tabrakan yang terjadi. Diharapkan dari proses pembelajaran ini, program dapat menentukan cara bergerak dengan sendirinya jika mengalami atau menemui halangan-halangan.

Jenis mobile robot yang digunakan adalah tricycle. Tricycle terdiri dari satu buah roda depan dan dua buah roda belakang. Dengan 4 input berupa sensor untrasonic dan output berupa 2 motor. Dimana input berupa sensor ultrasonic tersebut akan diolah pada Jaringan Syaraf Tiruan sehingga menghasilkan error yang dapat digunakan untuk proses backpropagation .

Kata kunci: Jaringan Syaraf Tiruan, Konvensional, Tricycle, sensor Ultrasonic, Backpropagation

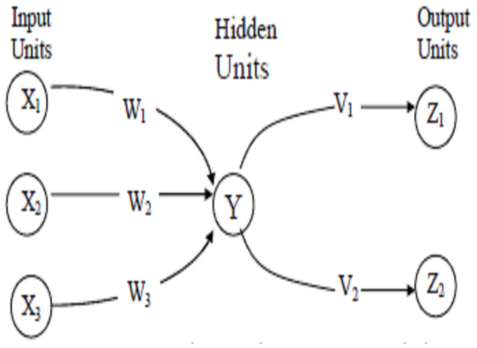
I. PENDAHULUAN

Neural Network atau Jaringan Syaraf Tiruan adalah suatu sistem yang terdiri dari arsitektur jaringan syaraf dan metode pembelajaran. Neural Network secara umum digunakan untuk menirukan sistem kerja otak yang memiliki kemampuan untuk belajar (beradaptasi atau mengikuti perubahan dan belajar atau menerima sesuatu yang baru). Karena itu, Neural Network secara konsep dapat digunakan untuk berbagai permasalahan yang berkaitan dengan belajar. Tentu saja hal ini tidak dapat digunakan secara luas dalam kenyataannya tidak bisa secara penuh ditirukan struktur dan mekanisme kerja dari otak. Untuk satu jenis permasalahan akan memerlukan arsitektur dan metode pembelajaran yang tersendiri. Bahkan untuk satu permasalahan dapat diselesaikan dengan berbagai macam pilihan. Termasuk untuk aplikasi pada sistem kontrol dapat diselesaikan dengan berbagai cara, dan dalam tulisan ini akan disajikan sistem kontrol yang menggunakan Neural Network dengan arsitektur dan metode yang disederhanakan untuk keperluan implementasi berbasis mikrokontroler.

II. TEORI PENUNJANG

A. ARSITEKTUR JARINGAN SYARAF TIRUAN

Arsitektur JST yang sederhana dapat dilihat pada gambar berikut:



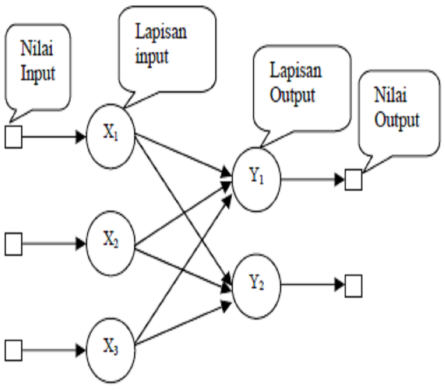
Gambar1 Model Dasar Jaringan Syaraf Tiruan

Gambar di atas merupakan model JST yang sederhana, yang terdiri dari input, output dan satu unit hidden yang terletak pada suatu lapisan tersembunyi (*hidden layer*). Pembagian arsitektur jaringan saraf tiruan bisa dilihat dari kerangka kerja dan skema interkoneksi. Kerangka kerja jaringan saraf tiruan bisa dilihat dari jumlah lapisan (layer) dan jumlah node pada setiap lapisan.

