KENDALI KECEPATAN DAN POSISI PADA MOBILE ROBOT YANG MENGGUNAKAN TRIANGLE OMNI-DIRECTIONAL WHEELS DENGAN **METODE PID**

TUGAS AKHIR



Oleh:

: Aditya Wiguna Saputra Nama

: 10.41020.0081 NIM

Program : S1 (Strata Satu)

: Sistem Komputer Jurusan

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA DAN TEKNIK KOMPUTER

SURABAYA

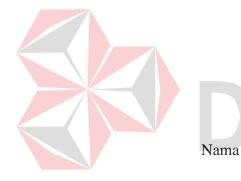
2014

KENDALI KECEPATAN DAN POSISI PADA MOBILE ROBOT YANG MENGGUNAKAN TRIANGLE OMNI-DIRECTIONAL WHEELS DENGAN METODE PID

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



UNIVERSITAS

Oleh

: Aditya Wiguna Saputra

NIM

: 10.41020.0081

Program

: S1 (Strata Satu)

Jurusan

: Sistem Komputer

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN TEKNIK KOMPUTER SURABAYA

2014

KENDALI KECEPATAN DAN POSISI PADA MOBILE ROBOT YANG MENGGUNAKAN TRIANGLE OMNI-DIRECTIONAL WHEELS DENGAN METODE PID

Dipersiapkan dan disusun oleh

Nama : Aditya Wiguna Saputra

NIM : 10.41020.0081

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji pada : 29 Agustus 2014

Susunan Dewan Penguji

| Pembi | mbing |
|-------|--------------------------------------|
| I. | Harianto, S.Kom., M.Eng. UNIVERSITAS |
| II. | I Dewa Gede Rai Mardiana, S.Kom. |
| Pengu | ji |
| I. | Helmy Widyantara, S.Kom, M.Eng. |
| II. | Dr. Jusak |

Tugas Akhir ini telah diterima sebagai salah satu persyaratan Untuk memperoleh gelar sarjana

<u>Pantjawati Sudarmaningtyas, S.Kom.,M.Eng.</u> Pembantu Ketua Bidang Akademik

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN TEKNIK KOMPUTER SURABAYA

PERNYATAAN

Dengan ini saya menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam Tugas Akhir ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya Tugas Akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Surabaya, 29 Agustus 2014



KATA PENGANTAR

Pertama-tama penulis panjatkan puji syukur kehadirat Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya akhirnya penulis dapat menyelesaikan laporan Tugas Akhir ini dengan sebaik-baiknya. Penulis membuat laporan Tugas Akhir yang berjudul kendali kecepatan dan posisi pada *mobile robot* yang menggunakan *triangle omni-directional wheels* dengan metode pid. Sistem ini diharapkan dapat memberikan suatu informasi yang bermanfaat mengenai kadar gas berbahaya di sekitar kita.

Dalam pelaksanaan Tugas Akhir maupun pembuatan laporan Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak. Oleh karena itu, penulis juga ingin mengucapkan terima kasih kepada:

- 1. Allah SWT yang selalu melimpahkan berkat dan rahmat sehingga penulis dapat menyelesaikan tugas akhir ini.
- 2. Orang tua beserta semua keluarga tercinta yang selalu memberikan doa, motivasi, inspirasi dan semangat kepada penulis. Terutama kepada ibu penulis yang begitu besar memberikan semangat kepada penulis hingga dapat menyelesaikan Tugas Akhir ini.
- 3. Bapak Harianto, S.Kom., M.Eng selaku dosen pembimbing I yang telah membimbing, memotivasi dan mengarahkan penulis dalam menyelesaikan tugas akhir ini.
- 4. Bapak I Dewa Gede, S.Kom. selaku dosen pembimbing II yang telah membimbing, memotivasi, dan mengarahkan penulis dalam menyelesaikan tugas akhir ini.
- Keluarga Besar SK yang telah memberikan inspirasi dan kenangan yang tiada ujung selama di STIKOM.
- 6. Semua kolega yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam laporan Tugas Akhir ini masih banyak kekurangan. Oleh karena itu penulis memohon maaf apabila terdapat kesalahan dan selalu menerima kritik, saran dan masukan dari pembaca. Semoga karya penulis dapat memberikan manfaat positif bagi para pembaca.

Surabaya, 29 Agustus 2014

Penulis



ABSTRAKSI

Mobile robot adalah robot yang memiliki mekanisme penggerak berupa roda (wheel) atau kaki (leg) untuk dapat berpindah tempat dari suatu tempat ke tempat yang lain pada umumnya mobile robot memiliki desain yang konvensional yang mana desain ini memiliki keterbatasan akan sudut gerak sehingga membutuhkan area yang lebih luas untuk berbelok. Perkembangan teknologi dalam mobile robot ini berkembang dengan sangat pesat terutama dalam metode kontrol yang digunakan untuk mengendalikan kecepatan gerak mobile robot salah satu motede yang sering digunakan yaitu metode PID (Proporsional-Integratif-Derivatif) dan metode Fuzzy.

Mobile robot yang menggunakan triangle omni-directional wheels memiliki keunggulan dalam hal mobilitas dilingkungan padat untuk dapat bergerak kesegalah arah tanpa harus berputar ditempat terlebih dahulu (rafiudin, 2012), sehingga dapat menghemat waktu tempuh dalam sebuah pergerakkan dan mampu bergerak ditempat yang memiliki ruang gerak yang lebih sedikit. Metode pengendalian kecepatan putaran dari motor DC pada mobile robot yang menggunakan metode PID ini memiliki kelebihan secara matematis daripada metode fuzzy.

Dari hasil pengujian ini didapatkan bahwa *respone* kesalahan rata-rata yang dimiliki oleh *mobile robot* ini adalah sebesar 3.5885 % terhadap sumbu X, 3.8645% terhadap sumbu Y dan sebesar 3.6205% terhadap orientasi sudut yang dibuat. Dimana nilai dari kesalahan orientasi sudut tersebut didapatkan dari perhitungan secara matematis, Data ini didapatkan dari 40 kali percobaan untuk setiap *input* posisi tujuan yang berbeda. Sesuai dengan metode *Ziegler Nichols* yang pertama yaitu osilasi maka didapatkan nilai untuk parameter-parameter yang sesuai dengan pengendalian *mobile robot* ini adalah dengan nilai Kp = 2.8, Ki = 5.6 dan Kd = 0.84.

DAFTAR ISI

| ABSTRA | KSI | | V |
|---------|------|---|------|
| KATA PE | ENGA | NTAR | vi |
| DAFTAR | ISI | | viii |
| DAFTAR | TABI | EL | xii |
| DAFTAR | GAM | BAR | xiii |
| BAB I | PEN | DAHULUAN | 1 |
| | 1.1. | Latar Belakang Masalah | 1 |
| | 1.2. | Perumusan Masalah | 2 |
| | 1.3. | Pembatasan Masalah | 2 |
| | 1.4. | TujuanUNIVERSITAS | 2 |
| | 1.5. | Sistematika Penulisan | _3 |
| BAB II | LAN | DASAN TEORI | 5 |
| | 2.1. | Magnetic Encoder | 5 |
| | 2.2. | Driver Motor | 6 |
| | | 2.2.1. <i>Driver L298H</i> | 7 |
| | | 2.2.2. <i>Relay</i> | 7 |
| | 2.3. | Metode PID (Proportional Integratif Derivative) | 9 |
| | | 2.3.1. Kontrol <i>Proporsional</i> | 10 |
| | | 2.3.2. Kontrol Integratif | 10 |
| | | 2.3.3. Kontrol <i>Derivatif</i> | 11 |
| | | 2.3.4. Metode Tunning Ziegler-Nichols | 12 |
| | 2.4. | Omni-directional Wheels | 14 |

| | | 2.4.1. | Roda | Omni | 15 |
|---------|------|---------|-----------|--|----|
| | | 2.4.2. | Sejara | h Roda <i>Omni</i> | 15 |
| | | 2.4.3. | Holon | omic dan non-holonomic robot | 17 |
| | | 2.4.4. | Kinen | natika <i>Omni Wheels</i> | 18 |
| | 2.5. | Mikrol | kontroll | er ATMEGA-32 | 22 |
| | | 2.5.1. | Konst | ruksi Mikrokontroler | 24 |
| | 2.6. | Wiznet | WIZ1 | 10SR | 26 |
| | | 2.6.1. | Fitur V | Jtama | 27 |
| | | 2.6.2. | Konfi | gurasi Jaringan | 28 |
| BAB III | MET | ODE P | ENELI | ΓΙΑΝ | 32 |
| | 3.1. | Alat da | an Baha | n Penelitian | 33 |
| | | 3.1.1. | Alat P | enelitian | 33 |
| | | 3.1.2. | Bahan | Penelitian | 34 |
| | 3.2. | Jalan I | Penelitia | an | 35 |
| | | 3.2.1. | Pengu | mpulan Literatur | 35 |
| | | 3.2.2. | Perano | eangan Perangkat Keras | 35 |
| | | | A. | Perancangan Badan Robot | 35 |
| | | | B. | Perancangan Rangkaian Minimum Sistem | 37 |
| | | | C. | Perancangan Rangkaian Driver Motor (L298H) |) |
| | | | | | 39 |
| | | | D. | Perancangan Rangkaian Regulator | 40 |
| | | | E. | Perancangan Rangkaian MAX232 | 42 |
| | | 3.2.3. | Perano | cangan Perangkat Lunak | 43 |
| | | | A. | Konfigurasi interfaces WIZ110SR | 43 |

| | | | B. | Konfigurasi Winsock Pada Visual Basic 6.0 | 45 |
|--------|------|---------|---------|--|------|
| | | | C. | Konfigurasi Serial pada Mikrokontroler | 46 |
| | | | D. | Perancangan Program PID Pada Mikrokontrole | er |
| | | | | | 47 |
| | | | E. | Perancangan Program pengerak Mobile robot | |
| | | | | | 49 |
| | | | F. | Perancangan Program Pengerak Omni-Direction | onal |
| | | | | | 52 |
| | 3.3. | Langk | ah Pen | gujian | 56 |
| | | 3.3.1. | Pengu | ijian Rangkaian <i>Regulator</i> | 57 |
| | | 3.3.2. | Pengu | ijian Rangkaian <i>Driver</i> Motor | 58 |
| | | 3.3.3. | Pengu | ijian Rangkaian <i>Minimum</i> Sistem | 59 |
| | | 3.3.4. | Pengu | ıjian Rangkaian MAX232 dan pengirima <mark>n</mark> serial | |
| | | | | | 60 |
| | | 3.3.5. | Pengu | ijian Program PID | 62 |
| | | 3.3.6. | Pengu | ijian Gerak <i>Omni-directional</i> | 63 |
| | | 3.3.7. | Pengu | ijian Sistem Secara Keseluruhan | 64 |
| BAB IV | HAS | IL DAN | I PEM | BAHASAN | 65 |
| | 4.1. | Hasil I | Penguji | an Perangkat Keras | 65 |
| | | 4.1.1. | Hasil | Pengujian Rangkaian Regulator | 65 |
| | | 4.1.2. | Hasil | Pengujian Rangkaian Driver Motor | 66 |
| | | 4.1.3. | Hasil | Pengujian Rangkaian Minimum Sistem | 67 |
| | 4.2. | Hasil I | Penguji | an Perangkat Lunak | 69 |
| | | 4.2.1. | Hasil | Pengujian Progam Pengirimanan Serial | 69 |

