

Sistem Pengemudian Otomatis pada Kendaraan Berroda dengan Model Pembelajaran On-line Menggunakan NN

Eru Puspita
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Kampus ITS Keputih Sukolilo Surabaya 60111, Indonesia
email: eru@eepis-its.edu

Abstrak

Cara pengemudian suatu kendaraan otomatis selama ini menggunakan teknik algoritma yang sudah ditentukan dalam program. Pada penelitian ini akan dicoba untuk membuat sistem pengendalian menggunakan NN yang didasarkan pada proses belajar sendiri dan terus menerus selama kendaraan tersebut bergerak. Program (NN) akan melakukan proses pembelajaran tertentu bagaimana untuk bergerak maju, mundur, ke kiri, ke kanan atau kemungkinan lain berdasarkan pengalaman tabrakan yang terjadi. Dari hasil pengujian didapatkan 82% berhasil berjalan maju dan sekitar 90% berhasil menghindari tabrakan dengan dinding.

1. Pendahuluan

Pada sistem kendali gerakan mobile robot yang digunakan dewasa ini banyak yang menggunakan pengaturan gerakan secara langsung berdasarkan informasi dari sensor dan dari sasaran gerakan yang diinginkan. Seluruh proses pergerakan dari motor-motor penggeraknya diatur sepenuhnya oleh program atau hardware. Dengan cara ini maka tingkat kepastian gerakan akan sesuai dengan yang diinginkan.

Pada penelitian ini akan dicoba untuk membangun sistem kendali gerakan berdasarkan proses belajar dari sistem itu sendiri menggunakan NN sebagai pengendali gerakannya. Hal ini didasari oleh NN yang memiliki kemampuan untuk belajar, sehingga pada penelitian ini ingin dibuktikan apakah NN benar-benar dapat belajar untuk mengendalikan gerakan dari mobile robot berdasarkan informasi dari sensor-sensor yang ada.

Ada beberapa cara yang dapat untuk mengendalikan gerakan mobile robot menggunakan NN, antara lain

- a. Pengendalian secara off-line, dimana NN diberikan pelatihan terlebih dulu bagaimana cara untuk bergerak maju dan menghindari halangan. Cara ini memungkinkan NN untuk tahu lebih dulu cara bergerak sebelum diujikan pada lapangan.

- b. Pengendalian berdasar pemberian referensi gerak yang dilakukan secara on-line. Cara ini mirip dengan pengendalian konvensional, namun sifatnya NN hanya diberikan perintah untuk bergerak ke mana tanpa memberitahukan cara Bergeraknya.
- c. Pengendalian berdasarkan menunjukkan bagian yang salah (error) dari sistem mobile secara on-line. Misalkan menunjukkan roda mana yang salah bergerak. Cara ini mirip dengan cara b, namun NN tidak secara langsung diberi perintah harus bergerak ke mana.
- d. Pengendalian berdasarkan hanya menunjukkan kalau ada yang salah (error), tanpa menunjukkan apa atau dimana yang salah. Cara ini lebih sulit, karena mengharuskan NN untuk mencari kemungkinan-kemungkinan bagian mana yang salah.

Pada penelitian ini akan dicoba dua cara, seperti pada point c dan d. Di makalah ini masih diuji cara pada point c.

1.1. Permasalahan dan Batasan Masalah

- Bagaimana NN dapat digunakan untuk mengendalikan gerakan dari mobile robot berdasarkan informasi dari sensor.
- Bagaimana caranya NN dapat belajar dengan sendirinya secara on-line berdasarkan informasi kesalahan yang diterima
- Penelitian hanya ditujukan agar mobile robot dapat melaju ke depan dengan berusaha menghindari halangan yang ada.
- Pada penelitian awal ini hanya diujikan pemodelan sistem kendalinya secara simulasi