Ada beberapa arsitektur Jaringan Syaraf Tiruan:

1. *Single-layer neural network*

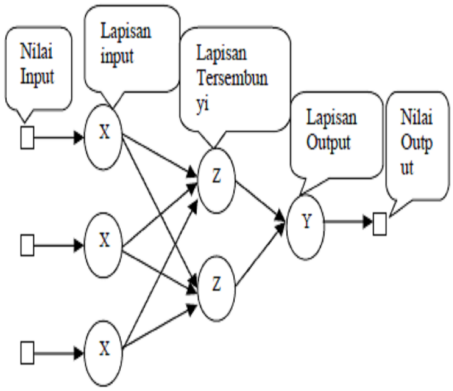
Adalah Sebuah net yang memiliki satu layer dari hubungan pembobot-pembobotnya, diantara unit-unit input dengan unit-unit output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi (Gambar 2.4). Pada Gambar 2.4 tersebut, lapisan input memiliki 3 neuron, yaitu X_1, X_2, X_3. Sedangkan pada lapisan output memiliki 2 neuron yaitu Y_1 dan Y_2. Neuron-neuron pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian. Semua unit input akan dihubungkan dengan setiap unit output.



Gambar2 Single Layer Neural Network

2. *Multilayer neural network*

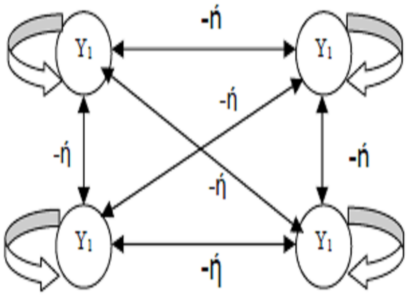
Adalah sebuah net yang terdiri dari banyak lapisan yang terdapat diantara lapisan input dan lapisan output dan juga memiliki 1 atau lebih lapisan tersembunyi. Untuk beberapa kasus, mungkin multilayer lebih menguntungkan, tetapi pada umumnya dengan satu layer saja sudahmemadai untuk menyelesaikan berbagai masalah. Umumnya, ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit dari pada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, pada banyak kasus, pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.



Gambar3 Multilayer Neural Network

3. *Competitive neural network*

Hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Gambar 2.6 menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot –ή.



Gambar4 Competitive Neural Network

B. ALGORITMA PEMBELAJARAN BACKPROPAGATION

Metode Neural Network atau Jaringan Syaraf Tiruan (JST) yang akan digunakan pada proyek akhir ini menggunakan algoritma Backpropagation. Aturan belajar algoritma ini adalah menggunakan
error atau ketidaksesuaian output dengan target untuk koreksi bobotnya. Bobot di koreksi sampai error dapat diterima (memenuhi toleransi yang kita berikan) atau sampai dengan jumlah epoch tertentu.

Prosedur pengajaran atau pembentukan bobot-bobot yang digunakan adalah sebagaimana yang digunakan dalam pengajaran jaringan yang bersifat supervised learning (pengajaran yang menggunakan target). Sehingga aturan ini memerlukan pasangan output untuk tiap input yang akan diajarkan. Dengan keadaan bobot awal random, tiap input dilewatkan ke bobot tersebut dan di hasilkan output untuk saat itu. Output tersebut dibandingkan dengan target yang diinginkan. Besar perbedaan yang terjadi digunakan sebagai faktor pengubah pembobot yang menghubungkan input dengan output tersebut (Update wight). Sehingga dengan bobot yang baru akan mengarahkan output ke target yang seharusnya. Proses perubahan bobot berdasarkan error ini dilakukan terus sampai output yang di hasilkan sesuai dengan yang di targetkan, atau mempunyai error yang dapat diterima.

Tujuannya, yaitu melatih sistem jaringan untuk mencapai suatu keseimbangan dalam merespon secara benar model input yang telah dilatihkan, dan kemampuan untuk memberikan respon yang masuk akal bagi input yang mirip tetapi tidak identik dengan input pada saat pembelajaran. Pada proses pengajaran, diperlukan semua pola data input yang akan diajarkan dan target yang telah di tentukan sebelumnya. Setiap pola yang diinputkan akan diolah dan diproses melalui bobot yang ada, dan hasilnya dibandingkan dengan data target yang diinginkan, kemudian dihitung error-nya (ketidaksamaan hasil saat itu dengan hasil yang diinginkan). Dimana error tersebut diumpan-balikkan (*backpropagation*) kebobot yang menghubungkan layer tersebut sebagai sinyal koreksi bobot, agar dengan bobot yang baru errornya berkurang sampai dengan harga yang diterima.