| | | 4.2.2. | Hasil Pengujian metode PID | 70 |
|---------|------|---------|---|----|
| | | 4.2.3. | Hasil Pengujian Gerak Omni-Directional | 76 |
| | 4.3. | Hasil I | Pengujian Sistem Secara Keseluruhan | 77 |
| | | 4.3.1. | Gerak Mobile Robot pada Kuadran Pertama | 77 |
| | | 4.3.2. | Gerak Mobile Robot pada Kuadran kedua | 79 |
| | | 4.3.3. | Gerak Mobile Robot pada Kuadran ketiga | 82 |
| | | 4.3.4. | Gerak Mobile Robot pada Kuadran Keempat | 84 |
| | | 4.3.5. | Pembahasan Keseluruhan | 86 |
| BAB V | PEN | UTUP | | 87 |
| | 5.1. | Kesim | pulan | 88 |
| | 5.2. | Saran | | 89 |
| DAFTAR | PUST | ГАКА | UNIVERSITAS | 90 |
| LAMPIRA | AN | | UNIVERSITAS | 91 |
| | | | | |
| | | | | |

DAFTAR TABEL

| Tabel 2.1 Penalaran Ziegler-Nichols Metode ke-1 | 13 |
|---|----|
| Tabel 2.2 Penalaran Ziegler-Nichols Metode ke-2 | 14 |
| Tabel 3.1 Pin I/O Minimum Sistem | 36 |
| Tabel 3.2 Pin Input/Output Driver Motor L298H | 39 |
| Tabel 3.3 Pin Input/Output Max232 | 43 |
| Tabel 3. Format Pengiriman Data Serial | 50 |
| Tabel 3.5 Koordinat Wilayah Mobile robot | 54 |
| Tabel 3.6 Pengujian Rangkaian Regulator | 58 |
| Tabel 3.7 Pengujian Rangkaian <i>Driver</i> Motor | 59 |
| Tabel 3.8 Pengujian Rangkaian <i>Minimum</i> Sistem | 60 |
| Tabel 3.9 Pengujian Rangkaian MAX232 dan pengiriman serial | 62 |
| Tabel 3.10 Pengujian Program PID | 63 |
| Tabel 3.11 Pengujian Omni-Directional | 64 |
| Tabel 3.12 Pengujian Omni-Directional | 65 |
| Tabel 4.1 Hasil Pengujian Rangkaian Regulator | 66 |
| Tabel 4.2 Sampel Data PID Motor 1 | 75 |
| Tabel 4.3 Gerak <i>Mobile</i> Robot pada Kuadran Pertama | 79 |
| Tabel 4.4 Presentasi Kesalahan Gerak Robot Pada Kuadran Pertama | 79 |
| Tabel 4.5 Gerak <i>Mobile</i> Robot pada Kuadran Kedua | 80 |
| Tabel 4.6 Presentasi Kesalahan Gerak Robot Pada Kuadran Kedua | 81 |
| Tabel 4.7 Gerak <i>Mobile</i> Robot pada Kuadran Ketiga | 82 |
| Tabel 4.8 Presentasi Kesalahan Gerak Robot Pada Kuadran Ketiga | 82 |
| Tabel 4.9 Gerak Mobile Robot pada kuadran ke Empat | 83 |
| Tabel 4.10 Presentasi Kesalahan Gerak Robot Pada Kuadran Ke empat | 84 |

DAFTAR GAMBAR

| Gambar 2.1 Magnetic Encoder | 5 |
|---|----|
| Gambar 2.2 Rotary Encoder | 6 |
| Gambar 2.3 L298H | 7 |
| Gambar 2.4 Relay | 8 |
| Gambar 2.5 Blok Diagram PID | 9 |
| Gambar 2.6 Kurva Tangapan Berbentuk S | 13 |
| Gambar 2.7 Karakteristik Keluaran Suatu Sistem dengan Penambahan Kp | 14 |
| Gambar 2.8. Omni-directional Wheels | 15 |
| Gambar 2.9 Macam-macam roda omni | 15 |
| Gambar 2.10. Penggambaran roda omni | 16 |
| Gambar 2.11 Salah satu aplikasi roda <i>omni</i> | 16 |
| Gambar 2.12 Non-Holonomic | 17 |
| Gambar 2.13 Holonomic | 18 |
| Gambar 2.14 Referensi vektor posisi | 18 |
| Gambar 2.15 Referensi Matrix Rotasi | 19 |
| Gambar 2.16 Referensi Kinematic Omni Wheels | 20 |
| Gambar 2.17 Konfigurasi <i>pin</i> ATmega32 | 25 |
| Gambar 2.18 Wiznet WIZ110SR | 27 |
| Gambar 3.1 Blok Diagram Sistem Keseluruhan | 32 |
| Gambar 3.2 Desain Badan Robot | 36 |
| Gambar 3.3 Bentuk Badan Robot | 36 |
| Gambar 3.4 Skematik rangkaian dari <i>minimum</i> sistem Atmega32 | 38 |
| Gambar 3.5 Rangkaian <i>Driver</i> Motor L298H | 39 |
| Gambar 3.6 Skematik Rangkaian Regulator | 41 |

| Gambar 3.7 Rangkaian Max232 | 42 |
|---|----|
| Gambar 3.8 Pengaturan jaringan dari WIZ110SR | 44 |
| Gambar 3.9 Pengaturan pengiriman serial dari WIZ110SR | 44 |
| Gambar 3.10 Pengaturan winsock pada Visual Basic 6.0 | 46 |
| Gambar 3.11 Konfigurasi USART pada CVAVR | 47 |
| Gambar 3.12 Flowchart Program Mikrokontroler | 50 |
| Gambar 3.13 Desain triangle omni-directional wheels | 53 |
| Gambar 3.14 Jarak dan sudut tempuh dari robot | 54 |
| Gambar 3.15 Langkah-Langkah Pengujian | 57 |
| Gambar 4.1 Pengujian Rangkaian Regulator | 66 |
| Gambar 4.2 Motor DC Berputar Kearah Kiri | 67 |
| Gambar 4.3 Motor DC Berputar Kearah Kanan | 68 |
| Gambar 4.4 Perbedaan Kecepatan Putaran Motor | 68 |
| Gambar 4.5 Tampilan Chip Signature | 69 |
| Gambar 4.6 Tampilan <i>Download</i> Program | 69 |
| Gambar 4.7 Pengiriman dari mikrokontroler ke Laptop | 70 |
| Gambar 4.8 Hasil Pengiriman Intruksi Data Satu | 70 |
| Gambar 4.9 Hasil Pengiriman Intruksi Data Dua. | 71 |
| Gambar 4.10 Sinyal Berosilasi Terus Menerus | 72 |
| Gambar 4.11 Control P (Proporsional) | 72 |
| Gambar 4.12 Kontrol PI (<i>Proporsional-Integral</i>) | 73 |
| Gambar 4.13 Control PD (Proporsional-Derivatif) | 74 |
| Gambar 4.14 Control PID (Proporsional-Integral-Derivatif) | 76 |
| Gambar 4.15 Posisi Awal Mobile Robot | 77 |
| Gambar 4 16 Posisi Setelah Mobile Robot Bernutar | 78 |

BABI

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi dalam dunia robotika semakin berkembang pesat, terutama dalam penggunaan metode sistem kontrol pada robot dan sistem pergerakkan dari robot. Penggunaan metode kontrol tersebut digunakan untuk mengendalian pergerakan robot dan mengoptimalkan waktu tempuh dari robot tersebut sehingga robot tersebut dapat melaksanakan apa yang ditugaskan terhadap robot tersebut.

Pada umumnya kebanyakan robot dibuat dengan desain yang konvensional yang mana desain ini memiliki keterbatasan akan sudut gerak sehingga membutuhkan area yang lebih luas untuk berbelok. Menurut Diegel, 2002 dalam sistem robot yang memiliki desain omnidirectional platform memiliki keunggulan besar dibanding desain konvensional pada umumnya dalam hal mobilitas di lingkungan padat sehingga dapat bergerak dengan leluasa kesegala arah. Untuk mengendalikan kecepatan putaran motor DC banyak metode yang dapat digunakan dimana salah satunya yaitu fuzzy. Dimana Bachri, 2004 menyatakan bahwa Untuk mendapatkan hasil yang optimal, maka kita dapat men-tuning parameter fuzzy control tersebut dengan cara try and error yaitu mengatur (adjust) membership function (range e-max dan demax) serta rules-rules yang ada, semua parameter tersebut akan dikelolah oleh mikrokontroller sehingga membutuhkan waktu dan proses eksekusi yang panjang. Hal ini menyebabkan pengendalian robot dengan omnidirectional platform yang memanfaatkan metode fuzzy memiliki respon yang lambat untuk mencapai posisi yang ditargetkan.

Maka dari itu dalam penelitian ini akan dibuat sebuah sistem kendali yang mempunyai algoritma yang tidak membutuhkan banyak pengelolahan data dan memiliki ketahanan

terhadap disturbance serta mempunyai respon yang cepat dan akurat, sehingga pengendalian mobile robot dengan omnidirectional platform ini nantinya dapat bergerak sesuai dengan yang diinginkan. Maka dari itu dalam hal ini metode yang dipilih untuk mengendalikan robot adalah metode PID. Karena metode ini dapat menghasilkan output dengan risetime yang tinggi dan kesalahan yang kecil (Liu, 2007).

1.2 Perumusan Masalah

Adapun rumusan masalah dalam pengerjaan penelitian ini adalah.

- 1. Bagaimana menentukan dan mendesain parameter parameter pada metode PID?
- 2. Bagaimana mengendalikan kecepatan dan arah dari tiga buah motor DC?
- 3. Bagaimana mengatur posisi dari *mobile robot* yang menggunakan *triangle omni-directional wheels platform*?

