

PEMETAAN POSISI DAN ORIENTASI KURSI RODA CERDAS BERBASIS PRINSIP *DEAD RECKONING*

¹Iwan Setiawan, Isnan Fauzan Akrom, Darjat

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jln. Prof. Sudharto, S.H., Tembalang, Semarang, Jawa Tengah, Indonesia
E-mail: ¹setiaone.iwan@gmail.com

Abstract. Smart wheelchairs is a kind of assistive technology implementation from mobile robot field. The main purpose of the system is enhance disable people mobility and guaranteeing safety aspect while wheelchair navigate around domestic environment. As a kind of semi autonomous smart navigation system, the smart wheelchair must be accommodate share control between automatic controller and the user (via joystick). The ultimate goal of this research is to know exactly the position and orientation of smart wheelchair using dead reckoning principle. To fulfill the goal, the smart wheelchair is equipt by quadrature rotary encoder for each wheels. From the experiments that we drawn, we have get some result: Position and orientation error will be larger for long duration of operation of wheelchairs. Fundamentally, these kinds of error result from systematic and non systematic error source.

Keywords: encoder, dead reckoning, state chart, komunikasi serial, smart wheelchair

Kursi Roda Cerdas atau SWC (*Smart Wheel Chair*) sudah banyak digunakan baik secara komersil maupun nonkomersil di masyarakat. Penanaman kecerdasan didalam kursi roda bertujuan untuk memudahkan penggunaan dan menambah kenyamanan. Perkembangan otomatisasi pada SWC semakin pesat seiring dengan perkembangan ilmu electrical engineering. Kecanggihannya akan bertambah jika SWC memiliki sistem navigasi yang mampu mengetahui posisi, orientasi dan lintasan SWC yang sudah dilalui. Jika SWC yang memiliki sistem navigasi tersebut diintegrasikan dengan laptop maka pengguna SWC dapat mengetahui mengetahui posisi, orientasi dan lintasan SWC secara visual.

Penelitian ini bertujuan memetakan posisi kursi roda dalam koordinat kartesian dan orientasi kursi roda dalam satuan derajat sehingga pengguna dapat mengamati posisi, orientasi, trajektori dan gerakan kursi roda secara visual melalui laptop.

PRINSIP *DEAD RECKONING* PADA ROBOT MOBIL PENGGERAK DIFFERENSIAL

Dead Reckoning adalah proses estimasi posisi saat ini berdasarkan pengukuran kecepatan saat ini pada rentang waktu tertentu dan posisi sebelumnya. Karena sifat perhitungannya yang integratif, maka navigasi

menggunakan metode ini memiliki kerugian dengan penambahan *error* secara kumulatif tiap proses perhitungan, sehingga semakin besar jarak yang ditempuh robot mobil, keakuratan nilai posisi yang dihasilkan akan semakin kecil. Keuntungan dari metode ini adalah kesederhanaan dalam perancangan karena hanya membutuhkan sensor inersial yang berupa sensor penghitung kecepatan atau jarak tempuh. Untuk mengimplementasikan prinsip *dead reckoning* pada navigasi robot mobil penggerak diferensial, maka perlu dipahami juga model kinematika robot mobil penggerak diferensial.

Kursi roda tipe penggerak diferensial memiliki 2 buah roda penggerak yang terpisah (kanan dan kiri). Kedua roda ini digerakkan oleh motor DC yang ditempatkan pada satu sumbu secara terpisah. Sehingga kedua roda ini berfungsi sebagai penggerak sekaligus sebagai kemudi kursi roda. Sebagai penyeimbang umumnya kursi roda ini dilengkapi juga dengan satu atau dua buah roda *castor* yang ditempatkan dibagian belakang kursi roda tersebut.

