Makalah Seminar Tugas Akhir

SISTEM KONTROL NAVIGASI PADA *MOBILE* ROBOT BERBASIS PCBC *(PIECEWISE CUBIC BEZIER CURVE)*

Arif Dwi Utomo [1], Iwan Setiawan, ST. MT.[2], Trias Andromeda, ST. MT.[2]

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstrak

Dewasa ini teknologi robot mobil telah mengalami perkembangan yang sangat pesat. Banyak aplikasi teknologi robot mobil yang digunakan dalam kehidupan sehari-hari. Seperti robot pembersih ruangan, penjinak bom, kursi roda cerdas, dan penghantar barang di pabrik-pabrik ataupun rumah sakit. Sebuah robot mobil memerlukan sistem navigasi agar dapat bergerak dari satu tempat ke tempat lain. Untuk itu robot mobil harus mampu untuk membangkitkan sendiri lintasan yang akan dilaluinya.

Pada tugas akhir ini, metode PCBC (Piecewise Cubic Bezier Curve) digunakan untuk membangkitkan lintasan robot. Nilai kelengkungan lintasan dan kinematika berperan dalam updating kecepatan roda kanan dan roda kiri robot mobil berdasarkan posisi robot setiap saat. Pengujian dilakukan dengan memberi masukan sembarang titik referensi dan mengamati respon kecepatan roda kanan dan roda kiri robot, baik secara visual maupun simulasi sampai didapatkan respon kecepatan terbaik.

Hasil pengujian menunjukkan robot mobil mampu menjejak lintasan yang dibangkitkan dengan metode PCBC walaupun hanya untuk titik tertentu saja. Titik masukan dan kecepatan referensi mempengaruhi respon kecepatan roda kanan dan roda kiri robot. Sensor ping yang digunakan mampu mendeteksi rintangan yang ada di lintasan robot. Kecepatan roda kanan dan roda kiri robot dipengaruhi oleh lebar bidang PWM yang digunakan.

Kata kunci : robot mobil, PWM, lintasan, PCBC.

I. PENDAHULUAN

1.1 Latar belakang

Perkembangan teknologi dibidang robotika telah menjadi perhatian yang cukup serius dalam beberapa tahun terakhir. Perkembangan teknologi robot terutama pada peran robot yang dapat menggantikan pekerjaan manusia terutama dalam lingkungan yang berbahaya, seperti daerah radiasi nuklir, penjelajahan ruang angkasa, perang, penjinak bom dan lain-lain.

Robot mobil adalah alat yang dapat bergerak secara otomatis untuk melakukan pekerjaan tertentu, diantaranya bergerak menuju lokasi atau daerah yang telah ditetapkan atau menuju sasaran tertentu. Ada banyak metode yang dapat digunakan untuk membuat sebuah trayektori bagi robot mobil. Salah satunya adalah dengan menggunakan metode berbasis PCBC (*Picewise Cubic Bezier* Curve) yang sudah umum digunakan dalam bidang komputer grafis dan juga pada beberapa pembuatan trayektori robot.

1.2 Tujuan

Tujuan dalam pengerjaan Tugas Akhir ini adalah merancang generator trayektori berbasis kurva Bezier yang akan dijadikan sebagai referensi lintasan oleh robot dan menjadikan pergerakan robot pada lintasan tersebut lebih *smooth*.

1.3 Pembatasan Masalah

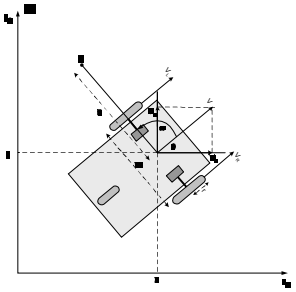
Agar permasalahan yang dibahas terfokus dan tidak melebar, maka Tugas Akhir ini mengambil batasan masalah sebagai berikut:

1. Pergerakan robot mengabaikan massa, percepatan, gaya, gesekan karena robot mobil yang dibuat berukuran kecil.
2. Robot mobil yang dibuat menggunakan penggerak diferensial.
3. Posisi awal (P0) robot diasumsikan berada di koordinat (0,0) dengan arah hadap robot adalah sumbu x.
4. Mikrokontroler yang digunakan adalah mikrokontroler AVR ATMega8535.
5. Untuk sistem kendali, digunakan sistem kalang terbuka (*open loop*).
6. Bahasa pemrograman yang digunakan adalah dengan menggunakan bahasa C yang diadaptasikan pada *software* CodeWizardAVR V1.24.0 Standard.
7. Robot mobil yang dirancang hanya dapat melakukan gerakan maju dan tidak didesain untuk manuver bergerak mundur.

II. DASAR TEORI

2.2 Model Kinematika Robot Mobil Penggerak Differensial

Salah satu jenis robot mobil yang umum digunakan, terutama untuk dioperasikan dalam ruang adalah robot mobil dengan pengemudian atau sistem penggerak diferensial (*differensial drive*). Alasan utamanya karena relatif lebih fleksibel dalam melakukan manuver serta kemudahan dalam pengontrolannya. Gambar 1 memperlihatkan arsitektur robot dilihat dari bagian atas.