1.3 Pembatasan Masalah

Dalam perancangan dan pembuatan alat ini, terdapat beberapa batasan masalah, antara lain:

UNIVERSITAS

- 1. Proses kontrol PID dilakukan oleh mikrokontroler ATmega32.
- 2. Plan sitem PID menggunakan motor DC dengan magnetik encoder.
- 3. Tidak ada halangan yang menggangu pergerakkan robot.

1.4 Tujuan

Adapun tujuan dari pembuatan kendali robot dengan metode PID ini yaitu agar dapat menentukan parameter – parameter yang ada pada PID sehingga kecepatan putaran motor DC bisa konstant (tetap) pada ketiga buah motor DC yang ada, serta mengendalikan *mobile robot* yang memiliki jenis roda *triangle omni directional* agar dapat bergerak menuju posisi atau koordinat yang ditentukan.

1.2 Sistematika Penulisan

Sistematika penulisan buku Tugas Akhir ini terdiri dari lima bab, dimana dalam tiap bab terdapat beberapa sub-bab. Ringkasan uraian dari tiap bab tersebut adalah sebagai berikut:

BAB I Pendahuluan

Pada bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan buku Tugas Akhir.

BAB II Landasan Teori

Pada bab ini menjelaskan tentang beberapa teori tentang komponen dan sistem pengontrolan yang digunakan dalam pengerjaan Tugas Akhir ini. Diantaranya adalah sensor *magnetic encoder*, mikrokontroler ATMEGA32, metode PID, *driver Motor* DC, *omni-directional wheels*, modul komunikasi *wireless* wiznet wiz110sr, *microcontroller* AVR, komunikasi USART.

BAB III Metode Penelitian

Bab ini berisi tentang penjelasan penulis dalam merancang dan membuat perangkat keras dan perangkat lunak. Dalam bab ini juga menjelaskan tentang cara kerja dari perangkat keras, seperti rangkaian yang digunakan pada sensor *magnetic encoder*, *driver Motor* DC, rangkaian *power suplay*, rangkaian *minimum system*, dan rangkaian komunikasi antara mikrokontroler dan komputer. Di bab ini juga di jelaskan tentang metode pengujian dan evaluasi yang digunakan.

BAB IV Pengujian dan Evaluasi Sistem

Bab ini berisi tentang pengujian dan evaluasi terhadap sensor yang digunakan yaitu sensor *magnetic encoder* untuk mendeteksi jumlah putaran dan kecepatan dari motor DC, metode PID yang digunakan untuk mengontrol

kecepatan motor dan pengujian terhadap posisi koordinat tujuan. Kemudian dilakukan pengujian pengiriman data sensor dari *microcontroller* ke komputer melalui komunikasi nirkabel menggunakan *wiznet* WIZ110SR.

BAB V Penutup

Pada bab ini menjelaskan tentang kesimpulan dari Tugas Akhir yang telah dikerjakan dan saran-saran yang diberikan oleh penulis.





BAB II

LANDASAN TEORI

2.1. Magnetic Encoder

Pada umumnya *magnetic encoder* berkerja sama persis dengan *rotary encoder* yang hasil keluaranya berupa sinyal pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *magnetic encoder* untuk diteruskan oleh rangkaian kendali seperti driver motor dan lain-lainnya. *magnetic encoder* umumnya digunakan pada pengendalian robot, pengendalian kecepatan *driver motor*, dan lain sebagainya (http://www.news.thomasnet.com, 2013).



Gambar 2.1. *Magnetic Encoder* (http://www.news.thomasnet.com, 2014)

Rotary Encoder dikenal sebagai salah satu sensor kecepatan, karena pada dasarnya bentuk fisik dari rotary encoder adalah suatu lempengan yang berbentuk lingkaran yang disalah satu busurnya mempunyai celah, dan celah ini yang dimanfaatkan sebagai tempat untuk memutus sinyal yang diberikan dalam bentuk cahaya, karena lempengan ini diletakkan di tengah antara sumber cahaya dan juga penerima cahaya, ini artinya setiap lempengan inti berputar tentunya dalam satu kali periode putaran penuh tentu akan satu kali sumber cahaya akan mengenai penerima cahaya, waktu yang diperlukan lempengan ini dalam berputar untuk

mengenai cahaya ini lah yang akan digunakan dalam perhitungan untuk menentukan kecepatan putaran.



Gambar 2.2. *Rotary Encoder* (http://id.autonics.com, 2014)

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau device berputar yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi.

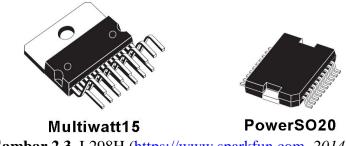
2.2. Driver Motor

Driver motor adalah sebuah rangkaian elektronika yang digunakan untuk mengerakkan kecepatan dari putaran sebuah motor dan menentukan gerak dari motor tersebut, jenis-jenis *driver* motor sangat banyak sekali semua itu bergantung pada jenis motor yang akan dikendalikan. Adapun beberapa jenis dari *driver* motor yaitu (fahmi, 2012).

2.2.1. L298

L298 adalah jenis IC driver motor yang dapat mengendalikan arah putaran dan kecepatan motor DC ataupun Motor stepper. Mampu mengeluarkan output tegangan untuk

Motor dc dan motor stepper sebesar 50 volt. IC 1298 terdiri dari *Transistor-Transistor Logik* (*TTL*) dengan gerbang nand yang memudahkan dalam menentukkan arah putaran suatu motor dc dan motor stepper. Dapat mengendalikan 2 untuk motor dc namun pada hanya dapat mengendalikan satu motor *stepper*. Penggunaannya paling sering untuk robot *line follower*. Bentuknya yang kecil memungkinkan dapat meminimalkan pembuatan robot *line follower*.



Gambar 2.3. L298H (https://www.sparkfun.com, 2014)

2.2.2. Relay

Relay adalah suatu piranti yang menggunakan elektromagnet untuk mengoperasikan seperangkat kontak saklar. Susunan paling sederhana terdiri dari kumparan kawat penghantar yang dililit pada inti besi. Bila kumparan ini dienergikan, medan magnet yang terbentuk menarik *armature* berporos yang digunakan sebagai pengungkit mekanisme saklar (fahmi, 2012).

Relay memiliki tiga jenis kutub yaitu common (kutu bacuan), normally close (kutub yang dalam keadaan awal terhubung pada common), dan normally Open (kutub yang pada awalnya terbuka dan akan terhubung dengan common saat kumparan relay diberi arus listrik).

Relay bisa dimanfaatkan sebagai motor *driver* untuk menggerakkan motor DC yang ada pada badan robot. Agar relay dapat bekerja dengan tegangan logika dari mikrokontroler maka diperlukan komponen yang dapat membantu hal tersebut (fahmi, 2012).



Gambar 2.4. Relay (www.omron.co.id, 2014)

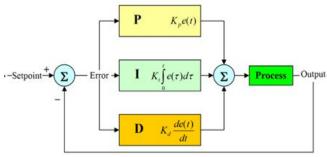
Pulse Width Modulation atau PWM merupakan suatu teknik teknik dalam mengatur kerja suatu peralatan yang memerlukan arus pull in yang besar dan untuk menghindari disipasi daya yang berlebihan dari peralatan yang akan dikontrol. PWM merupakan suatu metode untuk mengatur kecepatan perputaran motor dengan cara mengatur presentase lebar pulsa high terhadap periode dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke motor sebagai sumber daya. Semakin besar perbandingan lama sinyal high dengan periode sinyal maka semakin cepat motor berputar. (Nugroho, 2011).

2.3. Metode PID (Proportional Integratif Derivative)

Proportional Integral Derivative merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tesebut. (Zakaria, 2011)

Menurut Firdaus A, 2010 menjelaskan bahwa supaya kecepatan motor dapat stabil maka harus menggunakan PID dikarenakan metode ini memiliki ketahanan terhadap guncangan dan memiliki respone yang cepat.

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proporsional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant. Dalam waktu kontinyu, sinyal keluaran pengendali PID dapat dilihat seperti blok diagram berikut.



Gambar 2.5. Blok Diagram PID (fahmi, 2012)

Dari blok diagram diatas dapat dituliskan rumus matematisnya sebagai berikut ini.



- KKonstanta *Integratif* dikalikan dengan error pada waktu t
- Delta waktu dt
- K Konstanta Derivative

2.3.1. Kontrol Proporsional

Secara sederhana kontrol *proporsional* saja dapat diuraikan sebagai berikut.

Dimana nilai Kp berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya rise time dan settling time (fahmi, 2012).

2.3.2. Kontrol *Integratif*

Secara sederhana control Inegratif juga dapat diuraikan menjadi bentuk yang sederhana sebagai berikut ini.

Dimana nilai Ki adalah konstanta Integral, dan dari persamaan diatas, G(s) dapat dinyatakan sebagai delta error dikalikan dengan delta time Jika e(t) mendekati konstan (bukan nol) maka u(t) akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika e(t) mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan *respon steady-state*, namun pemilihan Ki yang tidak tepat dapat menyebabkan *respon transien* yang tinggi sehingga dapat menyebabkan ketidak stabilan sistem. Pemilihan Ki yang sangat tinggi justru dapat menyebabkan *output* berosilasi karena menambah orde *system* (fahmi, 2012).

2.3.3. Kontrol Derivatif

Secara sederhana control Inegratif juga dapat diuraikan menjadi bentuk yang sederhana sebagai berikut ini.

$$S K = \frac{e}{} (2.4)$$

Dari persamaan di atas, dapat dilihat dimana delta error dibagi dengan nilai dari delta waktu yang mengakibatkan bahwa kontrol D ini dapat mempengaruhi dalam konteks "kecepatan" atau rate dari error. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi kesalahan yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan kesalahan sehingga saat kesalahan statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri (fahmi, 2012).

Berdasarkan hal ini maka dapat dinyatakan pengaruh yang terjadi pada sebuah sistem apabila menggunakan metode PID ini adalah sebagai berikut.

1. Kontrol P (*Proporsional*)

Pada kontrol *proporsional* ini mengakibatkan perubahan terhadap sistem antara lain :

- 1) Menambah atau mengurangi kestabilan dengan menambah atau mengurangi nilai konstanta K_P (Kontrol *Proporsional*).
- 2) Dapat memperbaiki respon transien khususnya: rise time, settling time
- 3) Mengurangi *error steady state* (ESS). Untuk mengurangi ESS, dibutuhkan Kp besar, yang akan membuat system lebih tidak stabil. Kendali proporsional memberi pengaruh langsung (sebanding) pada *error*. Semakin besar *error*, semakin besar sinyal kendali yang dihasilkan pengendali.