Untuk aplikasi *dead reckoning* praktis, posisi dan orientasi kursi roda dapat didekati secara matematis oleh persamaan-persamaan berikut ini.

$$S_L(t) = v_L(t).t \quad (1)$$

$$S_R(t) = v_R(t)t \quad (2)$$

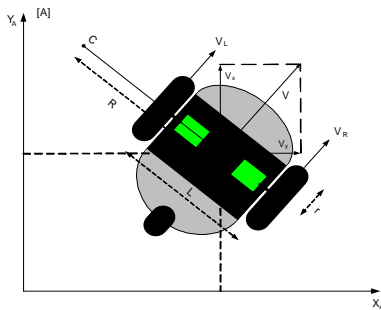
$$S(t) = \frac{v_L(t) + v_R(t)}{2} .t \quad (3)$$

$$\theta(t) = \frac{S_R(t) - S_L(t)}{L} + \theta_0 \quad (4)$$

$$x(t) = x_0 + S(t). \cos \theta(t) \quad (5)$$

$$y(t) = y_0 + S(t). \sin \theta(t) \quad (6)$$

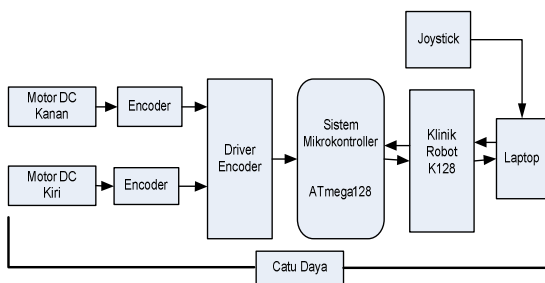
dengan $S_L(t)$, $S_R(t)$, $S(t)$, $\theta(t)$, $x(t)$, $y(t)$, $v_L(t)$, $v_R(t)$, L , θ_0 , x_0 , y_0 dan t secara berturut-turut adalah jarak linier roda kiri, jarak linier roda kanan, jarak tempuh kursi roda, sudut orientasi kursi roda, posisi kordinat x kursi roda, posisi kordinat y kursi roda, kecepatan linier roda kiri, kecepatan linier roda kanan, jarak antara pusat roda kanan dan pusat roda kiri, sudut orientasi awal, posisi kordinat x awal, posisi kordinat y awal, dan waktu *sampling*.



Gambar 1. Posisi dan Orientasi *Mobile Robot* dalam Sistem Koordinat Cartesien.

PERANCANGAN SISTEM Perancangan Perangkat Keras

Perangkat keras dari Penelitian ini meliputi sistem minimum mikrokontroler Atmega128, enkoder Autronics™ E30S4, Schmitt Trigger, dan laptop. Secara umum perancangan sistem yang akan dibuat dapat dilihat pada Gambar 3.1.



Gambar 2. Blok diagram sistem.

Perancangan Perangkat Lunak pada Mikrokontroler

Penelitian ini merupakan pengembangan atau bagian dari sistem kursi roda cerdas secara keseluruhan. Tiap tugas dalam program dikelompokkan kedalam fungsi-fungsi secara terpisah. Tiap fungsi dipanggil sesuai dengan pengaturan waktu pemanggilannya yang ditetapkan pada saat memprogram. Penjadwalan tugas atau pemanggilan fungsi pada penelitian ini menggunakan *template* Softtimer. Fungsi-fungsi tersebut antara lain:

a.) *update_kecepatan()*

Fungsi ini digunakan untuk mendapatkan hasil pencacahan pulsa enkoder. Fungsi ini dipanggil tiap 50 ms sekali.

b.) *odometri()*

Fungsi ini digunakan untuk perhitungan posisi dan orientasi kursi roda cerdas. Fungsi ini dipanggil tiap 50 ms sekali.

c.) *serial_task()*

Fungsi ini digunakan untuk pengiriman data posisi dan orientasi kursi roda melalui komunikasi serial. Fungsi ini dipanggil tiap 10 ms sekali.

d.) *update_joystick_signals()*

Fungsi ini digunakan untuk *update* data joystick yang diterima dari *laptop*. Fungsi ini dipanggil tiap 200 ms sekali.

Penjadwalan Tugas

Timer 3 pada ATmega128 digunakan untuk menangani penjadwalan tugas ini. Fungsi interupsi *timer 3* diaktifkan untuk menseleksi indeks *timer* mana yang aktif tiap periode *tik*. Nilai periode *tik* pada penelitian ini mengikuti perancangan sistem kursi roda cerdas yang sudah dibuat sebelumnya yaitu 100 us.