Gambar 1 Posisi dan orientasi robot mobil dalam sistem koordinat cartesian

Kecepatan linier robot mobil pada masing–masing roda kanan dan kiri berturut–turut adalah V_R dan V_L. Kecepatan rotasi masing–masing roda dengan jari–jari r adalah ω_R dan ω_L sesuai dengan Persamaan 2.1 dan 2.2 berikut :

$$\omega_R(t) = \frac{V_R(t)}{r} \quad \text{..... (2.1)}$$

$$\omega_L(t) = \frac{V_L(t)}{r} \quad \text{..... (2.2)}$$

Ketika robot melakukan gerak memutar (berotasi) sesaat dengan panjang jari–jari R diukur dari pusat rotasi dan titik pusat kedua titik (lihat Gambar 2.1) maka kecepatan rotasi disetiap titik robot tersebut selalu sama (robot adalah sistem mekanis yang rigid), sehingga Persamaan (2.3) dan/atau (2.4) berikut ini berlaku untuk menghitung kecepatan rotasi dari robot tersebut:

$$\omega(t) = \frac{V_R}{R + L/2} \quad \text{..... (2.3)}$$

$$\omega(t) = \frac{V_L}{R - L/2} \quad \text{..... (2.4)}$$

Berdasarkan Persamaan (2.3) dan (2.4) kecepatan rotasi robot tersebut dapat dihitung hanya bedasarkan informasi dari kedua kecepatan linier roda robot tersebut:

$$\omega(t) = \frac{V_R(t) - V_L(t)}{L} \quad \text{..... (2.5)}$$

Sedangkan jari–jari lintasan dapat dicari dengan mensubstitusikan Persamaan (2.4) ke dalam Persamaan (3), dan memecahkannya untuk R :

$$R = \frac{L(V_R + V_L)}{2(V_R - V_L)} \quad \text{..... (2.6)}$$

Terlihat pada Persamaan (2.6), jari–jari lintasan lingkaran sesaat berbanding terbalik dengan selisih kedua kecepatan roda robot. Semakin kecil selisih kedua kecepatan roda maka jari–jari lingkaran sesaat yang dibentuk oleh lintasan robot tersebut semakin panjang dan sebaliknya. Sedangkan jika kecepatan linier roda kanan sama dengan kecepatan linier roda kiri maka R = ∞, atau secara praktis robot akan bergerak membntuk lintasan yang lurus. Agar robot berotasi pada pusat sumbunya (R=0) maka berdasarkan persamaan (2.6), kecepatan kedua roda tersebut harus berlawanan.

Berdasarkan Persamaan (2.5) dan (2.6), maka kecepatan linier robot dapat dihitung dengan menggunakan Persamaan (2.7) berikut :

$$V(t) = \frac{V_R(t) + V_L(t)}{2} \quad \text{..... (2.7)}$$

Agar lebih sederhana, Persamaan (2.5) dan (2.7) dapat dikumpulkan dalam bentuk persamaan matrik – vektor sebagaimana berikut :

$$\begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/L & -1/L \end{bmatrix} \begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix} \quad \text{..... (2.8)}$$

Persamaan (2.8) di atas pada dasarnya memperlihatkan relasi antara kecepatan linier roda–roda robot terhadap kecepatan linier dan angular robot, sedangkan Persamaan (2.9) di bawah memperlihatkan relasi sebaliknya :

$$\begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix} = \begin{bmatrix} 1 & D/2 \\ 1 & -D/2 \end{bmatrix} \begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix} \quad \text{..... (2.9)}$$

Dengan mengetahui kecepatan linier dan angular robot setiap saat, maka kecepatan pada setiap sumbu kartesian dapat dicari dengan cara memproyeksikan vektor kecepatan robot kepada sumbu–sumbu tersebut.

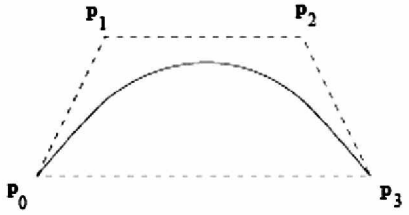
2.2 Kurva Bezier Jenis Kubik *(Cubic Bezier Curve)*

Kurva bezier jenis kubik menggunakan empat titik acuan yaitu P_0, P_1, P_2, dan P_3 dan persamaannya sebagai berikut :

$$P(q) = P_0*(1-q)^3+3*P_1*q*(1-q)^2+3*P_2*q^2*(1-q)+P_2*q^3$$

$$P(q)=P_q(1-q)^3+3*P_1*q*(1-q)^2+3*P_2*q^2*(1-q)+P_3*q^3$$

Dalam hal ini P adalah sebuah fungsi kontinyu yang terdefinisi untuk nilai q antara 0 sampai 1. P_0 dan P_3 adalah vektor yang merepresentasikan sebuah titik awal dan akhir pada fungsi tersebut, bentuk kelengkungan kurva dikontrol oleh titik–titik P_1 dan P_2. Gambar 2 berikut ini memperlihatkan kurva Bezier kubik pada sistem koordinat kartesian.