1.2. NN

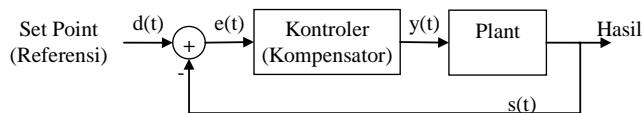
Neural Network atau Jaringan Syaraf Tiruan adalah suatu sistem yang terdiri dari arsitektur jaringan syaraf dan metode pembelajaran. NN secara umum digunakan untuk menirukan sistem kerja otak yang

memiliki kemampuan untuk belajar (beradaptasi atau mengikuti perubahan dan belajar atau menerima sesuatu yang baru). Karena itu, NN secara konsep dapat digunakan untuk berbagai permasalahan yang berkaitan dengan belajar. Tentu saja hal ini tidak dapat digunakan secara luas dalam kenyataannya, mengingat tidak bisa secara penuh ditirukan struktur dan mekanisme kerja dari otak.

Dalam kenyataannya, 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 NN dengan arsitektur dan metode yang disederhanakan untuk keperluan implementasi berbasis mikrokontroler.

1.3. Dasar Sistem Kontrol

Sistem kontrol secara umum dianggap sebagai sistem kontrol otomatis yang digambarkan sebagai kontrol tertutup, dimana diperlukan pembacaan keluaran dari plant untuk keperluan kontrol otomatis.



Gambar 1.1 Dasar sistem kontrol

1.4. Kontroler Menggunakan NN

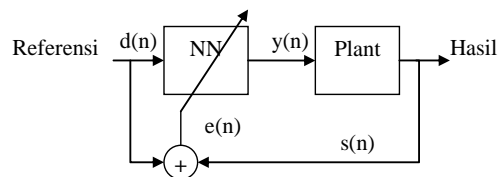
NN dipilih sebagai kontroler disebabkan NN memiliki kemampuan belajar yang dapat digunakan selain mempelajari karakteristik dari sistem kontrol juga untuk beradaptasi terhadap perubahan-perubahan yang terjadi.

Konsep dasar dari sistem kontrol menggunakan NN adalah NN dilatih untuk beradaptasi terhadap karakteristik dari sistem kontrol (khususnya karakteristik plant) dengan cara memaksa NN untuk mengeluarkan keluaran dari plant yang sesuai dengan referensi yang diberikan.

Cara kerja secara umum dari kontroler NN ini dapat diidentikkan dengan kontroler PID, namun memiliki kemampuan mengubah parameter-parameter dari NN sehingga sesuai dengan yang diinginkan.

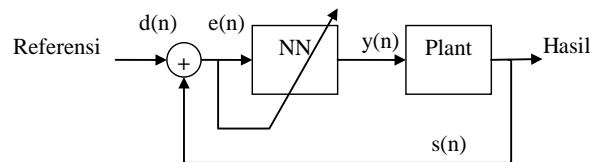
Ada berbagai macam bentuk arsitektur dari NN yang dapat digunakan sebagai kontroler. Contoh yang pertama adalah menggunakan referensi sebagai masukan dari NN. Cara ini memungkinkan NN untuk mempelajari bagaimana mengeluarkan sinyal $y(n)$ yang sesuai dengan sinyal referensi $d(n)$.

Catatan : Variable yang digunakan pada NN adalah indeks n , bukan waktu t , dikarenakan NN umumnya dibuat dalam sistem digital.



Gambar 1.2. Masukan NN dalam bentuk referensi

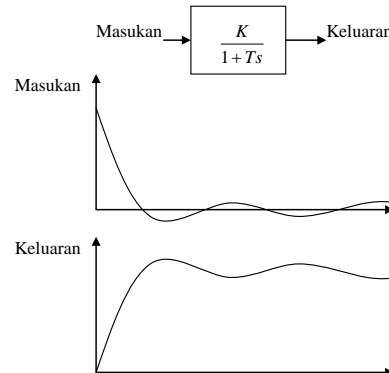
Contoh ke dua adalah menggunakan error sebagai masukan dari NN. Cara ini menjadikan NN tidak mengetahui berapa nilai referensi yang diberikan. NN hanya mengetahui kesalahan-kesalahan yang terjadi dan mengeluarkan sinyal $y(n)$ yang sesuai.