Inisialisasi Softtimer

Pengaturan *setting* awal waktu pemanggilan fungsi-fungsi di atas menggunakan fungsi *init_sch_timer()* dengan *listing* kode sebagai berikut.

```

init_sch_timer(5,50,500);
// timer untuk fungsi update_kecepatan()
init_sch_timer(6,60,500);
// timer untuk fungsi odometri()
init_sch_timer(7,100,100);
// timer untuk fungsi serial_task()
    
```

Penjelasan dari *listing* kode di atas misalkan untuk fungsi *update_kecepatan()* adalah sebagai berikut:

indeks *timer* = 5

$delay$ awal pemanggilan = $50 \times 100 \text{ us} = 5 \text{ ms}$
 periode pemanggilan = $500 \times 100 \text{ us} = 50 \text{ ms}$
 Pemberian nilai $delay$ awal pemanggilan bertujuan agar tidak terjadi *overlapping* pada penjadwalan tugas.

Pemanggilan fungsi

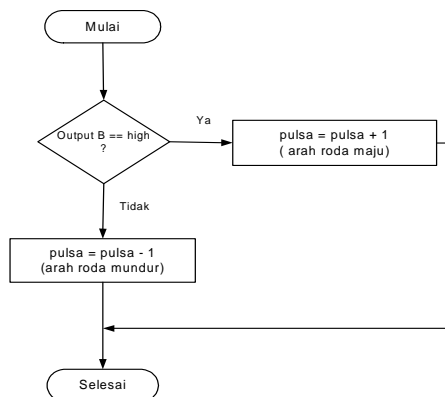
Pemanggilan fungsi-fungsi diatas diletakkan pada daerah *looping* pada fungsi utama program. *Listing* kode untuk Pemanggilan fungsi adalah sebagai berikut:

```

while (1)
{
    if(sch_timer[5].timeout)
    {
        update_kecepatan();
        sch_timer[5].timeout--;
    }
    if(sch_timer[6].timeout)
    {
        odometri();
        sch_timer[6].timeout--;
    }
    if(sch_timer[7].timeout)
    {
        serial_task();
        sch_timer[7].timeout--;
    }
};
  
```

Pembacaan Encoder

Pencacahan pulsa encoder untuk menghitung putaran yang ditempuh roda digunakan External Interrupt 0 untuk roda kanan dan External Interrupt 1 untuk roda kiri. Misalkan pada perancangan perangkat keras untuk roda kanan, *output* A encoder dipasang pada pin External Interrupt 0, sedangkan *output* B dipasang pada pin D.2. Ketika terdeteksi *falling edge* pada *output* A, maka fungsi interupsi External Interrupt 0 akan dipanggil. Pada fungsi interupsi External Interrupt 0 terdapat rutin untuk mendeteksi arah putaran roda dengan diagram alir yang ditunjukkan pada Gambar 3.



Gambar 3 Diagram alir pencacahan pulsa encoder dan pendeteksian arah putaran roda.

Perhitungan Posisi dan Orientasi Kursi Roda

Perhitungan posisi dan orientasi kursi roda digunakan prinsip *dead reckoning* dengan membaca putaran yang ditempuh roda tiap 50 ms. Persamaan 1 sampai 6 pada Bab II digunakan untuk perancangan perangkat lunak ini. Untuk mengkonversi jumlah pulsa enkoder menjadi jarak linier yang ditempuh roda misalkan untuk roda kanan digunakan rumus sebagai berikut:

$$S_R = \frac{n}{N_R} \times K \quad (7)$$

dengan n , N_R , dan K berturut turut adalah jumlah pulsa enkoder didapat, jumlah pulsa enkoder satu putaran roda untuk roda kanan, dan keliling roda.