Prinsip algoritma backpropagation memiliki 3 fase
1. Fase feedforward pada pola input pembelajaran.
2. Fase kalkulasi dan backpropagation error yang didapat.
3. Fase penyesuaian bobot.

Arsitektur yang digunakan adalah jaringan perseptron lapis banyak (*multi layer perseptron*), hal ini merupakan generalisasi dari arsitektur jaringan perseptron lapis tunggal. Secara umum, algoritma jaringan ini memerlukan waktu pembelajaran yang memang lambat, namun setelah pembelajaran selesai, aplikasinya akan memberikan output yang sangat cepat dikarenakan faktor pembobot yang lebih baik.

Algoritma pelatihan untuk jaringan saraf tiruan *Back Propagation* ini adalah sebagai berikut :

Langkah 0 : Inisialisasi nilai bobot dengan nilai acak yang kecil.

Langkah 1 : Selama kondisi berhenti masih tidak terpenuhi, laksanakan langkah 2 sampai 9.

Langkah 2 : Untuk tiap pasangan pelatihan, kerjakan langkah 3 sampai 8.

- *Feedforward :*

Langkah 3 : Untuk tiap unit masukan (X_i, i=1,…,n) menerima sinyal masukan x_i dan menyebarkan sinyal itu keseluruh unit pada lapis atasnya (lapis tersembunyi).

Langkah 4 : Untuk tiap unit tersembunyi (Z_j, j=1,…,p) dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$Z\_in_j = V_{0j} + \sum_{i=1}^{n} X_I V_{ij} \qquad (2.5)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi akti-vasi yang dipilih : $z_j = f(z\_in_j)$ dimana fungsi aktivasi yang digunakan ialah fungsi sigmoid biner yang mempunyai persamaan :

$$f_1(x) = \frac{1}{1+\exp(-z)} \qquad (2.6)$$

Hasil fungsi tersebut dikirim ke semua unit pada lapis di atasnya.

Langkah 5 : Untuk tiap unit keluaran (Y_k, k=1,..,m) dihitung nilai masukan dengan menggunakan nilai bobot-nya :

$$y\_in_k = W_{0k} + \sum_{j=1}^{p} Zj\ Wjk \qquad (2.7)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi :

$$y_k = \frac{1}{1+\exp(-y)} \qquad (2.8)$$

- Perhitungan backward :

Langkah 6 : Untuk tiap unit keluaran (Y_k, k=1,..,m) menerima pola target yang bersesuaian dengan pola masukan, dan kemudian dihitung informasi kesalahan :

$$\delta_k = (t_k - y_k)\, f'(y\_in_k) \qquad (2.9)$$

Kemudian dihitung koreksi nilai bobot yang kemudian akan digunakan untuk memperbaharui nilai bobot w_{jk} :

$$\Delta W_{jk} = \alpha\, \delta_k\, z_j \qquad (2.10)$$

Hitung koreksi nilai bias yang kemudian akan digunakan untuk memperbaharui nilai w_{0k} :

$$\Delta W_{0k} = \alpha \delta_k \qquad (2.11)$$

dan kemudian nilai δ_k dikirim ke unit pada layer sebelumnya.

Langkah 7 : Untuk tiap unit tersembunyi (Z_j, j=1,…,p) dihitung delta masukan yang berasal dari unit pada layer di atasnya :

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k W_{jk} \qquad (2.12)$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung informasi kesalahan :

$$\delta_j = \delta\_in_j\, f'(z\_in_j) \qquad (2.13)$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk mem-perbaharui nilai υ_{ij} :

$$\Delta V_{ij} = \alpha \delta_j X_i \qquad (2.15)$$

dan hitung nilai koreksi bias yang kemudian digunakan untuk memperbaharui υ_{oj} :

$$\Delta V_{0j} = \alpha \delta_j \qquad (2.16)$$

- Memperbaharui nilai bobot dan bias :