Gambar 1.3. Masukan NN dalam bentuk error

Mengingat kebanyakan plant memiliki orde setidaknya 1 dan karakteristik dasar dalam bentuk integral ($1/s$), maka bentuk masukan NN dalam bentuk error yang lebih sesuai.

Suatu plant dengan karakteristik integral hanya memerlukan masukan untuk menurunkan dan menaikkan nilai keluarannya, bukan harus memerlukan masukan tertentu untuk menentukan nilai keluaran tertentu.

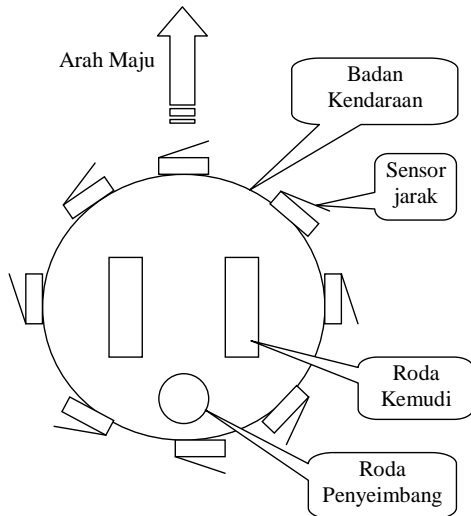


Gambar 1.4. Sistem dengan karakteristik integral

Dari gambar bentuk masukan dari sistem dengan sifat integrasi, dapat dilihat bahwa bentuknya sama dengan error yang dibangkitkan oleh sistem kontrol. Dengan alasan ini, maka kontroler NN dengan masukan error lebih sesuai.

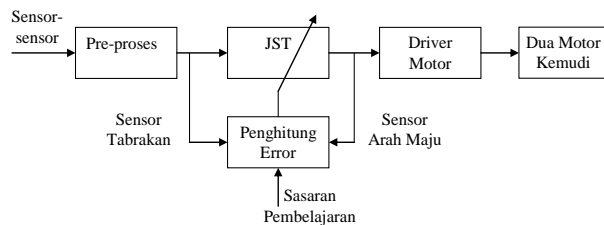
2. Konfigurasi Sistem

Secara umum konfigurasi yang diinginkan adalah seperti pada gambar 2.1. dan gambar 2.2.



Gambar 2.1. Konfigurasi mekanik robot

Gambar 2.1. menjelaskan bentuk mekanik robot dan gambar 2.2. menunjukkan konfigurasi sistem menggunakan NN dan proses pembelajaran secara on-line.



Gambar 2.2. Konfigurasi sistem

Cara Kerja

- 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),

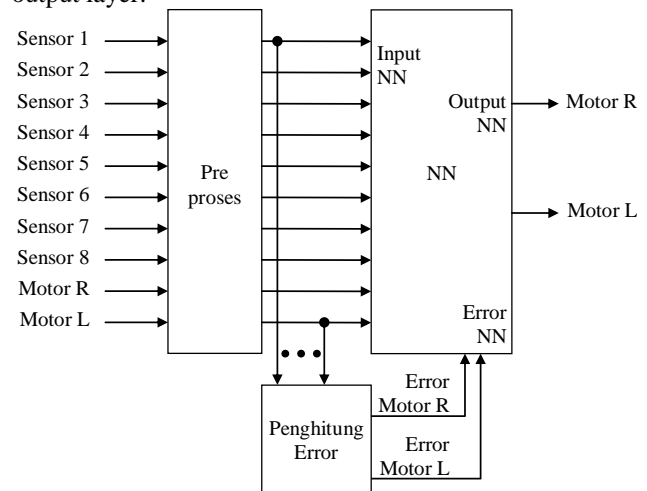
maka program menghitung nilai error yang terjadi.

- 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.

2.1. Arsitektur NN untuk kontroler

Dalam percobaan ini digunakan NN dengan konfigurasi 10 neuron pada input layer, beberapa (10 sampai 20) neuron pada hidden layer dan 2 neuron pada output layer.