Nilai posisi dan orientasi kursi roda direpresentasikan dalam kordinat kartesian seperti pada Gambar 1. Panjang diameter roda adalah 38.8 cm, sedangkan jarak pusat roda kanan dengan pusat roda kiri adalah 67.6 cm. Data nilai posisi, kordinat x dan y , yang dikirimkan ke laptop dalam satuan cm sedangkan data nilai orientasi yang dikirimkan ke laptop dalam satuan derajat.

Perancangan Komunikasi Serial pada Mikrokontroler

Komunikasi serial dalam penelitian ini digunakan untuk pengiriman data posisi dan orientasi kursi roda dari mikrokontroller ke laptop dan pengiriman data posisi joystick dari laptop ke mikrokontroller. Teknik komunikasi serial yang digunakan dalam penelitian ini adalah master-slave, dimana *master* adalah *device* yang menerima protokol, sedangkan *slave* adalah *device* yang menerima protokol. Protokol yang dikirimkan master ke slave akan menentukan apakah slave selanjutnya akan menerima data dari master atau mengirim data ke master. Pada penelitian ini laptop difungsikan sebagai *master* sedangkan mikrokontroller difungsikan sebagai *slave*. Tabel 1 memperlihatkan inisialisasi protokol dengan data yang dikirimkan oleh *slave*.

Tabel 1. Protokol komunikasi serial.

Protokol	Master
0x01	meminta posisi kursi roda kordinat x
0x02	meminta posisi kursi roda kordinat y
0x03	meminta sudut orientasi kursi roda
0x11	akan mengirim posisi <i>joystick</i>

Penerimaan Data Serial

Perancangan penerimaan data serial digunakan diagram *state* yang diperlihatkan pada Gambar 3.3. Hubungan Event terhadap kondisi diperlihatkan pada Tabel 2, sedangkan hubungan

Action terhadap perintah yang akan dilakukan ditunjukkan pada Tabel 3.

Pengiriman Data Serial

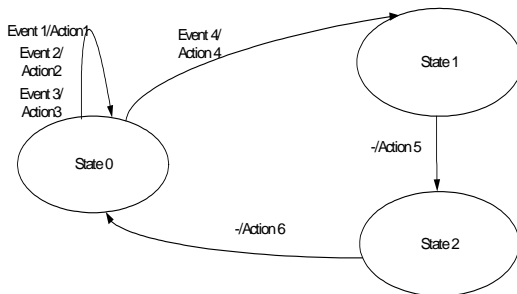
Pengiriman data serial pada mikrokontroller ditangani oleh fungsi `serial_task()`. Data posisi dan orientasi kursi roda bertipe integer dengan memori 16 bit, sehingga pengiriman datanya dilakukan 2 kali, masing-masing 8 bit. Perancangan pengiriman data serial ini digunakan diagram *state* yang diperlihatkan pada Gambar 4.

Tabel 2. Hubungan event terhadap kondisi.

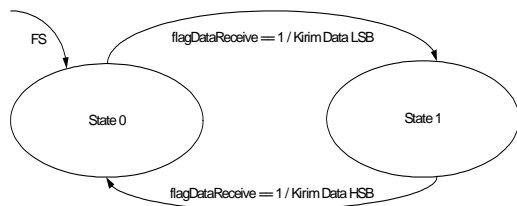
Event	Kondisi
Event 1	jika data yang diterima adalah 0x01
Event 2	jika data yang diterima adalah 0x02
Event 3	jika data yang diterima adalah 0x03
Event 4	jika data yang diterima adalah 0x11

Tabel 3. Hubungan action terhadap perintah.

Action	Perintah
Action 1	persiapan pengiriman data pos. koord x
Action 2	persiapan pengiriman data pos. koord y
Action 3	persiapan pengiriman data sudut orientasi
Action 4	persiapan penerimaan data joystick
Action 5	menerima data posisi x joystick
Action 6	menerima data posisi y joystick



Gambar 4. Diagram state penerimaan data serial pada mikrokontroler.



Gambar 5. Diagram state pengiriman data serial pada mikrokontroler.