Gambar 2 Kurva Bezier jenis kubik

Kurva Bezier yang digunakan sebagai lintasan robot adalah bidang dua dimensi yang didefinisikan oleh koordinat x dan y.[5] Oleh karena itu, persamaan bezier yang digunakan dapat dituliskan sebagai berikut :

$$Px = Ax*q^3 + Bx*q^2 + Cx*q + P0x \quad \text{..... (2.13)}$$

$$Py = Ay*q^3 + By*q^2 + Cy*q + P0y \quad \text{..... (2.14)}$$

Dengan

$$Cx = 3*(P1x - P0x) \quad \text{..... (2.15)}$$

$$Bx = 3*(P2x - P1x) - Cx \quad \text{..... (2.16)}$$

$$Ax = P3x - P0x - Cx - Bx \quad \text{..... (2.17)}$$

$$Cy = 3*(P1y - P0y) \quad \text{..... (2.18)}$$

$$By = 3*(P2y - P1y) - Cy \quad \text{.....(2.19)}$$

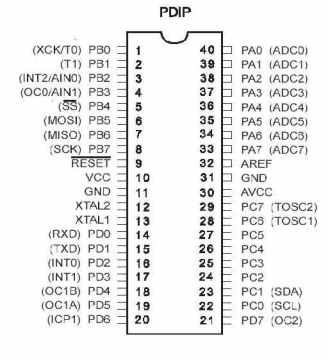
$$Ay = P3y - P0y - Cy - By \quad \text{.....(2.20)}$$

Sedangkan untuk kelengkungan kurva Bezier dapat dicari dengan menggunakan Persamaan 2.21 berikut :

$$\text{curve(t)} = \frac{\frac{dx}{dq}\frac{d^2y}{dq^2} - \frac{dy}{yq}\frac{d^2x}{dq^2}}{\sqrt[3]{\left(\frac{dx}{dq}\right)^2 + \left(\frac{dy}{dq}\right)^2}} \quad \text{.....(2.21)}$$

2.3 Mikrokontroller ATMega8535

ATMega8535 adalah mikrokontroller CMOS 8-bit daya rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, ATMega8535 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi komsumsi daya *versus* kecepatan proses.



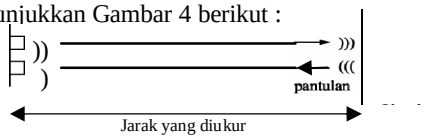
Gambar 3. Susunan kaki mikrokontroller ATMega8535

Penjelasan dari masing-masing kaki adalah sebagai berikut:

a. VCC (kaki 10) dihubungkan ke Vcc.
b. GND (kaki 31) dihubungkan ke *ground*.
c. Port A (PA7..PA0) (kaki 33-40) merupakan port 8 bit dua arah (*bidirectional*) I/O. Port ini berfungsi sebagai ADC atau dapat juga berfungsi sebagai port data/alamat I/O ketika menggunakan SRAM eksternal.
d. Port B (PB7-PB0) (kaki 1-8) merupakan port 8 bit dua arah (*bidirectional*) I/O, untuk berbagai keperluan (*multipurpose*).
e. Port C (PC7..PC0) (kaki 22-29) adalah port 8 bit dua arah I/O, dengan internal *pull-up* resistor. Port C ini juga berfungsi sebagai port alamat ketika menggunakan SRAM eksternal.
f. Port D (PD0..PD7) (kaki 14-21) adalah port 8 bit dua arah I/O dengan resistor *pull-up* internal. Port D juga dapat berfungsi sebagai terminal khusus.
g. *Reset* (kaki 9). Kondisi rendah yang lebih lama dari 50 ns akan mereset mikrokontroller.
h. XTAL1 (kaki 13) masukan dari osilator eksternal dan masukan bagi rangkaian osilator internal.
i. XTAL2 (kaki 12) keluaran dari rangkaian osilator internal. Kaki ini digunakan apabila dipakai osilator kristal.
j. ICP1 (kaki 20) adalah kaki masukan untuk fungsi *Timer/Counter*1 *Input Capture*.
k. OC1B (kaki 18) adalah kaki keluaran bagi fungsi *Output Compare* B keluaran *Timer/Counter*1.
l. AVCC adalah pin sumber tegangan untuk port A dan ADC. Harus terkoneksi dengan Vcc walaupun ADC tidak digunakan.
m. AREF merupakan referensi analog untuk ADC.

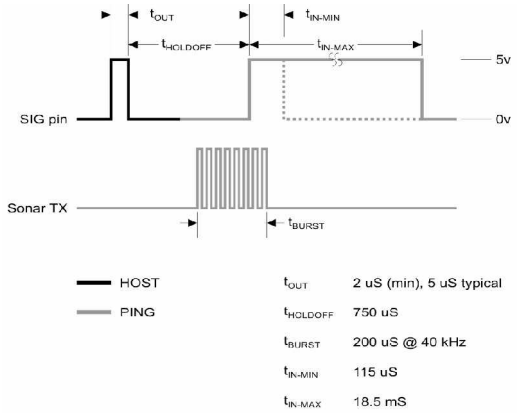
2.4 Sensor Ping)))™

Sensor ping digunakan untuk mendeteksi rintangan yang ada di depan robot. Sensor ini memancarkan gelombang ultrasonik berfrekuensi 40 KHz selama 200 μ s. Gelombang ultrasonik merambat di udara dengan kecepatan rambat suara di udara, yaitu 344 meter per detik, mengenai objek dan memantul kembali ke sensor seperti ditunjukkan Gambar 4 berikut :



Gambar 4 Ilustrasi cara kerja sensor ping)))™.

Jarak antara robot dan rintangan dihitung berdasarkan lama waktu yang dibutuhkan sensor untuk menerima kembali gelombang ultrasonik yang telah dipancarkannya. Sensor ping memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan $_{tOUT}$ min. 2 μs). Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. Sensor ping mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi Ping akan membuat *output low* pada pin SIG. Lebar pulsa High (t_{IN}) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur adalah [(t_{IN} s x 344 m/s) ÷ 2] meter. Berikut adalah diagram pewaktuan sensor ping :



Gambar 5. Diagram pewaktuan sensor ping)))™

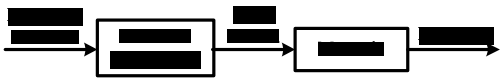
III PERANCANGAN

Pada sistem navigasi robot mobil yang berbasis PCBC ini, digunakan titik koordinat pada kuadran I sebagai pemandu gerak robot. Robot mobil diasumsikan berada pada titik awal (0,0) dan arah hadap robot sejajar

dengan sumbu x. Perpindahan robot dari titik awal sampai titik akhir sebanding dengan perubahan waktu.