2. Kontrol I (intergral)

Pada kontrol integral ini mengakibatkan pengaruh terhadap sistem antara lain:

- 1) Mengurangi error steady state (ESS)
- 2) Respon lebih lambat (dibandingkan dengan P)
- 3) Dapat menambah ketidakstabilan (karena menambah orde pada sistem).

Perubahan sinyal kontrol sebanding dengan perubahan error. Semakin besar error, semakin cepat sinyal kontrol bertambah atau berubah.

3. Kontrol D (*Derivative*)

Kontrol derivative memiliki pengaruh terhadap sistem antara lain :

- 1) Memperbaiki respon transien, karena memberikan aksi saat ada perubahan *error*.
- hanya berubah saat ada perubahan error, sehingga saat ada error statis D tidak beraksi.
 Sehingga D tidak boleh digunakan sendiri

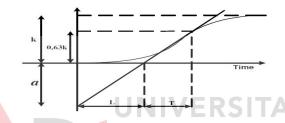
Besarnya sinyal kontrol sebanding dengan perubahan *error* (e). Semakin cepat *error* berubah, semakin besar aksi kontrol yang ditimbulkan (Wicaksono, 2004).

2.3.4. Metode *Tunning Ziegler-Nichols*

Metode tunning *Ziegler-Nichols* merupakan suatu metode untuk menentukan nilai dari parameter-parameter yang ada pada kontrol PID, cara untuk menentukan parameter PID ini ada dua cara yang dapat digunakan berdasarkan metode *Ziegler-Nichols* yaitu sebagai berikut.

A. Metode Ke-1 Ziegler-Nichols

Metode ke-1 didasarkan pada respon *plant* terhadap masukan tangga dalam kalang terbuka. *Plant* yang tidak mempunyai integrator, menghasilkan kurva tanggapan terhadap masukan tangga seperti kurva huruf S pada Gambar 2.6 Kurva tanggapan *plant* digunakan untuk mencari waktu tunda L dan konstanta waktu T. (eka, 2004)



Gambar 2.6. Kurva Tangapan Berbentuk S

Parameter-parameter yang didapat dari kurva reaksi digunakan untuk menentukan parameterparameter pengendali PID berdasarkan tetapan empiris Zielger-Nichols. Rumusrumus untuk parameter pengendali menggunakan metode kurva reaksi ditabelkan pada Tabel 2.1. (eka, 2004)

Tabel 2.1. Penalaran Ziegler-Nichols metode ke-1.

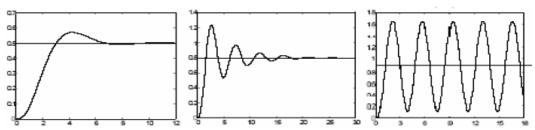
| Pengendali | Kp | Ti | Td |
|------------|-------------|-----|-----|
| P | 1/ <i>a</i> | 12- | |
| PI | 0,9/a | 3L | - |
| PID | 1,2/a | 2L | L/2 |

B. Metode ke-2 Ziegler-Nichols

Pada metode ke-2, penalaan dilakukan dalam kalang tertutup dimana masukan referensi yang digunakan adalah fungsi tangga (*step*). Pengendali pada metode ini hanya pengendali proporsional. Kp, dinaikkan dari 0 hingga nilai kritis Kp, sehingga diperoleh

keluaran yang terus-menerus berosilasi dengan amplitudo yang sama. Nilai kritis Kp ini disebut sebagai *ultimated gain*. (eka, 2004)

Tanggapan keluaran yang dihasilkan pada 2.7 kondisi penguatan proporsional ditunjukkan pada Gambar 2.7. Sistem dapat berosilasi dengan stabil pada saat $K_p = K_u$. (eka, 2004)



Gambar 2.7. Karakteristik keluaran suatu sistem dengan penambahan Kp

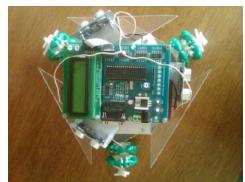
Nilai *ultimated period*, Tu, diperoleh setelah keluaran sistem mencapai kondisi yang terusmenerus berosilasi. Nilai perioda dasar, Tu, dan penguatan dasar, Ku, digunakan untuk menentukan konstanta-konstanta pengendali sesuai dengan tetapan empiris Ziegler-Nichols pada Tabel 2.2. (eka, 2004)

Tabel 2.2. Penalaan Ziegler-Nichols metode ke-2.

| Pengendali | Кр | Ti | Td |
|------------|----------|----------|-----------|
| P | $K_u/2$ | - | - |
| PI | $2K_u/5$ | $4T_u/5$ | - |
| PID | $3K_u/5$ | $T_u/2$ | $3T_u/25$ |

2.4. Omni-directional Wheels

Omni wheels robot adalah merupakan sebuah rancang mobile robot yang menggunakan roda omni (omni wheels) sehingga bisa bergerak ke segala arah, di sudut manapun, tanpa memutar badan robot terlebih dahulu. Beberapa Omni wheels robot menggunakan platform segitiga dan juga persegi empat. Dimana dalam platform segitiga tersebut, mobile robot yang menggunakan tiga buah roda omni jarak setiap bannya akan terpisah sejauh 120° dan pada platform persegi empat, robot omni menggunakan empat buah roda omni.



Gambar 2.8. Omni-directional Wheels (Syam, 2012)

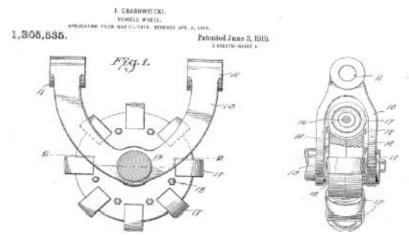
2.4.1 Roda Omni

Roda omni (*omni wheels*) adalah rancangan roda khusus yang tidak hanya mempunyai roda tunggal, tetapi banyak roda dalam satu roda inti.Ada roda inti besar, dan sepanjang tepi ada banyak roda kecil tambahan yang mempunyai sumbu tegak lurus terhadap sumbu roda inti (Syam, 2001).



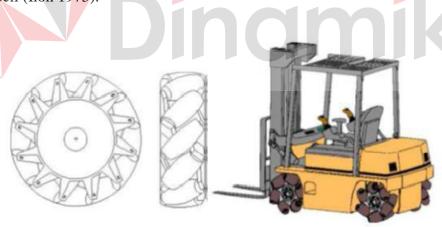
2.4.2 Sejarah Roda Omni

Roda *omni* telah digunakan bertahun tahun dalam dunia robot, industri dan logistik. Sumber utama pengguna terbanyak roda *omni* adalah perusahaan yang memproduksi untuk sistem *konveyor*, seperti untuk menangani paket/barang. Roda *omni* banyak juga digunakan untuk robot *omni*. Sebuah robot *omni* dapat berjalan lurus dari titik A ke titik B juga berputar agar dapat tiba di tujuannya. Roda *omni* juga digunakan untuk kursi roda, kendaraan servis di bandara dan lain-lain. Tidak banyak orang tahu bahwa roda *omni directioal* pertama dipatenkan oleh J. Grobowiecki di Amerika pada tahun 1919. Susunannya terdiri dari sebuah ban/roda utama dan roda *transversial*, penemunya yakin bahwa desain/rancangan dengan roda tersebut dapat berjalan kedepan dan kesamping tanpa mengubah posisi roda.



Gambar 2.10. Penggambaran roda omni (Syam, 2012)

Satu contoh roda *omni directional* modern adalah yang diproduksi oleh penemu Swedia, Bent Ilon sekitar tahun 1973. Bagan/gambar diatas menunjukkan rancangan roda nilon dan sebuah mesin pengangkat (*farklift*) yang dibuat menggunakan roda tersebut. Karakter (*profil*) dari ban tersebut cukup mudah berputar. Roda ini bersifat *omni directional*, tapi gerakan/gaya transversal nya masih menghasilkan gesekan yang cukup besar pada bagian roda yang kecil (Ilon 1975).

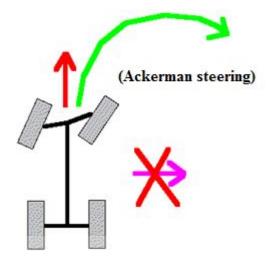


Gambar 2.11. Salah satu aplikasi roda *omni* (Syam, 2012)

2.4.3 Holonomic dan non-holonomic robot

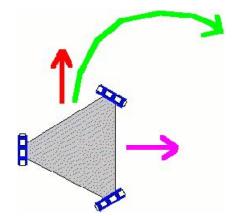
Hanya ada dua jenis mobile robot, robot *holonomic* dan robot *non-holonomic*. *Non-holonomic* robot adalah yang tidak bisa cepat bergerak ke segala arah, seperti mobil. Jenis robot untuk melakukan serangkaian gerakan untuk mengubah posisi. Sebagai contoh, jika

mobil ingin bergerak ke samping, harus melakukan gerakan 'parkir *paralel*' kompleks. Untuk mobil untuk gilirannya, harus memutar roda dan berkendara ke depan. Jenis robot akan '1.5' derajat kebebasan, yang berarti bahwa ia dapat bergerak baik dalam arah X dan Y, tetapi membutuhkan gerakan kompleks untuk mencapai arah X.



Gambar 2.12. Non-Holonomic (Syam, 2012)

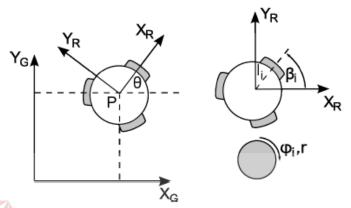
Berbeda dengan sebuah robot *holonomic* yang seketika dapat bergerak ke segala arah. Ini tidak perlu melakukan gerakan yang kompleks untuk mencapai posisi tertentu. Jenis robot akan memiliki dua derajat kebebasan karena dapat bergerak baik di X dan Y bebas (anonim 2006).



Gambar 2.13. Holonomic (Syam, 2012)

2.4.4 Kinematika *Omni Wheels*

Kinematika memberi sarana dalam merubah posisi antar koordinat global dan konfigurasi internal. Dalam rangka menentukan posisi global robot, hubungan ditetapkan diantara kerangka referensi global (XG, YG) dan kerangka lokal instan berpusat pada robot tersebut (XR, YR). Posisi tubuh robot tersebut didapatkan oleh rumus $G = [x, y, \Theta]T$. Pemetaan pergerakan diantara kerangka lokal dan global terkait hanya dengan rotasi matriks (mengingat kerangka lokal ini tetap dan tidak berputar terhadap robot tersebut) (Syam, 2012).