Update data Joystick pada Mikrokontroller

Data posisi joystick yang diterima dari laptop melalui komunikasi serial digunakan untuk menggerakkan kursi roda. Perlu diketahui juga pada sistem kursi roda cerdas ini terdapat tiga perilaku, yaitu Joystick, Obstacle Avoidance, dan Anti Collition yang sudah ada pada penelitian sebelumnya. Pada perilaku Joystick kursi roda akan bergerak sesuai dengan posisi joystick yang digerakkan oleh pengguna.

Perancangan Perangkat Lunak pada Komputer

Perancangan Komunikasi Serial pada Komputer

Penjadwalan pengiriman protokol-protokol tersebut menggunakan komponen Timer2. Sedangkan pengiriman dan penerimaan datanya menggunakan fungsi, properti dan event yang telah disediakan oleh komponen serialPort1. Port Komunikasi serial akan dibuka dan mulai mengirimkan data protokol jika komponen `buttonStartMap` diklik oleh *user* dan akan ditutup jika komponen `buttonStopMap` diklik oleh *user*.

Visualisasi Posisi, Orientasi, dan Trajektori Kursi Roda

Posisi, orientasi, dan trajektori kursi roda divisualisasikan secara dua dimensi menggunakan properti dan fungsi dari kelas-kelas yang telah disediakan oleh library GDI+. Visualisasi Gambar kursi roda dibuat menggunakan komponen *Control* buatan `ImageGDI1`. Komponen ini dibuat dengan memodifikasi program dari buku "GDI+ Custom Control with Visual C# 2005". Komponen ini mengambil gambar dari file berekstensi ".ico" dari suatu lokasi didalam komputer. Sudut orientasi dari gambar yang didapat dapat diubah dengan memberikan nilai pada properti `imageGDI1.ImageAngle`. Nilai properti ini ditentukan berdasarkan data sudut orientasi kursi roda yang diterima dari mikrokontroller, sedangkan posisi dari komponen ini ditentukan berdasarkan data posisi kursi roda yang diterima dari mikrokontroller. Data posisi tersebut juga digunakan untuk menggambar trajektori kursi roda. Lokasi `ImageGDI1` dan penggambaran trajektori diletakkan pada komponen `panelMap`. Update visualisasi posisi, orientasi, dan trajektori kursi roda dilakukan setiap 150 ms sekali dengan menggunakan komponen `timer1`.

Update data Joystick pada Komputer

Rutin *update* data *joystick* akan mengubah data posisi *joystick* yang akan dikirimkan ke mikrokontroler. Posisi *joystick* yang digunakan adalah posisi x dan posisi yang diakses dengan membaca properti X_{pos} dan Y_{pos} pada komponen *ActiveX Joystick*.

PENGUJIAN DAN ANALISIS

Pengujian Perhitungan Posisi Kursi Roda

Pengujian perhitungan posisi kursi roda dilakukan untuk mengetahui keakuratan perhitungan posisi menggunakan prinsip *dead reckoning*. Pengujian ini dilakukan dengan menggerakkan kursi roda ke arah tertentu. Setelah berhenti pada posisi tertentu, nilai posisi yang tampil pada program dicatat, kemudian dibandingkan dengan posisi sebenarnya yang diukur menggunakan meteran. Pengujian tersebut kemudian dilanjutkan sampai mendapatkan 5 posisi. Setelah itu mikrokontroler direset, kemudian proses diatas dilakukan sekali lagi, sehingga didapatkan 10 data posisi. Pada pengujian ini *error* antara pengukuran dan perhitungan posisi kursi roda adalah berupa jarak euklidean antara kedua posisi tersebut. *Error* euklidean (e_k) didapatkan menggunakan rumus.

$$e_k = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

dengan (x_0, y_0) adalah kordinat posisi pengukuran, dan (x_1, y_1) adalah koordinat posisi perhitungan.

Pada perancangan, inisialisasi awal posisi kordinat kursi roda adalah (0,0) dengan sudut orientasi 90° , sehingga kursi menghadap ke sumbu y positif ketika mikrokontroler pertama kali dihidupkan. Tabel 4 memperlihatkan hasil pengujian ini. Tabel 4 nomor 1 adalah posisi awal ketika mikrokontroler dihidupkan.