Robot menggunakan sistem kemudi jenis differensial yang secara umum memiliki tingkat keluwesan dalam melakukan manuver atau berotasi dalam gerakkannya. Gerakan atau manuver dilakukan dengan mengendalikan kecepatan putar antara roda kanan dan roda kiri.

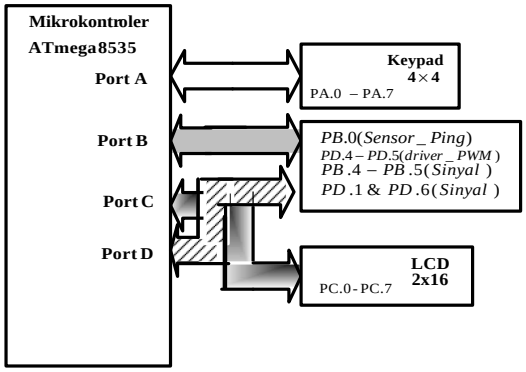
Untuk sistem kendali, digunakan sistem kalang terbuka (*open loop*). Sistem pengaturan kalang terbuka, keluaran tidak diumpanbalikkan untuk dibandingkan dengan masukan acuan. Diagram blok sistem pengaturan kalang terbuka dapat dilihat pada Gambar 6.



Gambar 6 Diagram blok sistem pengaturan kalang terbuka

3.1 Perancangan Perangkat Keras

Diagram blok perancangan perangkat keras, ditunjukkan Gambar 7 berikut :



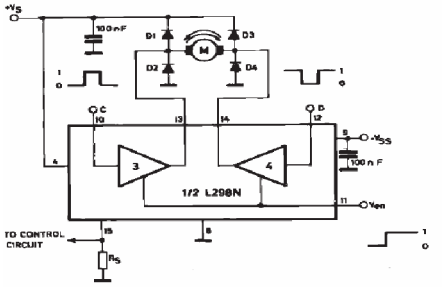
Gambar 7 Diagram blok perancangan perangkat keras.

Secara umum perancangan sistem ini dapat dijelaskan sebagai berikut :

1. Mikrokontroller ATMega8353 sebagai pusat pengolahan data yang diprogram dengan menggunakan bahasa C embended.
2. Sensor Ping sebagai pendeteksi rintangan dan untuk mengetahui jarak robot dengan rintangan.
3. Dua motor dc sebagai penggerak roda kanan dan roda kiri robot. Kecepatan putarnya dikendalikan melalui perubahan *duty cycle* PWM.
4. Keypad digunakan untuk memberi titik masukan. Pemberian titik akhir, kecepatan referensi dan jarak rintangan dilakukan melalui keypad.
5. LCD digunakan untuk tampilan saat memasukkan titik akhir, kecepatan referensi robot dan jarak rintangan terhadap robot.
6. *Driver* motor dc menggunakan L298N yang mampu men*drive* arus sampai 4 A. Sinyal masukkan PWM ke *driver* dari mikrokontroller ATMega8535 pinD.4 (OCR1B) dan pinD.5 (OCR1A).

3.2 *Driver* Motor Dc

Driver motor dc menggunakan L298N yang didesain mampu men*drive* arus sampai 4 A dengan tegangan berkisar antara 2.5 V – 46 V. Gambar 8 berikut adalah rangkaian ½ L298N :



Gambar 8 *Driver* motor dc bidirectional ½ L298N.

Driver L298N yang digunakan pada robot ini, menggunakan Vss (*logic supply voltage*) sebesar 5 Volt dan tegangan referensi (Vs) sebesar 12 Volt. Masukkan PWM pada L298N melalui pin 5,7 (diaktifkan dengan memberikan logitka "1" pada pin 6) dan 10,12 (diaktifkan dengan memberikan logitka "1" pada pin 11) dengan konfigurasi masukan sebagai berikut :

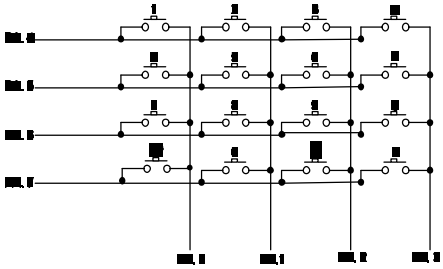
Tabel 3.1 Konfigurasi masukkan PWM

Ven(6,11)	Masukan	Fungsi
H	C = H ; D = L	Putar maju
	C = L ; D = H	Putar mundur
	C = D	Motor berhenti
L	C = X*; D = X*	Motor berhenti

*X = *don't care*

3.3 *Keypad*

Keypad digunakan untuk memasukkan titik akhir, kecepatan referensi dan jarak rintangan terhadap robot. *Keypad* menggunakan tipe matrik 4x4 dengan rangkaian *keypad* menggunakan PORTA. Skematik rangkaian *keypad* diperlihatkan pada Gambar 9 berikut :



Gambar 9 Rangkaian keypad tipe matrik 4x4.