Gambar 2.3. Arsitektur NN untuk sistem kontrol

2.2. On-line Update

Umumnya NN dilatih dengan cara diberikan contoh-contoh masukan dan keluaran yang seharusnya dikeluarkan NN (mode supervised). Jika antara keluaran NN dan keluaran yang diinginkan tidak sesuai, maka akan dihitung error yang nantinya error ini digunakan sebagai proses update. Jika dilakukan secara terus menerus, maka NN akan dapat menyesuaikan keluarannya sesuai dengan yang diinginkan. Proses ini dikenal dengan nama pelatihan (training) dan biasanya dilakukan terlebih dahulu sebelum NN digunakan.

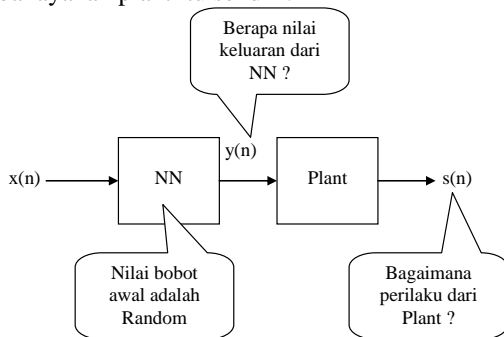
Tujuan dari cara tersebut adalah mempersiapkan agar NN memiliki karakteristik sesuai dengan yang diinginkan sebelum NN digunakan. Cara ini akan menjamin NN tidak akan memiliki perilaku atau

karakteristik yang tidak terprediksi yang dapat membahayakan sistem. Namun jika dapat dipastikan sistem akan tolerir atau perilaku tidak terduga dari NN tidak akan membahayakan sistem, maka proses pembelajaran atau pelatihan NN dapat dilakukan sekaligus saat NN digunakan. Cara ini disebut sebagai On-line Training atau Learning atau Update.

Jika memungkinkan, maka cara ini akan lebih praktis, karena proses dapat dilakukan sekali jalan, sambil jalan sambil belajar. Selain itu dimungkinkan melakukan adaptasi (proses untuk sedikit belajar mengikuti perubahan karakteristik sistem) secara langsung, yang tidak dapat dilakukan pada cara off-line (pelatihan yang terpisah).

2.3. Sistem Pengaman Keluaran NN

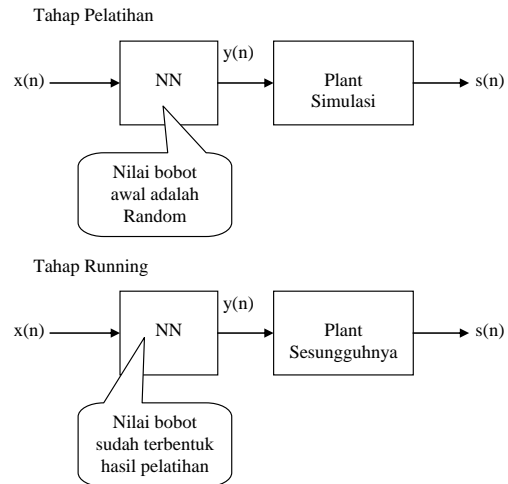
Dalam keadaan awal NN bekerja, umumnya bobot-bobot NN ditentukan secara acak. Jika demikian, apabila masukan NN diberikan nilai tertentu, maka berapakah nilai keluaran dari NN? Jawaban pastinya adalah tidak dapat diketahui atau acak. Dalam keadaan seperti ini, jika NN digunakan untuk menggerakkan suatu plant secara langsung, maka dapat berakibat timbulnya hal-hal yang tidak dapat ditentukan yang dapat membahayakan plant itu sendiri.