Tabel 4. Hasil pengujian perhitungan posisi kursi roda.

No.	Pengukuran		Perhitungan		<i>Error</i> Euklidean (e_k) (cm)
	x_0 (cm)	y_0 (cm)	x_1 (cm)	y_1 (cm)	
1	0	0	0	0	0
2	77	104	81	109	6.40
3	172	231	185	238	14.76
4	232	375	201	383	32.02
5	19	316	32	333	21.40
6	108	100	101	105	8.60
7	-70	114	-72	114	2.00
8	-144	267	-138	281	15.23
9	-239	383	-227	402	22.47
10	-94	332	-91	317	15.30
11	39	206	-4	148	72.20

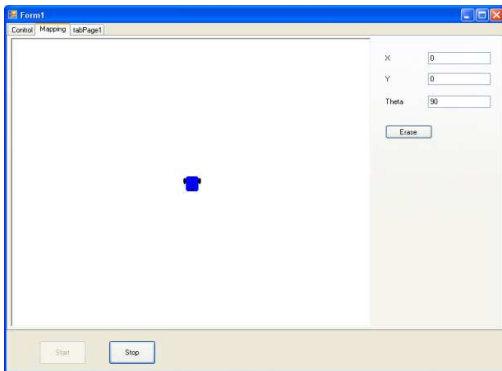
Dari Tabel 4 dapat dilihat bahwa terdapat *error* antara posisi hasil perhitungan menggunakan prinsip *dead reckoning* dengan posisi sebenarnya. *Error* posisi dan orientasi kursi roda tersebut disebabkan oleh kesalahan sistematis dan nonsistematis. Kesalahan sistematis prinsip *dead reckoning* adalah kesalahan pada saat kita menentukan parameter-parameter perhitungan posisi dan orientasi kursi roda, seperti diameter roda, keliling roda, dan jumlah pulsa enkoder satu putaran roda. Kesalahan nonsistematis *dead reckoning* adalah kesalahan pada interaksi kursi roda dengan lingkungan seperti kondisi jalan yang tidak datar dan selip antara roda dengan jalan.

Kesalahan sistematis pada navigasi kursi roda ini salah satunya disebabkan oleh faktor mekanis, seperti pemasangan dudukan enkoder yang tidak benar-benar pas sehingga poros enkoder tidak benar benar seporos dengan motor dan celah-celah gear konversi putaran motor dengan roda yang terlalu lebar sehingga ketika motor sedikit berputar, roda belum tentu langsung ikut berputar. Kesalahan ini menimbulkan *error* pada jumlah pulsa enkoder satu putaran roda. *Error* sistematis lain adalah kurang cermatnya dalam pengukuran diameter roda dan jarak antara kedua pusat roda kanan dan pusat roda kiri karena pengukuran dilakukan menggunakan penggaris biasa. Meskipun hanya *error* beberapa milimeter namun kesalahan kecil tersebut tetap akan terakumulasi pada saat perhitungan dengan prinsip *dead reckoning*. Kurang cermatnya dalam pengukuran jarak atau posisi sebenarnya menggunakan meteran pada saat pengujian juga dapat menambah *error* pada pengujian ini.

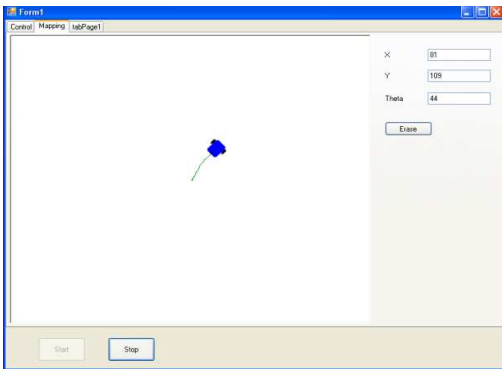
Dari Tabel 5 dapat dilihat juga bahwa semakin jauh kursi roda digerakkan, maka semakin besar kesalahannya. Kesalahan yang semakin besar tersebut adalah akibat dari proses integrasi dalam menghitung posisi tiap sampling waktunya, karena pada dasarnya prinsip *dead reckoning* adalah menentukan posisi sekarang berdasarkan posisi sebelumnya, sehingga *error* perhitungan posisi sebelumnya akan menambah *error* perhitungan posisi sekarang. Proses integrasi ini akan menambah *error* secara kumulatif sehingga semakin jauh jarak tempuh kursi roda maka semakin besar *error* perhitungan posisinya.