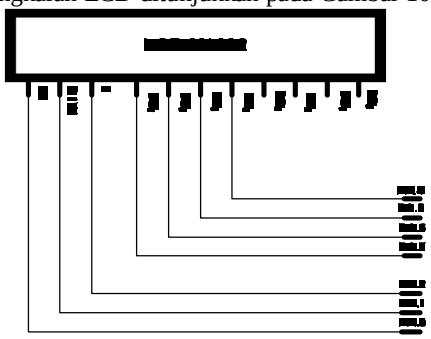
Keypad tipe matrik 4x4 ini memiliki 16 tombol dengan fungsi yang berbeda – beda. Berikut tabel fungsi tiap tombol :

Tabel 3.2 Fungsi tombol *keypad*

Tombol	Keterangan
0 – 9	Masukan nilai
CAN	Hapus nilai masukkan
ENT	*Enter*
↓	Menjalankan robot

3.4 LCD dan *Driver* LCD (*Liquid Crystal Display*)

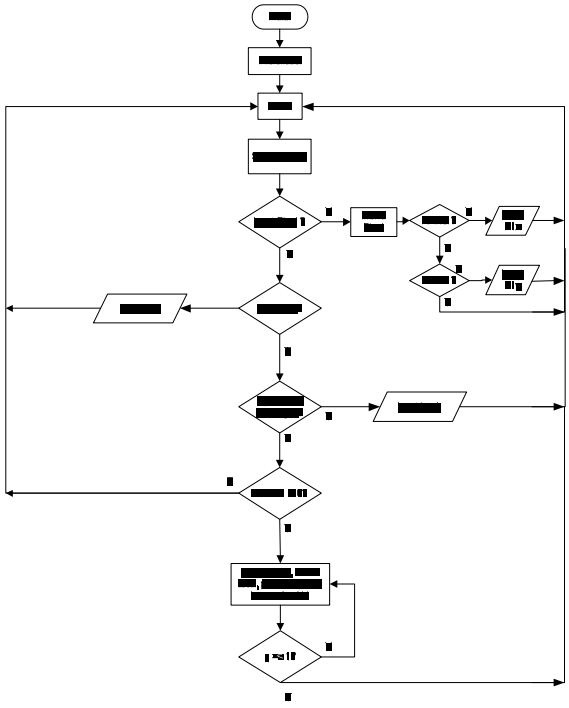
Pada robot mobil ini, LCD digunakan sebagai tampilan saat memasukkan titik–titik referensi dan kecepatan referensi. Pada mikrokontroller ATMega8535, LCD ini dihubungkan melalui port A (pin 33 – pin 40). Rangkaian LCD ditunjukkan pada Gambar 10.



Gambar 10 Rangkaian LCD M1632

3.5 Perancangan Perangkat Lunak

Pemrograman pada ATMega8535 menggunakan bahasa C *embended* CodeVisionAVR versi 1.24.0 standard. Program utama pada robot mobil ini adalah untuk membangkitkan titik–titik lintasan dan kecepatan berdasarkan titik referensi dan kecepatan referensi. Titik–titik lintasan diperoleh melalui persamaan bezier. Berikut adalah diagram alir utama pemrograman robot mobil ini :



Gambar 11 Diagram alir pemrograman robot mobil

Diagram alir di atas menunjukkan apabila nilai kecepatan robot masih sama dengan nol dan tombol jalan sudah ditekan maka robot tidak akan berjalan dan akan meminta operator untuk mengulang *input* kecepatan referensi.

3.2.3. Sub Rutin Interupsi *Timer0*

Sub rutin interupsi *timer0* digunakan untuk meng-*pdate* posisi dan kecepatan robot setiap saat. *Update* titik dan kecepatan dilakukan setiap 0.05 detik.

3.2.4. Sub Rutin Interupsi *Timer2*

Sub rutin interupsi *timer2* digunakan untuk menghitung waktu yang diperlukan sensor ping untuk membaca gelombang pantulan.

3.2.5 Sub Rutin Masukan Final dan Kecepatan Referensi

Titik referensi robot terdiri dari titik awal dan akhir (P0 dan P1) dan titik kontrol (P1 dan P3) yang berada pada kuadran I garis koordinat sehingga semuanya bernilai positif. Posisi awal robot di titik (0,0) sehingga nilai P0 adalah selalu (0,0).

3.2.6 Sub Rutin *Entr*y

Sub rutin ini mendeteksi masukkan angka melalui *keyapad* dan menghitung hasil masukkan tersebut.

3.2.7 Sub Rutin Pengambilan Masukan *Keypad*

Terdapat dua proses utama dalam pemindaian *keypad*, yaitu pemindaian baris dan pemindaian kolom. Masing-masing baris dan kolom memiliki nilai yang apabila hasilnya digabungkan akan menentukan nilai tombol yang ditekan.

3.2.8 Sub Rutin Pengendali Tampilan

Sub rutin ini mengatur tampilan saat memasukkan nilai untuk menentukan titik final, kecepatan referensi dan jarak rintangan robot.

3.2.9 Sub Rutin Pengendali Sensor Ping

Sensor ping digunakan untuk mendeteksi adanya halangan di depan robot dengan memancarkan gelombang ultrasonik.

3.2.10 Sub Rutin *length*

Sub rutin lenght dipanggil untuk menghitung panjang lintasan yang harus dilalui oleh robot.

3.2.11 Sub Rutin *Updating* Posisi dan Kecepatan

Sub rutin ini untuk menghitung kecepatan robot setiap saat berdasarkan posisi robot. Sub rutin ini dipanggil saat *timer0 overflow* interrupsi.