Gambar 2.4. Ilustrasi keadaan awal NN

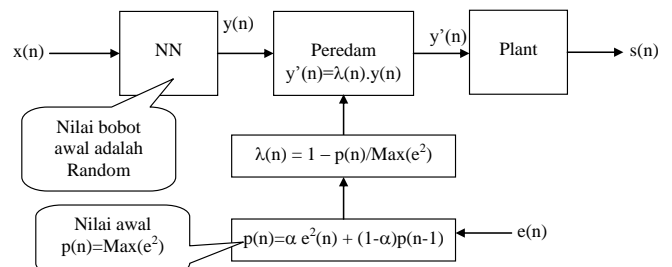
Ada dua cara yang dapat dilakukan,

1. Menggunakan pembelajaran awal (off-line training), dimana NN diberikan plant dalam bentuk miniatur atau simulasi, sehingga keadaan awal dari NN yang dapat membahayakan plant dapat dihindari. Namun cara ini tidak praktis, karena harus menggunakan dua tahap operasional, training dan running.



Gambar 2.5. Teknik mengatur keluaran dari NN menggunakan pelatihan awal

2. Menggunakan sub-sistem pengereman, dimana sub-sistem ini akan meredam keluaran dari NN jika NN masih memiliki rata-rata error yang besar. Nilai peredaman bergantung dengan besar error dari NN. Dalam keadaan awal, nilai error dapat dianggap besar, sehingga peredaman ini dapat mencegah perilaku awal dari NN agar tidak langsung masuk ke plant.



Gambar 2.6. Teknik mengatur keluaran dari NN menggunakan pengereman

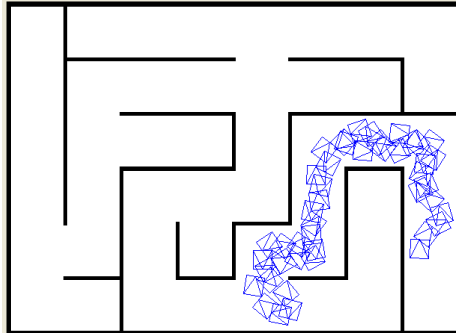
3. Pengujian

Berikut ini diberikan screen shoot pengujian NN secara simulasi menggunakan komputer, untuk memberikan gambaran kemampuan NN dalam menentukan pergerakan mobile robot hanya berdasar informasi kesalahan yang diberikan.

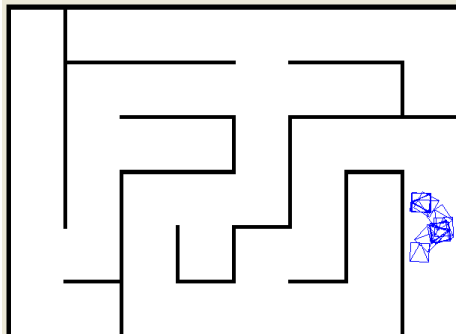
Pada pengujian diberikan arena dalam bentuk maze, dimana tugas utama dari NN adalah dia harus dapat menggerakkan mobile robot berjalan di sepanjang lorong-lorong yang ada tanpa harus menabrak dinding pembatas.

Pengujian dibagi tiga tahap, tahap pertama adalah tahap awal pembelajaran NN, dimana nilai bobot dari NN masih acak. Pada pengujian ini akan didapatkan berbagai macam hasil yang tidak dapat dipikirkan.

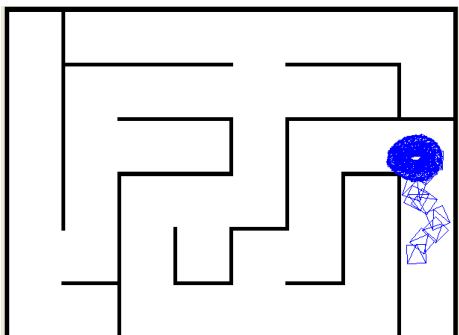
Dari gambar 3.1. sampai 3.4. dapat dilihat berbagai kemungkinan hasil pengujian saat awal pembelajaran. Pada gambar 3.2. dan 3.4. mobile robot terjebak dalam jalur yang sempit, sehingga hanya berputar-putar pada tempat tersebut. Pada gambar 3.1. dan 3.4. mobile robot berhasil mencapai jarak tempuh cukup jauh.