Pengujian Visualisasi Posisi dan Orientasi Kursi Roda dan Komunikasi Serial

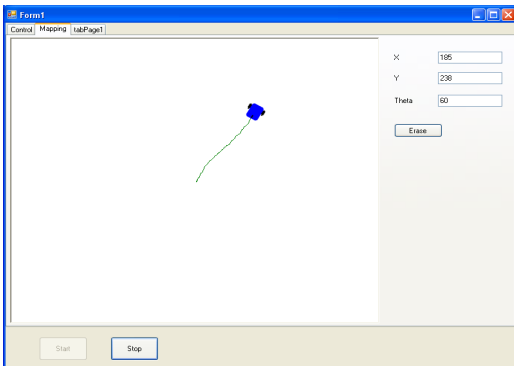
Pengujian visualisasi posisi dan orientasi kursi roda dan komunikasi serial dilakukan secara bersamaan. Pengujian ini dilakukan dengan cara menggerakkan kursi roda ke 3 posisi tertentu. Perubahan posisi pada visualisasi pengujian ini adalah 2 cm/pixel. Pada saat pengujian nilai posisi dan orientasi kursi roda yang tampil pada TextBox dibandingkan dengan yang tampil pada LCD. Hasil visualisasi pengujian ini diperlihatkan pada Gambar 4.1, Gambar 4.2, Gambar 4.3, dan Gambar 4.4. Tabel 4.3 memperlihatkan perbandingan data posisi dan orientasi kursi roda pada laptop dan LCD.



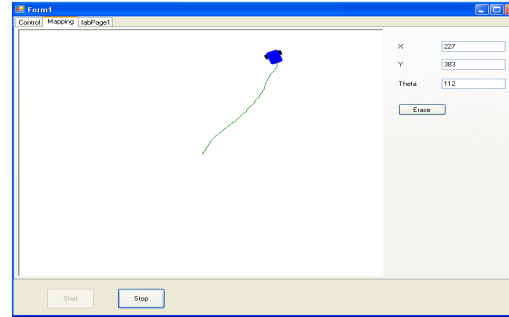
Gambar 6. Visualisasi posisi awal.



Gambar 7. Visualisasi menuju ke posisi tujuan 1.



Gambar 8 Visualisasi menuju ke posisi tujuan 2.



Gambar 9. Visualisasi menuju ke posisi tujuan 3.

Tabel 5 Hasil pengujian visualisasi posisi dan orientasi kursi roda dan komunikasi serial.

Gerakan		Posisi Awal	Menuju Tujuan 1	Menuju Tujuan 2	Menuju Tujuan 3
posisi x	komputer	0	81	185	227
	LCD	0	81	185	227
posisi y	komputer	0	109	238	383
	LCD	0	109	238	383
sudut orientasi	komputer	90	44	60	112
	LCD	90	44	60	112

Pengujian ini bertujuan untuk melihat apakah data nilai posisi dan orientasi kursi roda dapat dikirimkan dari mikrokontroler ke komputer dan divisualisasi dengan benar. Dari Tabel 4.2 dapat dilihat bahwa nilai posisi dan orientasi kursi roda pada laptop sama dengan nilai posisi dan orientasi kursi roda pada LCD atau mikrokontroler.

PENUTUP

Berdasarkan pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut. Semakin jauh jarak tempuh kursi roda, maka semakin besar *error* perhitungan posisi dan orientasi kursi roda menggunakan prinsip *dead reckoning*.

Apabila dilakukan gerakan yang berlawanan arah dari gerakan sebelumnya, *error* posisi dan orientasi dapat lebih kecil dari *error* posisi dan orientasi sebelumnya.