IV. ANALISA DAN PEMBAHASAN

4.1 Pengujian Sensor Ping

Sensor ping digunakan untuk mendeteksi halangan/rintangan pada lintasan robot. Berikut hasil pengujian sensor ping :

Tabel 4.1 Perbandingan jarak masukkan dan jarak pengukuran

Masukkan Jarak (cm)	Pengukuran (cm)
3	3
5	5
8	8
10	10
17	17
20	20
23	23

25	25
27	27
30	30

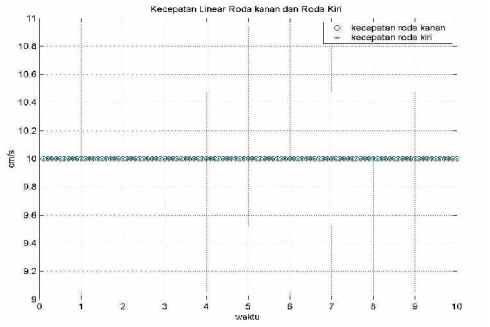
Dari Tabel 4.1 terlihat bahwa hasil masukkan titik dan hasil pengukuran jarak robot terhadap rintangan ada perbedaan. Hal ini disebabkan oleh prinsip kerja sensor yang menggunakan pantulan gelombang suara. Selain itu juga tergantung pada bentuk permukaan rintangan, karena akan mempengaruhi arah pantulan gelombang suara.

4.2 Hasil Simulasi dan Percobaan Robot Mobil

Simulasi dilakukan untuk mengetahui nilai kecepatan roda kanan dan roda kiri serta membandingkannya dengan nilai yang digunakan untuk masukan ke OCR1AL dan OCR1BL Berikut beberapa hasil simulasi dan percobaan dengan beberapa titik variasi titik :

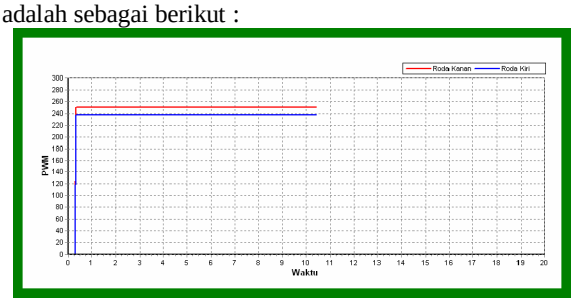
4.2.1 Simulasi dan Percobaan 1

Hasil simulasi dengan titik P1(0,0), P2(0,0), P3(100,0) dan kecepatan referensi 10 cm/detik adalah sebagi berikut :



Gambar 12 Diagram kecepatan roda kanan dan roda kiri hasil simulasi

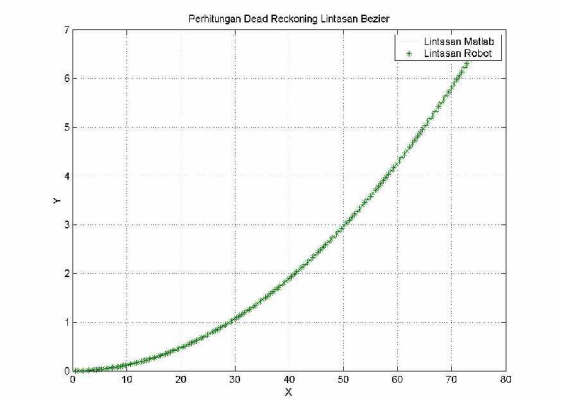
Pada Gambar 12 terlihat bahwa antara kecepatan linier roda kanan dan roda kiri sama, sehingga berdasarkan simulasi robot berjalan lurus. Sedangkan untuk hasil pembacaan kecepatan robot pada percobaan adalah sebagai berikut :



Gambar 13 Diagram kecepatan roda kanan dan roda kiri hasil percobaan pada robot

Pada Gambar 13 menunjukkan kecepatan roda kanan dan kiri robot konstan pada nilai tertentu. Tetapi dari grafik tersebut terlihat bahwa nilai kecepatan roda kiri (warna biru) lebih kecil dari roda kanan (warna merah). Hal ini terjadi karena adanya perbedaan karakteristik motor dc yang digunakan seperti yang telah dijelaskan sebelumnya.

Pada penggambaran lintasan antara perhitungan simulasi dan pembacaan dari mikro, terdapat perbedaan. Berikut adalah perbandingan lintasan antara simulasi dan pembacaan dari mikro :

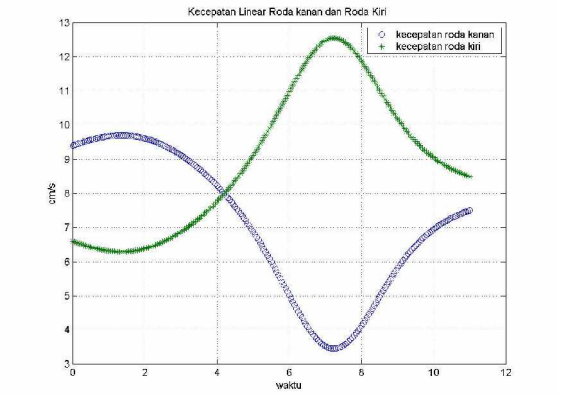


Gambar 14 Perbandingan lintasan berdasarkan perhitungan kecepatan simulasi dan mikro.