Gambar 3.1. Pembelajaran awal

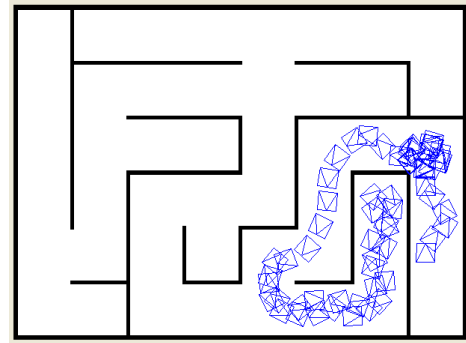


Gambar 3.2. Pembelajaran awal, tidak dapat bergerak

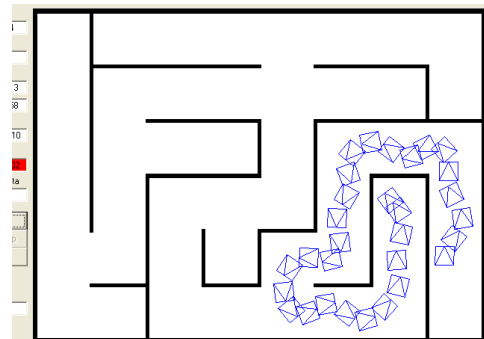


Gambar 3.3. Pembelajaran awal, berputar di satu lokasi

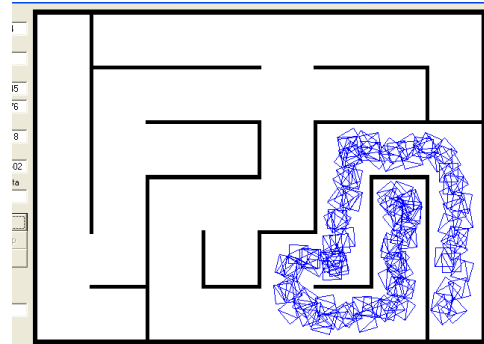
Pengujian kedua dilakukan saat iterasi mencapai lebih dari 10000 kali seperti pada gambar 3.6. dan posisi mobile robot dikembalikan ke posisi awal. Pada gambar tersebut terlihat, bahwa NN sudah tidak mengalami banyak kesulitan saat bergerak. Jika pengujian diteruskan sampai ribuan iterasi berikutnya, ternyata mobile robot hanya berputar-putar pada lokasi yang telah dikenal tersebut, tanpa bisa mencapai tempat lain (seperti pada gambar 3.6.).



Gambar 3.4. Pembelajaran awal, berhasil mencapai satu lokasi



Gambar 3.5. Setelah iterasi lebih dari 10000, posisi dimulai dari awal

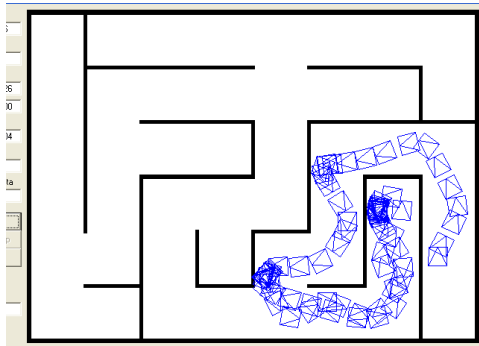


Gambar 3.6. Berputar di sekitar lokasi yang dikenali

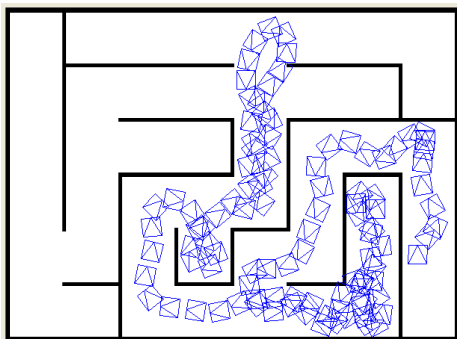
Bahkan setelah mencapai puluhan ribu iterasi, mobile robot hanya dapat berputar pada lokasi yang sama. Pada pengujian ketiga, dicoba untuk mengubah beberapa parameter dari robot, antara lain target kecepatan menjadi 90% dari kecepatan tertinggi (gambar 3.7.). Memperbesar nilai error kecepatan 1,5 kali dari nilai semula (lebih besar 50%), seperti gambar 3.8. Menambah kecepatan mobile robot sebesar 2 kali dan 4 kali dari nilai semula, seperti pada gambar 3.9., 3.10., 3.11.