Langkah 8 : Tiap nilai bias dan bobot (j=0,…,p) pada unit keluaran (Y_k, k=1,…,m) diperbaharui :

$$W_{jk}\ (new) = W_{jk}\ (old) + \Delta W_{jk}$$

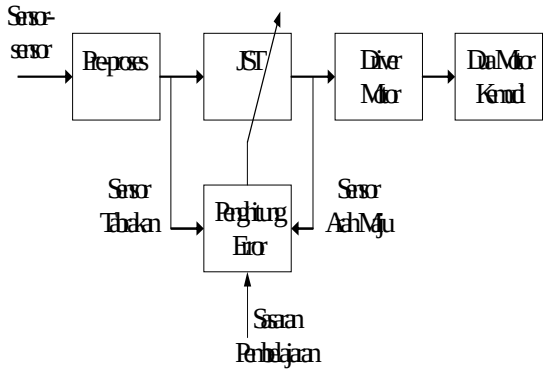
$$V_{ij}\ (new) = V_{ij}\ (old) + \Delta V_{ij} \qquad (2.17)$$

Langkah 9 : Menguji apakah kondisi berhenti sudah terpenuhi. Kondisi berhenti ini

terpenuhi jika nilai kesalahan yang dihasilkan lebih kecil dari nilai kesalahan referensi.

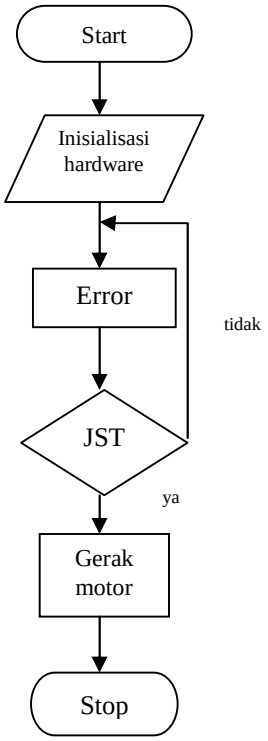
III. PERANCANGAN PROGRAM

A. BLOK DIAGRAM SISTEM



Gambar5 Blok Diagram Sistem

B. FLOWCHART SISTEM

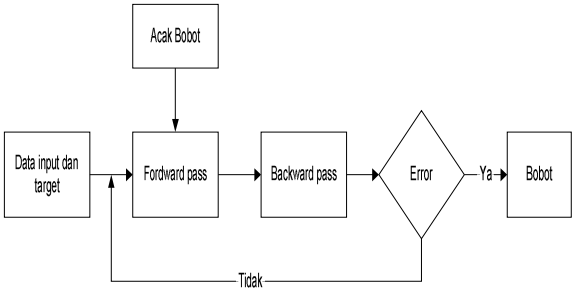


Gambar6 Flowchart Algoritma Keseluruhan Sistem

Cara kerja keseluruhan sistem adalah pertama kali program tidak memiliki kemampuan mengemudi sama sekali, atau mengambil nilai kemampuan yang telah disimpan sebelumnya. Jika program dalam keadaan tidak memiliki kemampuan sama sekali (baru pertama dioperasikan), maka JST akan mengeluarkan tegangan kontrol ke motor secara acak dan tak terkendali. Untuk mencegah hal yang tidak diinginkan, dalam keadaan belum memiliki kemampuan mengemudi, dilengkapi dengan pembatas kecepatan motor. Berdasarkan sensor tabrakan, sensor arah maju dan sasaran pembelajaran yang telah ditentukan (dalam hal ini program diperintahkan untuk menghindari dari tabrakan dan bergerak maju), maka program menghitung nilai error yang terjadi. Setiap terjadi tabrakan, dianggap ada suatu kesalahan. Besarnya nilai error digunakan untuk mempengaruhi JST agar melakukan proses pembelajaran. Selama proses pembelajaran, JST akan mengeluarkan tegangan-tegangan kontrol ke motor yang berubah-ubah dan sulit ditentukan keadaannya. Jika proses pembelajaran berhasil, JST akan mengeluarkan tegangan kontrol ke motor yang menyebabkan motor bergerak untuk menghindari dari tabrakan sambil berusaha maju ke depan.