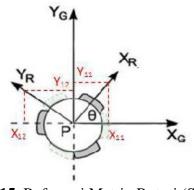


Gambar 2.14. Referensi vektor posisi (Syam, 2012)

Dengan vektor posisi dari omni wheels robot didefenisiskan sebagai berikut.

$$R \theta q \qquad (2.2)$$

Dimana R(θ) adalah matriks rotasi dari mobile robot:



Gambar 2.15. Referensi Matrix Rotasi (Syam, 2012)

$$R(\Theta) = \tag{2.3}$$

Dimana.

$$= xR\cos\Theta = xR\sin\Theta = 0$$

$$= -yR\sin \Theta \qquad = yR\cos \Theta \qquad = 0 \quad (2.4)$$
$$= 0 \qquad = 1$$

Jadi, matriks rotasi-nya yaitu : R
$$(\Theta) = \begin{pmatrix} \cos\theta & s & \theta & 0 \\ -s & \theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
 (2.5)

Sehingga diperoleh vektor posisi sebagai berikut :

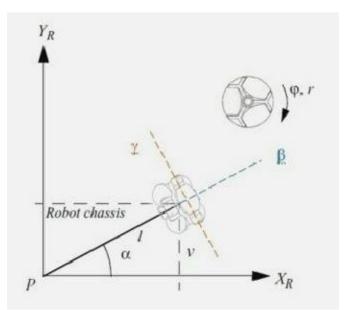
Untuk menentukan vektor kecepatan dari omni wheels robot maka harus diperoleh dari turunan vektor posisi. Kinematika kecepatan dapat diketahui dengan memperhatikan fungsi dari arah pergerakan setiap roda, sumbu robot, kecepatan putaran sudut dan bentuk geometri dari robot itu sendiri (Syam, 2012).

$$\rho$$
, r , θ , φ (2.7)

Sehingga secara umum vektor kecepatan untuk mobile robot dapat dituliskan dengan persamaan:

$$\theta$$
 (2.8)

Dimana merupakan konstribusi atau pengaruh yang berasal dari roda yang ada pada robot.Oleh karena itu, untuk mengetahui pengaruh dari roda, maka perlu diperhatikan bentuk dan jenis roda yang digunakan (Syam, 2012).



Gambar 2.16. Referensi Kinematic Omni Wheels (Syam, 2012).

Dari gambar struktur roda omni maka didapat rumus matriks kecepatan untuk omni wheels robot sebagai berikut.

$$\theta \qquad \qquad \varphi \qquad \qquad (2.9)$$

Dimana J1f adalah matriks hambatan geser, J2 adalah untuk matriks jari-jari roda dan φ adalah kecepatan putaran roda. Persamaan kecepatan didapat dari hubungan antara gerak robot untuk berputar, hambatan geser dan kecepatan putaran roda. Dimana desain dari roda *omni* itu sendiri yang memiliki roda-roda kecil yang ikut berputar sehingga akan timbul hambatan geser. Dimana dalam rancangan *omni* wheels robot ini, roda *omni* yang akan digunakan, masing-masing dipasang pada posisi sudut $(\alpha 1, \alpha 2, \alpha 3) = (\pi/3, \pi, -\pi/3)$, dengan *steering position* masing masing roda $(\beta 1, \beta 2, \beta 3) = 0^{\circ}$, nilai nol diperoleh karena tidak adanya perubahan sudut *steering position* pada roda *omni* yang digunakan. Dalam perhitungan gaya geser dari roda *omni*, maka roda-roda kecil yang ada di sekeliling roda utama harus diperhatikan yaitu antara lain adalah sudut yang terbentuk antara roda-roda kecil dengan porosnya (γ) . Dan roda omni yang digunakan disini memiliki desain dimana roda-roda kecil dan porosnya saling berhimpitan sehingga dalam artian bahwa $(\gamma 1, \gamma 2, \gamma 3) = 0^{\circ}$. Dalam perhitungan persamaan kecepatan dari robot *omni wheels* ini, dapat diketahui dengan

memperhatikan arah pergerakan setiap roda, sumbu robot, kecepatan putaran sudut dan jarijari roda (Syam, 2012).

 $cos\theta$

0

Karena nilai β dan γ untuk jenis roda *omni* adalah 0, maka persamaannya diatas dapat disederhanakan menjadi (Syam, 2012).

Dari persamaan diatas, maka dapat dinyatakan membentuk sederhana untuk mendapatkan jarak tempuh dan sudut tempuh yaitu sebagai berikut.

Jarak Tempuh (c) =
$$\sqrt{}$$
 (2.15)

Sudut Tempuh (
$$\theta$$
) = archtan (x/y) * (180 / 3.14) (2.16)

2.5. Mikrokontroller ATMEGA-32

Mikrokontroler adalah otak dari suatu system elektronika seperti halnya mikroprosesor sebagai otak komputer. Didalam mikrokontroller terdapat memori dan *Port Input/Output* dalam suatu kemasan IC (*Integrated Circuit*). Kemampuannya yang *programmable*, fitur yang lengkap *ADC* (*Analog to Digital Converter*) *internal, EEPROM*

(*Electrically Erasable Programmable Read-Only Memory*) internal, Port I/O, Komunikasi Serial, dll juga harga yang terjangkau memungkinkan mikrokontroler digunakan pada berbagai *system* elektronis, seperti pada robot, automatisasi industri, *system* alarm, peralatan telekomunikasi, hingga peralatan rumah tangga (fahmi, 2012).

Beberapa tahun terakhir, mikrokontroler sangat banyak digunakan terutama dalam pengontrolan robot. Seiring perkembangan elektronika, mikrokontroler dibuat semakin kompak dengan bahasa pemrograman yang juga ikut berubah. Salah satunya adalah mikrokontroler AVR (*Alf and Vegard'sRisc processor*) ATmega32 yang menggunakan teknologi RISC (*Reduce Instruction Set Computing*) dimana program berjalan lebih cepat karena hanya membutuhkan satu *siklus clock* untuk mengeksekusi satu instruksi program. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu kelas ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masingmasing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama (fahmi, 2012).

Sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan *interface* I/O. Untuk mengontrol robot, maka digunakan mikrokontroler dengan pertimbangan faktor ukuran yang *relative* kecil sehingga cocok untuk pengontrol robot dan peralatan-peralatan elektronika. Sistem mikrokontroler lebih banyak melakukan pekerjaan-pekerjaan sederhana yang penting seperti mengendalikan motor, saklar, resistor variabel, atau perangkat elektronik lain (fahmi, 2012).

Fitur-fitur yang dimiliki oleh mikrokontroler AVR ATmega32 adalah sebagai berikut.

- 1. Saluran I/O ada 32 buah, yaitu Port A, Port B, Port C, dan Port D.
- 2. ADC 10 bit sebanyak 8 channel.
- 3. Tiga buah *Timer / Counter* dengan kemampuan pembanding.
- 4. CPU (Central Processing Unit) yang terdiriatas 32 buah register.

- 5. 131 instruksi handal yang umumnya hanya membutuhkan 1 siklus clock.
- 6. Watchdog Timer dengan osilator internal.
- 7. Dua buah timer / counter 8 bit.
- 8. Satu buah timer / counter 16 bit.
- 9. Tegangan operasi 2.7 V-5.5 V pada ATmega 32A
- 10. Internal SRAM (Static Random Acces Memory) sebesar 1 Kb.
- 11. Memory flash sebesar 32 Kb.
- 12. Unit interupsi internal dan eksternal.
- 13. Komunikasi Serial standar USART dengan kecepatan maksimal 2,5 Mbps.
- 14. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
- 15. Antarmuka komparator analog
- 16. Empat channel PWM (*Pulse with Modulation*)
- 17. 32x8 general purpose register.
- 18. Hampir mencapai 16 MIPS pada Kristal 16 MHz.
- 19. Port USART (Universal Synchronus and Asyncrhonous Serial Receiver And Transmitter) untuk komunikasi serial.

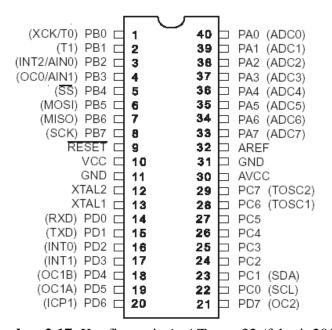
2.5.1. Konstruksi Mikrokontroler

Menurut fahmi, 2012 Mikrokontroler terdiri dari beberapa bagian diantaranya.

- 1. CPU (*Central Prosesing Unit*), pada bagian ini yaitu sebagai otak atau pusat dari pengontrolan, pengontrol utama dalam suatu mikrokontroler. CPU yang terdapat pada mikrokontroler ini ada yang berukuran 8 bit dan ada juga yang berukuran 16 bit.
- 2. ROM (*Read Only Memori*) merupakan alat untuk mengingat yang memiliki sifat bisa dibaca saja ini berarti memori ini tidak dapat ditulis, memori ini biasanya untuk menyimpan program bagi mikrokontroler tersimpan dalam format biner (0 dan 1).

- 3. RAM (*Random Access Memory*) berbeda dengan ROM sebelumnya, RAM dapat dibaca dan ditulis berulang kali.
- 4. I/O yaitu sebagai penghubung dunia luar mikrokontroler menggunakan port ini untuk download data yang bisa melalui PC (*Personal Computer*) maupun perangkat elektronika lainya.
- Transmisi Data memungkinkan pengiriman data dari computer ke *interfaces* lainnya.
 (Muthusubramanian, 2000). Mode Transmisi data dapat digolongkan menjadi dua bagian berdasarkan cara pengiriman datanya yaitu.
 - a. Transmisi *Serial*, Data dikirimkan 1 bit demi 1 bit lewat kanal komunikasi yang telah dipilih.
 - b. Transmisi *Paralel*, Data dikirim sekaligus misalnya 8 bit bersamaan melalui 8 kanal komunikasi, sehingga kecepatan penyaluran data tinggi, tetapi karakteristik kanal harus baik dan mengatasi masalah "Skew" yaitu efek yang terjasi pada sejumlah pengiriman bit secara serempak dan tiba pada tempat yang dituju dalam waktu yang tidak bersamaan.

2.5.2. Konfigurasi Mikrokontroler ATmega32



Gambar 2.17. Konfigurasi *pin* ATmega32 (fahmi, 2012)

Gambar diatas merupakan susunan standar 40 pin DIP (*Dual Inline Package*) mikrokontroler AVR ATmega32. Berikut penjelasan fungsi dari masing-masing pin ATmega32.