Penambahan kecepatan pada bagian akhir pengujian ternyata membawa dampak mobile robot mampu untuk menjangkau daerah-daerah lain yang lebih jauh. Namun yang perlu diingat, penambahan kecepatan

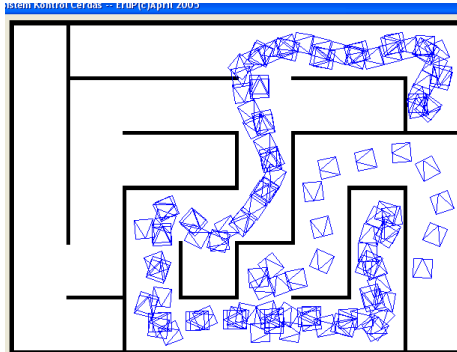
ini hanya dapat dilakukan jika NN sebelumnya sudah belajar.



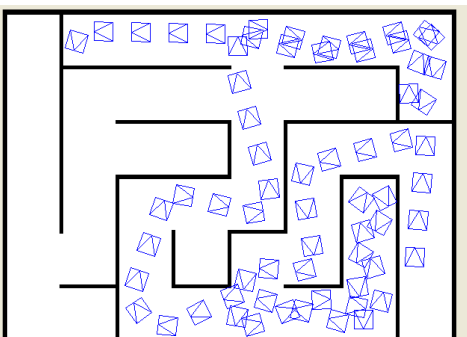
Gambar 3.7. Kecepatan maju ditingkatkan menjadi 90%



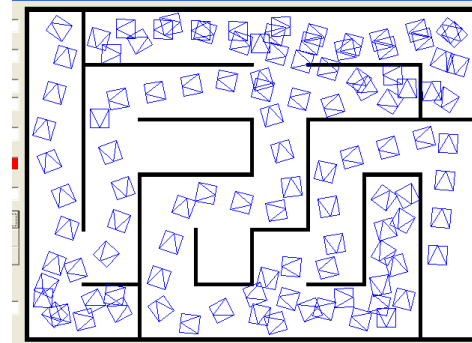
Gambar 3.8. Error kecepatan diperbesar 50%



Gambar 3.9. Menambah kecepatan motor 2 x



Gambar 3.10. Menambah kecepatan 4 x



Gambar 3.11. Hasil akhir pada kecepatan 4x

4. Kesimpulan

- Secara umum, jika target pengujian adalah mobile robot dapat bergerak terus ke depan, maka ditemukan 2 dari 11 pengujian, dimana mobile robot terjebak berputar-putar pada satu titik sempit.
- Pengamatan terhadap kemungkinan lebih dari 200 rawan tabrakan, didapatkan sekitar 20 kali robot menyentuh dinding pembatas, bahkan ditemukan mobile robot menerobos dinding pada pengujian kecepatan motor yang lebih tinggi.
- Asalkan yang diinginkan adalah kemampuan bergerak secara mandiri, metode yang digunakan dalam penelitian ini dapat dimanfaatkan.

5. Referensi

- [1] Kreyzig, Erwin, "Advanced Engineering Mathematics", John Wiley and Sons, Inc., New York, 1993
- [2] Bellanger, Maurice G., "Adaptive Digital Filters and Signal Analysis", Marcel Dekker, Inc., New York and Basle, 1987.
- [3] Betts, J. A., "Signal Processing, Modulation and Noise", The English Universities Press Limited, London, 1970.
- [4] Drapper, N.R., and H. Smith, "Applied Regression Analysis", Second Edition, John Wiley & Sons, New York, 1981.
- [5] Freeman, James A. and David M. Skapura, "Neural Networks, Algorithms, Applications, and Programming Techniques", Addison-Wasley, New York, 1992.