C. PROSES PEMBELAJARAN

Pada tahap pembelajaran terjadi prosees pengaturan besarnya pembobot dalam jaringan saraf tiruan . Pembelajaran dianggap berhasil jika nilai pembobot mencapai konvergen. Terdapat 3 tahap pembelajaran yaitu feedforward, error dan backpropagation, serta penyesuaian nilai bobot agar mencapai konvergen. Konvergen disini maksudnya dapat dijelaskan pada gambar 3.4 berikut:



Gambar7 Blok diagram yang konvergen dalam pembelajaran

IV. PENGUJIAN DAN ANALISA

Pada bab ini akan dilakukan pengujian dan analisa terhadap beberapa algoritma yang telah dirancang dan dibuat pada bab sebelumnya.

A. PENGUJIAN DENGAN ITERASI

1) Tujuan

Untuk mengatur manuver gerakan robot agar dicapai gerakan yang diinginkan yaitu dapat mendeteksi halangan dengan cepat.

2) Parameter yang digunakan

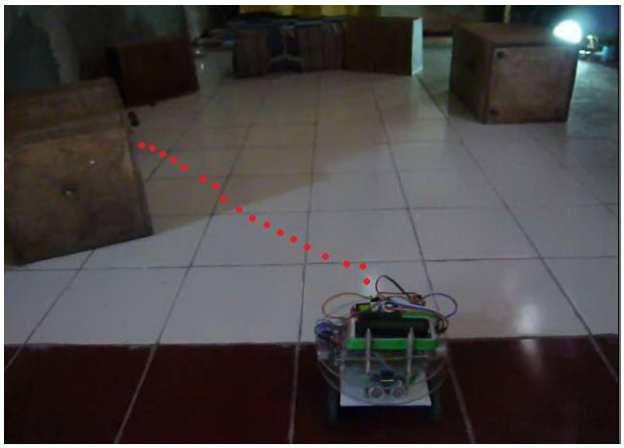
- Input : 4 sensor ultrasonik
- Hidden layer : 3
- Output : 2 motor
- Learning Rate : 1

3) Setting pengujian

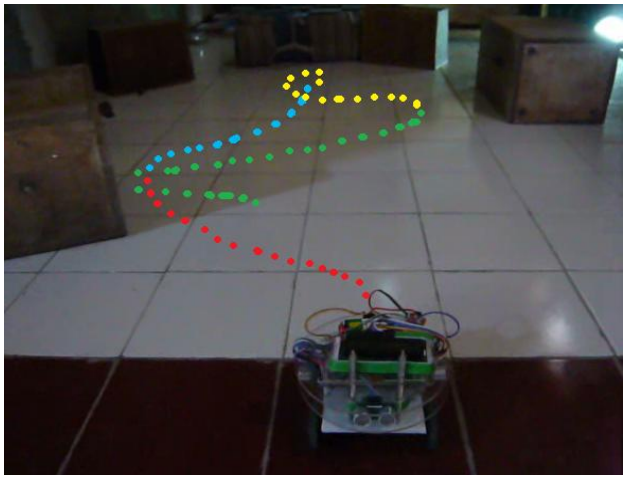
Robot diberi program yang sudah terdapat metode jaringan syaraf tiruan. Dengan mengganti banyak iterasi supaya dihasilkan gerakan yang maksimal (meminimalisasi halangan, agar tidak menabraknya).

4) Pengujian yang dilakukan

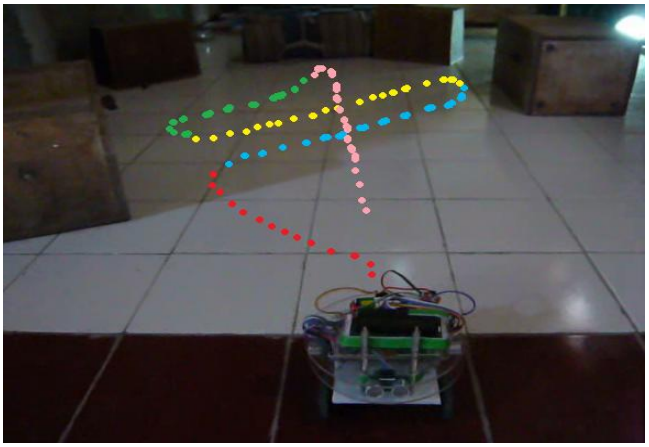
- Dengan iterasi 10 dan 20 hasilnya sebagai berikut:



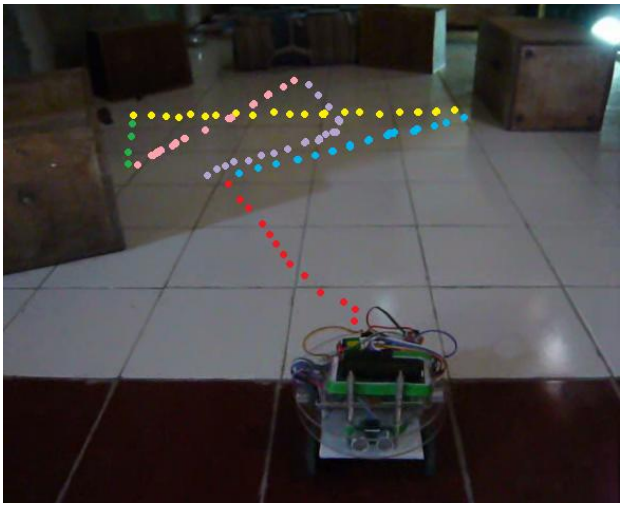
- Dengan iterasi 30 dan 50 hasilnya sebagai berikut:



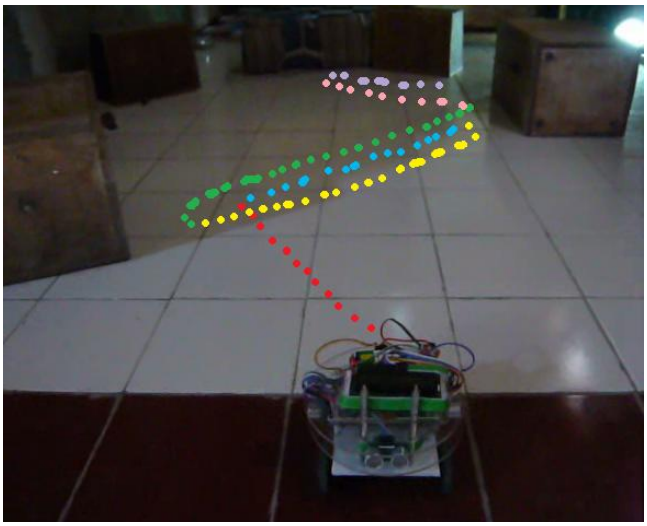
- Dengan iterasi 70 dan 100 hasilnya sebagai berikut:



- Dengan iterasi 500 dan 700 hasilnya sebagai berikut:



- Dengan iterasi 1000 dan 2000 hasilnya sebagai berikut:



5) Analisa

Pertama-tama dilakukan pengujian pada iterasi=10. Pada iterasi ini manuver gerakan robot masih tidak konvergen

artinya manuver gerakannya masih belum sesuai dengan yang diinginkan. Prosentasenya 0%. Hal ini disebabkan karena iterasi yang dilakukan sangat singkat yaitu 10 begitupula dengan iterasi=20. Sedangkan pada iterasi=30 sampai iterasi=100. Pada iterasi ini manuver gerakan sudah seimbang yaitu prosentase 80%. Manuver gerakan yang dihasilkan sudah sesuai dengan yang diinginkan. Misalnya pada saat robot mobile diberi *environment* yang mengalangi setiap sensor depan dan belakang. Maka robot akan bergerak maju dan belok kanan sesuai dengan manuver gerakan yang telah dipelajari pada iterasi sebelumnya. Sedangkan pada iterasi=200 sampai iterasi=2000, meskipun membutuhkan waktu yang lama tetapi iterasi ini menghasilkan manuver yang sempurna sampai mencapai 100%.

Dari pengujian tersebut dapat diketahui bahwa jaringan syaraf tiruan sangat berpengaruh terhadap manuver dari mobile robot apabila diberikan iterasi yang berbeda-beda. Hal ini dilakukan dengan mengubah iterasi-iterasi pada program yang telah dibuat. Semakin banyak iterasi maka semakin sempurna manuver gerakan meskipun semakin lama iterasi maka semakin lama juga jaringan syaraf tiruan melakukan pembelajaran.