- 1. VCC merupakan pin masukkan positif catu daya.
- 2. GND sebagai pin Ground.
- 3. Port A (PortA0...PortA7) merupakan pin *input/output* dua arah dan pin masukan ADC.
- 4. Port B (PortB0...PortB7) merupakan pin *input/output* dua arah dan dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog dan SPI.
- 5. Port C (PortC0...PortC7) merupakan pin *input/output* dua arah dan pin fungsi khusus, yaitu TW1, Komparator *analog* dan *Timer Osilator*.
- 6. Port D (PortD0...PortD7) merupakan pin *input/output* dua arah dan pin fungsi khusus, yaitu Komparator *Analog*, interupsi *eksternal* dan komunikasi *serial*.
- 7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
- 8. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.
- 9. AVCC merupakan pin masukan tegangan untuk ADC.
- 10. AREFF merupakan pin masukan tegangan referensi ADC.

2.6. Wiznet WIZ110SR

WIZ110SR adalah *gateway* modul yang mengkonversi protokol RS-232 ke *TCP/IP* protokol. Ini memungkinkan jauh mengukur, mengelola dan mengendalikan perangkat melalui jaringan berbasis pada *Ethernet* dan *TCP/IP* dengan menghubungkan ke peralatan yang ada dengan serial RS-232 *interface*. Dengan kata lain, WIZ110SR merupakan sebuah protokol konverter yang mentransmisikan data yang dikirim dengan serial peralatan sebagai *TCP/IP* tipe data dan mengkonversi kembali *TCP/IP* data yang diterima melalui jaringan ke data *serial* untuk mengirimkan kembali ke peralatan.



Gambar 2.18. Wiznet WIZ110SR (www.wiznet.co.kr, 2014)

WIZ110SR merupakan sebuah protokol konverter yang mentransmisikan data yang dikirim dengan serial *Ethernet* dan peralatan untuk mengkonversi kembali *TCP/IP* data yang diterima melalui jaringan ke data serial untuk mengirimkan kembali ke peralatan. Ketika data diterima dari port serial, itu dikirim ke W5100 oleh MCU. Jika ada data yang dikirim dari *Ethernet*, adalah diterima penyangga *internal* W5100, dan dikirim ke port serial oleh MCU. MCU di kontrol modul data konfigurasi sesuai dengan nilai yang ditetapkan pengguna. (datasheet, 2012)

2.6.1. Fitur Utama

Fitur utama yang ada pada perangkat wiznet WIZ110SR ini adalah sebagai berikut.

- 1. Koneksi langsung ke Serial Device
- 2. Jaringan Menambahkan Fungsi λ Cukup dan Cepat
- 3. Customization λ Menyediakan Firmware
- 4. Sistem Stabilitas dan Keandalan dengan menggunakan W5100 Hardware Chip
- 5. Supports PPPoE Connection
- 6. Mendukung Serial Konfigurasi dengan perintah Sederhana dan Mudah
- 7. Mendukung Password untuk Keamanan
- 8. Configuration Tool Program
- 9. 10/100 Ethernet dan max 230Kbps Interface Serial Interface
- 10. RoHS Compliant

2.6.2. Spesifikasi

Menurut datasheet, 2014 spesifikasi yang dimiliki oleh perangkat ini adalah.

- 1. MCU
- 2. 8.051 compliant (memiliki 62K *internal Flash*, 16K SRAM, 2K EEPROM)
- 3. TCP/IP
- 4. W5100 (Ethernet MAC & PHY Tertanam)
- 5. Protokol TCP, UDP, IP, ARP, ICMP, MAC, DHCP, PPPoE, DNS
- 6. Network Interface
- 7. 10/100 Mbps (Auto detection), RJ-45 Connector
- 8. Serial Interface
- 9. RS232 (DB9)
- 10. Serial Sinyal
- 11. TXD, RXD, RTS, CTS, GND
- 12. Serial Parameter
- 13. Paritas: Tidak ada, Genap, Ganjil
- 14. Data Bits: 7,8
- 15. Flow Control: None, RTS / CTS, XON / XOFF
- 16. Speed: up to 230Kbps
- 17. Input Tegangan
- 18. DC 5V
- 19. Konsumsi arus 180mA
- 20. Suhu
- 21. 0° C ~ 80 °C (Operasi), -40° C ~ 85 °C (Storage)
- 22. Kelembaban 10 ~ 90%

2.6.3. Konfigurasi Jaringan

Konfigurasi jaringan dari perangkat ini dibagi menjadi beberapa jenis yaitu sebagai berikut.

- Versi: Menampilkan versi *firmware*. Anda dapat menggunakan PPPoE, Serial Komando,
 Sandi fungsi dari *firmware* Ver.2.5. Serial
- 2. Aktifkan *Debug Mode*: Jika mode ini dicentang, Anda dapat memonitor status dan pesan soket WIZ110SR (mendengarkan OK, sambungkan gagal dll) melalui terminal serial. Jika Debug mode aktif, pesan debug dapat menyebabkan pengoperasian normal perangkat serial. Oleh karena itu, hanya menggunakan mode ini hanya untuk Debug mode.
- 3. Board List: Jika Anda mengklik tombol "Cari", semua alamat MAC pada subnet yang sama, akan ditampilkan.
- 4. IP Konfigurasi Metode: Pilih modus pengaturan IP, Anda dapat memilih di antara Static, DHCP, PPPoE modus. Statis: "Statis" adalah pilihan untuk menetapkan WIZ110SR dengan alamat IP statis. Pertama pilih alamat MAC yang Anda ingin mengaturnya sebagai IP statis dalam "daftar papan". Lalu, yang "Lokal IP, Subnet, Gateway kotak diaktifkan. Input Static IP address dan klik "setting" tombol. Kemudian IP adalah ditetapkan sebagai yang Anda inginkan. (PPPoE ID, kotak Password akan dinonaktifkan) DHCP: Tetapkan pilihan ini untuk menggunakan DHCP mode. Pertama, cek "DHCP" dan klik "Setting" tombol. Jika alamat IP berhasil diperoleh dari server DHCP, alamat MAC akan ditampilkan pada daftar papan. (Butuh beberapa waktu untuk mendapatkan alamat IP dari server DHCP) Ketika sebuah modul di dalam daftar papan dipilih, alamat IP, Subnet mask dan Gateway akan ditampilkan. Jika modul jaringan tidak bisa memperoleh informasi dari server DHCP, alamat IP, Gateway Address dan Subnet mask akan diinisialisasi ke 0.0.0.0. PPPoE: WIZ110SR mendukung PPPoE untuk pengguna ADSL. Ketika Anda memilih PPPoE di "IP Configuration Metode" PPPoE kotak ID dan Password akan diaktifkan. Untuk mengatur PPPoE, hubungkan PC ke WIZ110SR

langsung dan melaksanakan program *Configuration Tool* pada PC. (Configuration Tool harus versi 2.1 atau di atas), Pilih "PPPoE" dari "IP Configuration Metode" tab dan masukan ID & Password. Kemudian klik "setting" tombol untuk menerapkannya. Hubungkan Module to ADSL Line. Jika "Aktifkan *Mode Debug Serial*" dipilih, Anda dapat melihat status akses PPPoE melalui konsol serial.

- 5. Lokal IP / Port: WIZ110SR "s alamat IP dan nomor Port untuk koneksi jaringan, Pastikan untuk memasukkan informasi jaringan yang benar. Jika Anda salah memasukkan alamat IP, itu dapat menyebabkan tabrakan jaringan atau mal-fungsi.
- 6. Subnet: WIZ110SR "s subnet mask
- 7. Gateway: WIZ110SR" s alamat Gateway.
- 8. PPPoE ID / *Password*: Jika Anda memilih "PPPoE" mode, masukan ID / *Password* yang Anda terima dari perusahaan ISP.
- 9. Server IP / Port: Ketika WIZ110SR tersebut ditetapkan sebagai "Client mode" atau "Mixed mode", server IP dan port harus ditetapkan. WIZ110SR mencoba untuk menghubungkan alamat IP ini.
- 10. Modus Jaringan. *Klien / server /* campuran: Ini adalah untuk memilih metode komunikasi yang didasarkan pada TCP. TCP merupakan protokol untuk menetapkan sambungan sebelum data komunikasi, tetapi UDP hanya proses komunikasi data tanpa pembentukan sambungan. Mode Jaringan WIZ110SR dapat dibagi menjadi TCP *Server*, TCP *Client* dan modus Campuran sambungan sesuai dengan metode membangun. Pada modus *server* TCP, WIZ110SR beroperasi sebagai *server* pada proses koneksi, dan menunggu sidang sambungan dari klien. WIZ110SR beroperasi sebagai klien pada *Client* TCP modus pada proses koneksi, dan mencoba untuk menyambung ke *server* IP dan Port. Mixed mode mendukung kedua dari *Server* dan *Client*. TCP *Server* modus ini dapat bermanfaat bila pusat pemantauan mencoba terhubung ke perangkat (di mana WIZ110SR diinstal) untuk

memeriksa status atau memberikan perintah. Dalam waktu normal WIZ110SR adalah pada status menunggu, dan jika ada permintaan sambungan (SYN) dari pusat pemantauan, sambungan dibuat (DIRIKAN), dan komunikasi data diproses (Data Transaksi). Akhirnya sambungan ditutup (FIN). Agar dapat bekerja dengan modus ini, Lokal IP, *Subnet*, *Gateway* Alamat dan Lokal Port Number harus dikonfigurasi terlebih dahulu.

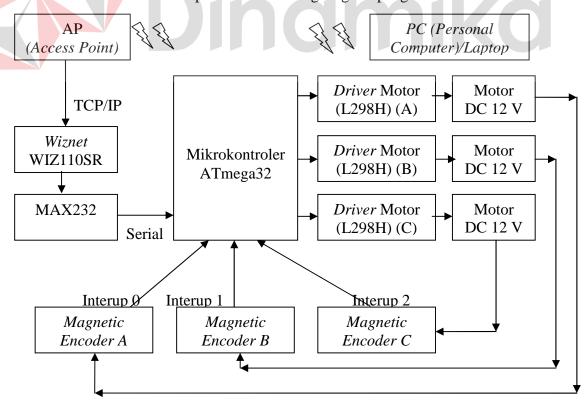


BAB III

METODE PENELITIAN

Metode penelitian yang digunakan meliputi studi kepustakaan dan penelitian laboratorium. Studi kepustakaan dilakukan untuk mencari teori atau informasi dari buku, jurnal, datasheet, dan artikel-artikel yang berkaitan dengan permasalahan. Antara lain, omni directional-wheels, gerak odometry, Metode PID, Magnetic Encoder, mikrokontroler ATmega32, modul komunikasi wireless WIZ110SR, komunikasi serial asynchronous (UART).