Pada Gambar 14 di atas, terjadi perbedaan lintasan. Sebenarnya dalam pengamatan, robot berjalan lurus. Penggambaran lintasan di atas berdasarkan nilai kecepatan yang telah didapat, baik dari simulasi ataupun dari pembacaan mikro. Seperti telah dijelaskan sebelumnya bahwa karakteristik motor dc kanan dan kiri berbeda sehingga diperlukan konstanta pengali untuk menyamakan keluaran kecepatan motor..

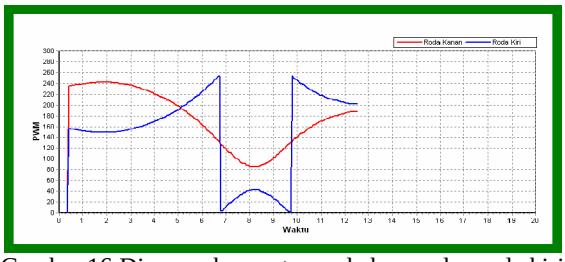
4.2.2 Simulasi dan Percobaan 3

Hasil simulasi dengan titik P1(40,0), P2(40,40), P3(80,0) dan kecepatan referensi 8 cm/detik adalah sebagi berikut :



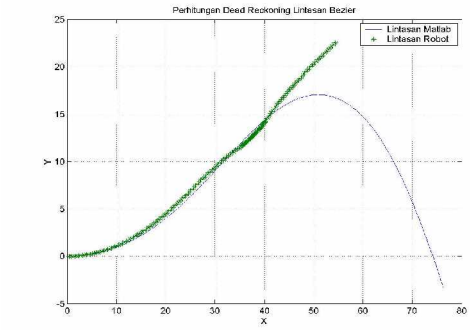
Gambar 15 Diagram kecepatan roda kanan dan roda kiri hasil simulasi

Gambar 15 di atas menunjukkan bahwa kecepatan roda kanan dan kiri robot berubah secara teratur. Hasil perhitungan kecepatan menunjukkan nilai kecepatan roda kiri lebih besar dari 10 cm/s (nilai kecepatan maksimum robot) sehingga pada aplikasinya, perhitungan nilai PWM akan *overflow* karena lebar bidang PWM yang digunakan hanya 8 bit (255), sebagaimana ditunjukkan Gambar 16 berikut :



Gambar 16 Diagram kecepatan roda kanan dan roda kiri hasil percobaan pada robot

Pada Gambar 16 di atas terlihat bahwa pada saat tertentu, nilai kecepatan roda kiri (warna biru) berubah dengan drastis. Hal itu disebabkan karena nilai hasil perhitungan kecepatan yang diumpankan ke PWM melebihi lebar bidang PWM yang digunakan. Berikut perbandingan lintasan robot berdasarkan pengamatan dan simulasi :

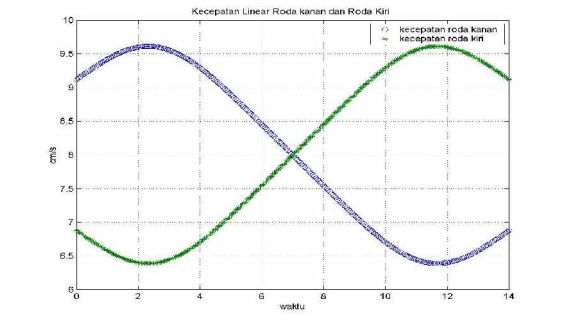


Gambar 17 Perbandingan lintasan berdasarkan perhitungan kecepatan simulasi dan mikro

Pada Gambar 17 terlihat bahwa lintasan robot benar-benar berbeda dengan lintasan simulasi. Hal ini disebabkan karena PWM yang digunakan hanya memiliki lebar 8 bit saja, sehingga terjadi kesalahan perhitungan di mikro.

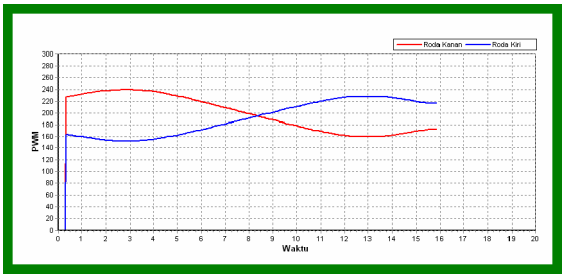
4.2.3 Simulasi dan Percobaan 3

Setelah beberapa simulasi dan percobaan, respon kecepatan robot menunjukkan nilai yang cukup baik dan bentuk lintasan dengan kesalahan gerak yang minimal untuk konfigurasi titik masukan dengan rumusan sebagai berikut : untuk koordinat titik akhir yang akan dituju P3(Xfinal,Yfinal), titik-titik kontrol yang memberikan hasil akhir dengan kesalahan minimal adalah P1(Xfinal/2,0), P2(Xfinal/2,Yfinal). Berikut hasil simulasi untuk titik P1(50,0), P2(50,50), P3(100,50) dan kecepatan referensi 8 cm/s :



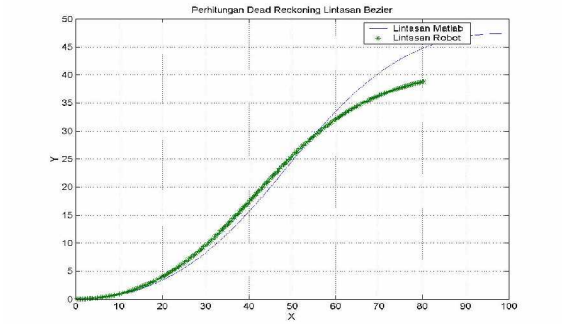
Gambar 18 Diagram kecepatan roda kanan dan roda kiri hasil simulasi

Gambar 18 menunjukkan simulasi respon kecepatan untuk titik masukkan P1(50,0), P2(50,50), P3(100,50) dan kecepatan referensi 8 cm/s. Pada percobaan robot didapatkan respon kecepatan sebagai berikut :



Gambar 19 Diagram kecepatan roda kanan dan roda kiri hasil percobaan pada robot.