B. PENGUJIAN DENGAN MENGGANTI NILAI TIAP NEURON HIDDEN

1) Tujuan

Untuk membandingkan manuver gerakan antara nilai neuron hidden 2,3 dan 4.

2) Parameter yang digunakan

Penggantian neuron hidden	: 2,3 dan 4.
Input	: 4
Output	: 2
Learning rate	: 1
Iterasi	: 50,100,500

3) Setting pengujian

Dengan setting program yang sama, dilakukan penggantian jumlah neuron hidden yaitu 2, 3 dan 4 dengan diberi iterasi masing-masing 50, 100 dan 500. Prosentase manuver antara 0% sampai 100% dengan perincian sebagai berikut:

- 0% sampai 30 % = manuver tidak sempurna masih mengalami tabrakan.
- 40% sampai 60% = manuver belum sempurna pada sensor tertentu.
- 70% sampai 100% = manuver sudah sempurna

4) Pengujian yang dilakukan

- Neuron hidden : 2
- Iterasi : 50, 100 dan 500

Tabel 4.7 Manuver pada neuron hidden 2

Iterasi	Manuver robot	Waktu melakukan iterasi
50	10%	13 detik
100	40%	30 detik
500	50%	2 menit 50 detik

- Neuron hidden : 3
- Iterasi : 50, 100 dan 500

Tabel 4.8 Manuver pada neuron hidden 3

Iterasi	Manuver robot	Waktu melakukan iterasi
50	30%	14detik
100	50%	50 detik
500	80%	3 menit 31 detik

- Neuron hidden : 4
- Iterasi : 50, 100 dan 500

Tabel 4.9 Manuver pada neuron hidden 4

Iterasi	Manuver robot	Waktu melakukan iterasi
50	60%	20detik
100	70%	1 menit 10 detik
500	90%	3 menit 40 detik

5) Analisa

Dari percobaan yang telah dilakukan diketahui bahwa pada saat neuron hidden=2 dan iterasi=50, *mobile robot* masih mengalami tabrakan pada saat *running* awal. Hal ini disebabkan neuron hidden yang diberikan kecil dan juga iterasi yang diberikan juga kecil. Kemudian pada saat iterasi selanjutnya yaitu pada saat iterasi=100, manuver *mobile robot* sudah sempurna pada saat *running* awal tetapi masih mengalami tabrakan pada saat 3 dari 4 sensor mendeteksi halangan. Hal ini juga terjadi pada iterasi=500. Pada saat neuron hidden=3 manuver yang dihasilkan pada tiap iterasi juga hampir sama dengan neuron hidden=2. Kemudian pada neuron hidden=4,iterasi=500 manuver yang dihasilkan sudah mendekati sempurna hanya saja pada saat halangan terletak lebih tinggi atau lebih rendah daripada letak sensor ultrasonik pada *mobile robot*, sensor tidak mendeteksi adanya halangan.

Pada tabel tersebut juga diketahui bahwa semakin besar neuron hidden yang diberikan pada program maka semakin lama waktu iterasi yang dibutuhkan oleh *mobile robot.*

C. PENGUJIAN DENGAN MENGGANTI NILAI TIAP NEURON HIDDEN

1) Tujuan

Untuk membandingkan manuver *mobile robot* antara learning rate 0,5 kemudian 0,9 dan 1.

2) Parameter yang digunakan

Merubah learning rate	: 0.5; 0,9 dan 1
Input	: 4
Output	: 2
Neuron hidden	: 3
Iterasi	: 50,100,500

3) Setting pengujian

Dengan setting program yang sama, tetapi mengganti nilai learning rate 0,5 kemudian 0,9 dan 1 dengan prosentase manuver yaitu 0% sampai 100%. Perinciannya sebagai berikut:

- 0% sampai 30 % = manuver tidak sempurna masih mengalami tabrakan.
- 40% sampai 60% = manuver belum sempurna pada sensor tertentu.
- 70% sampai 100% = manuver sudah sempurna

4) Pengujian yang dilakukan

- Dengan learning rate 0,5 hasilnya sebagai berikut:

Tabel 4.10 Manuver pada learning rate 0,5

Iterasi	Manuver robot	Waktu melakukan iterasi
50	40%	16 detik
100	50%	45 detik
500	60%	3 menit 40 detik

- Dengan learning rate 0,9 hasilnya sebagai berikut:

Tabel 4.11 Manuver pada learning rate 0,9

Iterasi	Manuver robot	Waktu melakukan iterasi
50	40%	25 detik
100	60%	45 detik
500	80%	3 menit 50 detik

- Dengan learning rate 1 hasilnya sebagai berikut:

Tabel 4.12 Manuver pada learning rate 1

Iterasi	Manuver robot	Waktu melakukan iterasi
50	50%	30 detik
100	70%	1 menit
500	90%	4 menit 20 detik

5) Analisa

Dari percobaan dan simulasi yang telah dilakukan, dapat diketahui bahwa manuver pada saat learning rate=0,5 manuver tidak sempurna seperti pada saat iterasi 50. Dikarenakan pada saat iterasi tersebut sensor depan dan belakang kurang sempurna mendeteksi halangan sehingga *mobile robot* sering menabrak halangan. Kemudian pada saat learning rate=0,9; iterasi=100, manuver sudah hampir sempurna mendeteksi halangan sedangkan pada iterasi=500 *mobile robot* masih belum bisa mendeteksi halangan dengan cepat pada saat sensor depan bernilai 1 akan tetapi *mobile robot* dapat bergerak mundur dengan cepat. Saat learning rate=1, merupakan manuver robot paling optimal yaitu hampir mendekati 100% meskipun pada iterasi 50 robot masih menabrak halangan dan berhenti. Hal ini karena tidak didukung oleh nilai iterasi yang banyak.

Di samping itu nilai learning rate yang banyak jugaberpengaruh pada waktu pembelajaran *mobile robot.* Seperti dapat dilihat pada masing-masing learning rate yaitu 0.5;0.9 dan 1.

V. KESIMPULAN

Setelah melakukan tahap perancangan dan pembuatan sistem yang kemudian dilanjutkan dengan tahap pengujian dan analisa maka dapat diambil kesimpulan sebagai berikut :

1. Proses pengenalan secara offline dengan menggunakan tabel referensi memberikan pengaturan terhadap gerakan robot supaya bisa bergerak ke kanan, kiri, maju dan mundur.
2. Semakin banyak iterasi yang dilakukan pada pembelajaran jaringan syaraf tiruan, maka semakin baik manuver gerakan yang dihasilkan.
3. Simulasi terbaik yang dilakukan terdiri dari 4 input, 4 neuron hidden, 2 output, 1 learning rate dan iterasi lebih besar dari 500.
4. Dengan menggunakan metode jaringan syaraf tiruan dengan diberikan iterasi yang besar maka manuver robot akan lebih *smooth* dan dapat mendeteksi halangan dengan cepat.

DAFTAR PUSTAKA

[1]. Membangun Jaringan Syaraf Tiruan, Sri Kusumadewi, 2004, Graha Ilmu, Yogyakarta.

[2]. Andri Kristanto, 2004, *Jaringan Syaaf Tiruan: Konsep Dasar, Algoritma, dan Apilkasi*, Gava Media, Yogyakarta.

[3]. Diah Puspitaningrum, 2006, *Pengantar Jaringan Saraf Tiruan*, Andi, Yogyakarta.

[4]. Introduction to Neural Network by K. Gurney. http://www.shef.ac.uk/psychology/gurney/notes/contents.html

[5]. Neural Network by Christos Stergiou and Dimitrios Siganos, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report

[6]. Neural Network by Nikolay Nikolaef http://homepages.gold.ac.uk/nikolaef/cis311.html.course_outline_for_fall_2004.

[7]. Subiyanto, Pemakaian Jaringan Syaraf Tiruan Perambatan-Balik Sebagai Cara Lain Prakiraan Beban Jangka Pendek Di Jawa Tengah-D.I.Y.. Tugas Akhir Teknik Elektro Fakultas Teknik UNDIP, Semarang, 1998.

[8]. Valluru B. Rao and Hayagriva V. Rao, C++ Neural Network And Fuzzy Logic, Miss: Press, New York, 1993.