Dari informasi dari studi kepustakaan yang diperoleh maka dilakukan penelitian laboratorium, yaitu perancangan perangkat keras dan perangkat lunak seperti pada Gambar 3.1. Setelah sistem keseluruhan selesai sistem harus diuji secara keseluruhan dan diimplementasikan langsung dilapangan.



Gambar 3.1. Blok Diagram Sistem Keseluruhan

Dari gambar 3.1 dapat dilihat keseluruhan dari sistem ini, dimana input dari sistem ini ada dua yaitu *magnetic encoder* dan perintah dari PC/laptop. Awalnya *user* memasukkan perintah koordinat tujuan ke program *visual basic 6.0* pada komputer, kemudian program tersebut akan merubah koordinat X dan Y dari tujuan tersebut menjadi arah dari perputaran roda dan juga jarak tempuh untuk setiap roda *omni*. Selanjutnya data tersebut dikirimkan oleh komputer secara nirkabel ke *acces point* (AP) yang ada pada *mobile robot* dan diteruskan ke *wiznet* (WIZ110SR) agar dapat berkomunikasi dengan mikrokontroler menggunakan komunikasi USART (*universal synchronous asynchronous receive/transmiter*). Setelah mikrokontroler menerima informasi dari wiznet maka mikrokontroler akan mengerakkan setiap motor sesuai dengan jarak berserta arah yang telah ditentukan oleh informasi awalnya. Nantinya setiap perputaran dari roda ini akan di hitung oleh mikrokontroler berdasarkan *magnetic encoder* yang ada.

3.1. Alat dan Bahan Penelitian

Alat dan bahan yang akan digunakan dalam penelitian ini terdiri dari beberapa komponen, alat-alat ini akan digunakan untuk mengukur dan melihat hasil yang akan terjadi terhadap sistem yang akan dibuat oleh peneliti. Adapun alat dan bahan yang akan digunakan sebagai berikut.

3.1.1. Alat Penelitian

Alat yang digunakan untuk menyelesaikan penelitian ini adalah sebagai berikut.

 Tachometer (RPM Meter), digunakan untuk mengukur kecepatan putaran dari motor DC pada mobile robot.

- 2. Personal Computer (PC) / Laptop, digunakan untuk menguji pengiriman serial antar computer dan mikrokontroler serta untuk mengelolah data koordinat.
- 3. *Driver* Motor, digunakan untuk mengendalikan kecepatan dan arah putaran dari motor DC.
- 4. *Minimum* Sistem, digunakan untuk menproses semua data pergerakkan dan komunikasi *mobile robot*.
- Access Point (AP), digunakan untuk menghubungkan antara PC/Laptop berbasis TCP/IP.
- 6. Pengaris, digunakan untuk mengukur jarak tempuh *mobile robot*.
- 7. Busur, digunakan untuk mengukur putaran robot.
- 8. Module WIZ110SR, digunakan untuk merubah protocol TCP/IP menjadi serial.
- 9. Rangkaian MAX232, digunakan untuk merubah level tegangan pada module wiznet dan mikrokontroler sehingga dapat saling berkomunikasi.
- 10. *Batteray* Lippo 12V, digunakan sebagai sumber daya untuk semua komponen elektronik pada *mobile robot*.

3.1.2. Bahan Penelitian

Bahan yang akan diteliti oleh penulis dalam penelitian kali ini meliputi beberapa aspek sebagai berikut.

- 1. Gerak *mobile robot* yang menggunakan *triangle omni-directional* secara keseluruhan.
- 2. Pulsa dari *magnetic encoder* yang akan menjadi sinyal pengukur kecepatan pada *mobile robot*.

- 3. Sinyal yang dihasilkan oleh *magnetic encoder* pada setiap Motor DC yang ada.
- 4. Komunikasi serial yang terjadi antara *mobile robot* dan PC (*personal Computer*) untuk memberikan interuksi atau perintah.

3.2. Jalan Penelitian

Tahap-tahap yang akan peneliti lakukan untuk menyelesaikan penelitian ini adalah sebagai berikut.

3.2.1. Pengumpulan Literatur

Pengumpulan literatur, data, dan informasi untuk menyelesaikan permasalahan yang ada pada penelitian ini dilakukan dengan cara mencari buku, jurnal, dan sumber-sumber lainya yang dapat membantu untuk menyelesaikan penelitian ini.

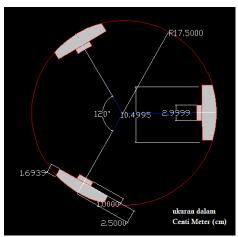
3.2.2. Peranc<mark>an</mark>gan Perangkat Keras

Dalam perancangan perangkat keras ini dibagi menjadi beberapa bagian pengerjaan yang juga menjadikan beberapa modul utama yaitu sebagai berikut.

A. Perancangan Badan Robot

Untuk perancangan dari badan robot ini peneliti menggunakan program AUTOCAD 2004 untuk mendapatkan desain yang sesuai dengan apa yang penulis inginkan. Komponen atau bahan yang peneliti gunakan untuk merancang badan robot ini terdiri dari *acrylic* bening berbentuk bundar yang digunakan untuk meletakkan semua modul elektronika dan motor DC 12 V sebagai badan robot. Dalam perancangan ini karena menggunakan tiga buah motor DC yang pasangkan dengan ban *omni-directional* maka posisi dari ketiga ban atau motor DC yang digunakan harus sebesar 120° sesuai dengan peryataan oleh rafiudin.

Ukuran yang akan dibuat dalam penelitian ini dapat dilihat pada gambar 3.2 ini. Penulis medesain ukuran diameter dari *mobile robot* ini sebesar 35cm dikarenakan panjang dari setiap motor DC adalah sebesar 12 cm.



Gambar 3.2. Desain Badan Robot

Struktur Metal yang akan digunakan untuk membuat robot ini nantinya adalah sebagai berikut.

- 1. Mur dan Baut.
- 2. Almunium.
- 3. Acrylic 5mm.
- 4. Besi (Socket Ban Omni).

Untuk bentuk secara nyatanya dapat dilihat sebagai berikut.



Gambar 3.3. Bentuk Badan Robot

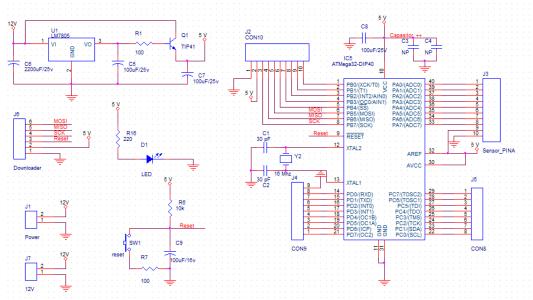
Dalam pembuatan badan robot ini menggunakan *machine grafis desain* yaitu *autocad* dan proses pemotongan *acrylic* yang akan digunakan sebagai badan dari *mobile robot* ini menggunakan teknologi *laser cutting* yang memiliki tingkat kepresisian tinggi sehingga kesalahan yang ada akan menjadi sangat minim.

B. Perancangan Rangkaian Minimum Sistem

Rangkaian minimum sistem dibuat untuk mendukung kerja dari mikrokontroler ATmega32 dimana mikrokontroler ini tidak bisa berdiri sendiri melainkan harus ada rangkaian dan komponen pendukung seperti halnya rangakaian catu daya, kristal dan lain sebagaianya yang biasanya disebut *minimum* sistem.

Mikrokontroler berfungsi sebagai otak dalam mengolah semua instruksi baik *input* maupun *output* seperti halnya pemroses data *input* dari *magnetic encoder* yang kemudian menghasilkan *output* yang berupa putaran dan arah dari motor DC.

Minimum sistem yang dirancang oleh peneliti kali ini menggunakan minimum sistem mikrokontroler ATmega32. Mikrokontroler ATmega32 digunakan karena pada perancangan penelitian ini tidak memerlukan banyak pin I/O.



Gambar 3.4. Skematik rangkaian dari minimum sistem Atmega32

Skematik *minimum* sistem ATmega32 pada Gambar 3.4 terdiri dari mikrokontroler ATmega32, rangakaian oscillator, rangkaian reset, rangkaian catu daya, rangkaian konektor untuk *downloader* program, serta pin-pin konektor untuk komunikasi USART, dan PWM untuk *driver* motor dan lain-lain. Tabel 3.1 merupakan daftar pin I/O yang digunakan dalam penelitian ini.

Tabel 3.1. Pin I/O *Minimum* Sistem

| PIN input/Output | Fungsi | | |
|--------------------------|---|--|--|
| VCC | <i>Power</i> 4.5 – 5.5 VDC | | |
| GND | Ground | | |
| PINA.0 – PINA.5 | Input direction untuk driver motor. Satu | | |
| | driver motor butuh dua buah input. | | |
| PD4, PD5 dan PD7 (OCR1A, | Sebagai input masukkan PWM (pulse width | | |
| OCR1B, OC2) | Modulation) dari driver motor DC. | | |
| PD2, PD3 dan PB2 (INT0, | Pin interup sebagai input masukkan dari | | |
| INT1, dan INT2) | magnetic encoder. | | |
| Pin D.0 (RxD) | Sebagai pin receiver (penerima) data dari | | |
| | WIZ110SR | | |
| Pin D.1 (TxD) | Sebagai pin Transitter (pengirim) data dari | | |
| | WIZ110SR | | |

Setelah pembuatan skematik dan pembuatan alokasi list untuk semua portnya sudah jadi, selanjutnya diteruskan dengan melakukan desain *layout* dari

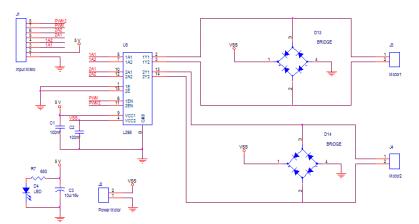
rangkaian *minimum* sistem ini. Applikasi untuk melakukan desain *layout* ini adalah *orcad family release 9.0*.

C. Perancangan Rangkaian Driver Motor (L298H)

Rangkain *driver* motor L298H ini dibuat untuk mengendalikan kerja motor DC 12V dimana mikrokontroler nantinya akan memberikan inputan pada pin-pin dari L298H ini untuk mempercepat, memperlambat, dan memutar arah putaran dari motor DC.

Rangkaian *driver* motor L298H ini dibuat untuk mendukung kerja dari L298H, karna L298H ini tidak dapat berdiri sendiri melainkan harus ada rangkaian dan komponen pendukung lainya seperti catu daya, kapasitor, dan *conector* yang biasanya disebut sebagai *driver* motor.