Berdasarkan Gambar 19, terlihat bahwa respon kecetan robot cukup baik. hal ini berdasarkan pengamatan bahwa robot berjalan dengan lembut dan pada sesuai dengan titik yang dimasukkan. Berikut gambar lintasan robot :



Gambar 20 Perbandingan lintasan berdasarkan perhitungan kecepatan simulasi dan mikro

Berdasarkan pengamatan, robot berjalan dengan lembut (*smooth*) dan berjalan pada lintasan yang sesuai, yang sudah dibangkitkan. Tetapi terdapat sedikit kesalahan pencapaian titik final. Hal ini terjadi karena beberapa hal, yaitu robot menggunakan sistem kendali kalang terbuka sehingga tidak ada umpan balik kecepatan dan posisi, serta motor dc sebagai penggerak roda kanan dan roda kiri, memiliki perbedaan karakteristik.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan, pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan hal-hal sebagai berikut :

1. Metode PCBC (*piecewise cubic bezier curve*) mampu membangkitkan lintasan (trayektori) robot dengan baik pada koordinat di kuadran I.
2. Respon robot sangat tergantung pada masukan titik yang dijadikan sebagai titik referensi robot untuk membangkitkan lintasan (trayektori) dengan metode PCBC.
3. Tidak semua titik dapat dijadikan titik referensi untuk membangkitkan trayektori robot berbasis

PCBC, sehingga untuk menyederhanakan masukan titik agar kesalahan minimal, digunakan perumusan sebagi berikut : untuk koordinat titik akhir yang akan dituju P3(Xfinal,Yfinal), titik-titik kontrol yang memberikan hasil akhir dengan kesalahan minimal adalah P1(Xfinal/2,0) dan P2(Xfinal/2,Yfinal).

4. Karakteristik motor penggerak dan lebar PWM mempengaruhi ketepatan robot dalam bergerak menuju titik yang dituju.
5. Permukaan rintangan yang tidak rata mempengaruhi ketepatan jarak robot untuk berhenti di depan rintangan.

5.2 SARAN

Untuk pengembangan sistem lebih lanjut, maka penulis memberikan saran-saran sebagai berikut:

1. Dapat digunakan sistem kalang tertutup dengan sensor kecepatan dan posisi sebagai umpan balik, sehingga ketepatan robot dalam mencapai titik akhir lebih baik.
2. Robot mobil dapat dikembangkan untuk menghindari rintangan dan kemudian kembali lagi ke lintasan bezier.
3. Lintasan yang dapat dibangkitkan dapat ditingkatkan untuk koordinat pada kuadran yang lain.

DAFTAR PUSTAKA

[1] Bourke, Paul. **" Bezier Curve".** Internet. 2006.

[2] Bourke, Paul. **"Picewise Cubic Bezier Curve"**. Internet. 2006.

[3] Fleury, Sara. Soueres. Jean-Paul Laumond. Raja Chatila. **"Primitives for Smoothing Mobile Robot Trayektory".** Internet. 2006.

[4] Hartono, Jogiyanto. "**Konsep Dasar Pemrograman Bahasa C".** Penerbit Andi. Yogyakarta. 2000.

[5] Hwang, Jung-Hoon. Ronald C Arkin. Dong-Soo Kwon. "**Mobile robot at your fingertip: Bezier curve on-line trayektory generation for supervisory control".** Internet. 2006

[6] Lucas, GW. **"A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators".** Internet. 2006.

[7] Nagy, Bryan. Alonso Kelly. **"Trayektory Generation For Car-Like Robots Using Cubic Curvature Polynomials"**. Internet. 2006.

[8] Setiawan, Iwan. Trias A. Darjat. **"Rancang Bangun Sistem Kontrol Robot Mobil untuk Keperluan Navigasi Darat Berbasis Trayektori Bezier".** Universitas Diponegoro. 2006

[9] Wardhana L. 2006. **"Belajar Sendiri Mikrokontroler AVR Seri ATMega8535 Simulasi, Hardware dan Aplikasi".** Penerbit Andi: Yogyakarta.

[10] www.paralax.com

[11] --------- . "**ATMega8535(L) Preliminary Complete"**. Atmel Corporation http://atmel.com/dyn/resource/prod_documents/doc2502.pdf.

[12] ----------. **"Curvature".** Internet. 2006.

[13] --------- . "**Liquid Crystal Display Module M1632".** *User Manual*. Seiko Instrument Inc. Japan. 1987.

[14] ----------. **"The Math Behind the Bezier Curve"**. Internet. 2006.

BIODATA MAHASISWA

Arif Dwi Utomo
(L2F 002 558)

Mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro, mengambil pilihan kosentrasi kontrol.

Mengetahui/Mengesahkan,

Pembimbing I	Pembimbing II
Iwan Setiawan,ST,MT	**Trias Andromeda,ST,MT**
NIP. 132 283 183	**NIP. 132 283 185**