Data posisi dan orientasi kursi roda dapat dikirimkan dari mikrokontroler ke laptop dan divisualisasikan dengan benar.

Untuk pengembangan sistem lebih lanjut, maka dapat diberikan saran-saran sebagai berikut penambahan sensor atau sistem navigasi lain yang dapat mengkalibrasi ulang navigasi berbasis prinsip *dead reckoning* pada kursi roda akan menambah keakuratan perhitungan nilai posisi dan orientasi, sehingga pemetaan posisi dan orientasi kursi roda dapat dilakukan untuk jarak tempuh yang jauh.

Karena navigasi berbasis prinsip *dead reckoning* pada penelitian ini menggunakan sensor inersial enkoder, maka penambahan sistem navigasi lain yang paling tepat adalah navigasi dengan sensor eksternal seperti GPS (*Global Positioning System*).

Perlu dibandingkan juga navigasi kursi roda cerdas berbasis prinsip *dead reckoning* antara menggunakan sensor enkoder dengan menggunakan sensor inersial lainnya seperti *gyroscope* dan *accelerometer* untuk mengetahui mana yang lebih baik.

Program visualisasi pemetaan posisi dan orientasi kursi roda cerdas dapat dikembangkan agar tampilan lebih menarik seperti mengubah tampilan 2D menjadi 3D, mempercantik antarmuka program dan lain-lain.

DAFTAR PUSTAKA

- Chand, Mahesh, *GDI+ Programming with C#*, Pearson Education, Boston, U.S.A., 2004.
- Cyberton Solution, *Cara Mudah Menguasai Microsoft C#*, ANDI, Yogyakarta, 2009.
- Heryanto, M.Ary dan Wisnu Adi P., *Pemrograman Bahasa C untuk Mikrokontroler ATMEGA 8535*, Penerbit Andi, Yogyakarta, 2008.
- Iulian Serban, Dragos Brezoim, Tiberiu Radu, dan Adam Ward, *GDI+ Custom Controls with Visual C#*, PACKT Publishing, 2005.
- Joni, I Made dan Budi Harharjo, *Pemrograman C dan Implementasinya*, Penerbit Informatika, Bandung, 2006.
- Kurniawan, Erick, *Pemrograman GDI+ dengan C# (Part 1)*, <http://actualtraining.wordpress.com>, Oktober 2009.
- Matteo Golfarelli, Dario Maio, dan Stefano Rizzi, *Elastic Correction of Dead-Reckoning Errors in Map Building*, DEIS, Bologna, Italia.
- Miftahurrozaq, *Perancangan Sistem Kursi Roda Cerdas Menggunakan Pendekatan Model Fungsional dan Model Tingkah Laku dan Perancangan Perilaku Obstacle Avoidance*, Skripsi S-1, Universitas Diponegoro, Semarang, 2010.
- Prasetyo, Rudy, *Implementasi Sistem Kontrol Navigasi Reaktif Subsumption Pada Kursi Roda Cerdas*, Skripsi S-1, Universitas Diponegoro, Semarang, 2008.
- Setiawan, Iwan, *Fungsi Soft Timer untuk Keperluan Operasi Tundaan dan Penjadwalan (Scheduling) pada Sistem Embedded*, <http://iwan.blog.undip.ac.id>, Januari 2010.
- Setiawan, Iwan, *Model Kinematika Robot Mobile Jenis Differential Drive*, <http://iwan.blog.undip.ac.id>, Juli 2009.
- , *AT Mega 128 Data Sheet*, <http://www.atmel.com>, Oktober 2009.
- , *Dead reckoning*, <http://en.wikipedia.org>, Januari 2010.
- , *Joystick Control Help*, <http://www.globalmajic.com>, Januari 2010.
- , *Linear and Rotary Encoders*, <http://zone.ni.com>, Januari 2010.
- , *Rotary Encoder (Incremental Type) E30S4 Series Manual*, <http://www.autonics.com>, Oktober 2009.
- , *Simple Serial Communication with Micro soft Visual C# Express*, <http://csharp.simpleserial.com>, Oktober 2009.