Pada perancangan driver motor L298H ini juga menggunakan aplikasi yang sama dengan minimum sistem yaitu mengunakan aplikasi orcad. Setelah melakukan perancangan dari rangkain driver motor kemudian melakukan layout dari rangkaian tersebut. Adapun hasil dari rangkaian driver motor L298H yang akan digunakan dalam penelitian ini seperti pada gambar 3.5 berikut ini.



Gambar 3.5. Rangkaian Driver Motor L298H

Skematik *driver* motor L298H pada gambar 3.5 terdiri dari L298H, *diode* bridge, conector motor, rangkaian led, serta rangkaian conector input untuk direction dan PWM dari mikrokontroler. Pada tabel 3.2 merupakan daftar pin input/ouput yang digunakan dalam penelitian ini.

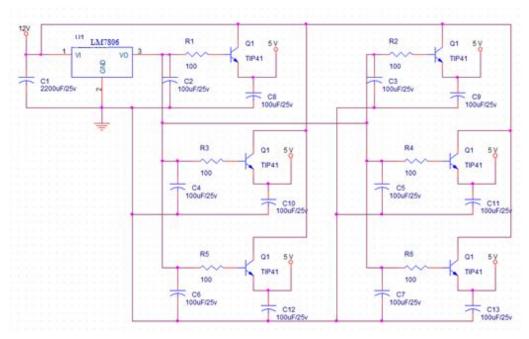
Tabel 3.2. Pin Input/Output Driver Motor L298H

| Pin I/O | Fungsi |
|----------|---|
| 1 (GND) | Ground |
| 2 (VCC) | VCC 5 VDC |
| 3 (1A1) | Direction satu motor satu |
| 4 (1A2) | Direction dua motor satu |
| 5 (2A1) | Direction satu motor dua |
| 6 (2A2) | Direction dua motor dua |
| 7 (PWM) | PWM (pulse width Modulation) motor satu |
| 8 (PWM2) | PWM2 (pulse width Modulation) motor dua |

Setelah perancangan dan *allocation list* dari rangkaian *driver* motor L298H ini selesai selanjutnya melakukan *layout* dari rangkaian ini. Pada perancangan ini *layout* dibagi menjad tiga buah bagian yaitu *layout* bagian atas, *layout* bagian komponen dan *layout* bagian bawah.

D. Perancangan Rangkaian Regulator

Rangkaian *Regulator* ini dibuat untuk merubah level tegangan *input* dari batteray sebesar 12 Volt menjadi 5 Volt sesuai dengan *range* kerja dari mikrokontroler yang berada pada level tegangan 4,5 Volt hingga 5,5 Volt. Maka dari karakteristik ini dibuat sebuah skematik rangkaian *regulator* yang dapat dilihat pada gambar 3.6 berikut ini.



Gambar 3.6. Skematik Rangkaian Regulator

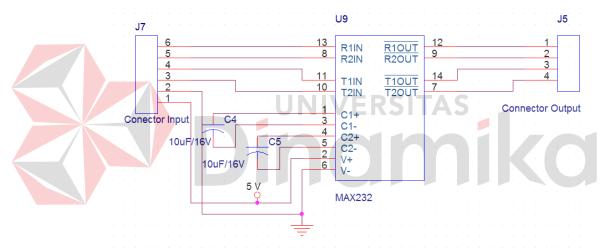
Pada rangkaian ini memiliki *input* tegangan sebesar 12 Volt dari *batteray* yang akan masuk ke rangkaian *regulator*, dikarenakan komponen *regulator* LM7805 hanya memiliki keluaran arus sebesar satu *ampere* maka akan diperkuat dengan menggunakan rangkaian penguat arus, rangkaian penguat arus ini mengakibatkan turunnya tegangan keluaran dari komponen *regulator* 7805 sehingga peneliti menggunakan komponen 7806 yang memiliki keluaran tegangan sebesar 6 Volt sebagai penganti komponen 7805 yang memiliki keluaran tegangan sebesar 5 Volt.

Jumlah komponen penguat arus yang digunakan dalam penelitian ini sebanyak enam buah komponen TIP41 dikarenakan jumlah arus yang dibutuhkan untuk menjalankan *mobile robot* ini nantinya lebih dari empat *ampere*. Besar arus yang dibutuhkan ini berasal dari rangkaian *minimum* sistem, rangkaian *Driver* Motor, *Access point*, Wiznet WIZ110SR dan *magnetic encoder*.

Hasil keluaran yang diharapkan dari rangkaian regulator ini yaitu sebesar 5 Volt sehingga berada dalam sistem kerja dari rangkaian mikrokontroller dan komponen pendukung lainnya yang akan digunakan pada *mobile robot* ini.

E. Perancangan Rangkaian MAX232

Rangkaian Max232 ini dibuat untuk merubah *level* tegangan antar *input* dan *output* pada komunikasi *serial* sehingga nantinya komunikasi *serial* (USART) antar kedua belah pihak dapat saling berkomunikasi dengan *level* tegangan yang berbeda. Rangkaian ini dibuat untuk mendukung kerja komunikasi USART antara WIZ110SR dengan mikrokontroler.



Gambar 3.7. Rangkaian Max232

Skematik rangkaian max232 pada gambar 3.7 terdiri dari *conector input* yang digunakan untuk masuk ke WIZ110SR, dan *conector output* yang digunakan untuk masuk ke mikrokontroler serta kapasitor untuk penyimpan tegangan. Pada rangkaian ini R1IN akan terhubung pada *transmitter* dari modul Wiznet dan kaki R1out akan masuk ke kaki *receiver* Mikrokontroller. Sedangkan kaki R2IN akan masuk ke kaki transmitter dari mikrokontroller dan kaki R2Out akan masuk ke kaki receiver WIZ110SR. Pada tabel 3.3 berikut merupakan daftar pin *input/ouput* yang digunakan dalam penelitian ini.

Tabel 3.3. Pin Input/Output Max232.

| Pin I/O | Fungsi |
|---------------------------|----------------------|
| Conector input (6 dan 5) | Receive dari input |
| Conector input (3 dan 4) | Transmite dari input |
| Conector input (1 dan 2) | VCC dan Ground |
| Conector output (1 dan 2) | Receive dari output |
| Conector output (3 dan 4) | Transmite dari ouput |

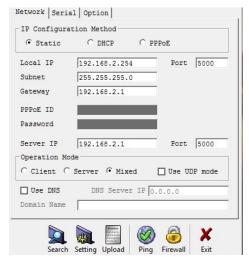
3.2.3. Perancangan Perangkat Lunak

Pada perancangan perangkat lunak ini dibagi atas beberapa bagian yang memiliki fungsi masing-masing untuk menjalankan *mobile robot* ini. Adapun bagian-bagian dari perancangan perangkat lunak ini dibagi sebagai berikut.

A. Konfigurasi interfaces WIZ110SR

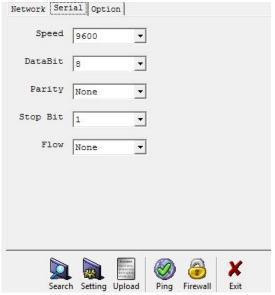
Pada perangkat wiznet WIZ110 SR aplikasi yang digunakan untuk melakukan setup terhadap interfaces ini adalah aplikasi WIZ110SR configtool, konfigurasi dari perangkat ini harus sesuai dengan pengaturan ip addressing yang mana pengaturannya dapat disesuaikan dengan pengunaannya. Dalam perancangan penelitian ini Konfigurasi jaringan dari interfaces perangkat ini diatur sebagai berikut.

Ip konfigurasi method diatur menjadi static konfigurasi ip address dikarenakan dengan penggunaan ip address yang static maka alamat yang dituju untuk melakukan komunikasi tidak akan berganti sehingga dapat memudahkan untuk melakukan hubungan terhadap perangkat. Konfigurasi interfaces ini terdiri atas local ip address 192.168.2.254 dengan subnetmask 255.255.255.0 dan gateway 192.168.2.1.



Gambar 3.8. Pengaturan jaringan dari WIZ110SR

Konfigurasi *serial* dari antarmuka perangkat ini diatur menjadi kecepatan pengiriman data 9600 bps (*BaudRate*) untuk pengaturan kecepatan pengiriman datanya bisa lebih tinggi lagi sampai dengan 115200 bps (*BaudRate*) namun dalam penelitian kali ini peneliti merasa dengan kecepatan 9600bps sudah cukup untuk mengirimkan informasi yang diinginkan, dengan jumlah pengiriman data perbitnya yaitu 8 *bit*, *stop bit* hanya 1 bit, *parity* dan *flow* diatur menjadi kosong. Konfigurasi seperti ini adalah konfigurasi standar pada umumnya dari semua perangkat komunikasi serial yang ada.



Gambar 3.9. Pengaturan pengiriman serial dari WIZ110SR

Untuk dapat melakukan komunikasi secara serial antara antarmuka ini dengan antarmuka lainya maka pengaturan untuk speed, databit, parity, stop bit, dan *flow* harus sama. Karena apabila pengaturan dari pengiriman ini berbeda maka tidak akan dapat terjalin suatu komunikasi. Sehingga peraturannya diatur agar menjadi seperti gambar 3.9.

Untuk membuat *mobile robot* ini menjadi sangat mudah digunakan maka ditambahkan sebuah *access point* yang akan digunakan sebagai pemancar sinyal berbasis *wireleas*, sehingga semua *user* yang ingin mengendalikan robot ini tidak harus menggunakan kabel untuk dapat berkomunikasi. Interfaces TCP/IP dari wiznet WIZ110SR dirhubung secara langsung dengan *acces point*.

B. Konfigurasi Winsock Pada Visual Basic 6.0

Untuk dapat melakukan komunikasi berbasis TCP/IP *protocol* maka penulis dalam memilih menggunakan *Visual Basic* 6.0 untuk menyelesaikan *Penelitian* ini dikarenan pada *visual basic* 6.0 ada aplikasi winsock yang digunakan sebagai pengaturan komunikasi antar TCP/IP *protocol*.

Pada program *Visual Basic 6.0* ini digunakan *Microsoft winsock* sebagai komunikasi TCP/IP untuk dapat saling berkomunikasi dengan mikrokontroler. Program *winsock* ini nantinya akan mengirimkan data kepada wiznet melewati *access point* dan wiznet akan meneruskan data dari TCP/IP tersebut dengan komunikasi serial ke mikrokontroler. Adapun pengaturan dari winsock ini adalah sebagai berikut.

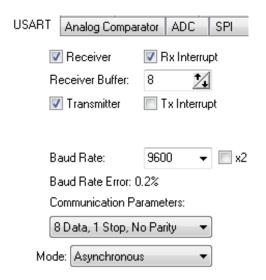
| Alphabetic Categorized | | | | |
|--------------------------|-------------------|--|--|--|
| (About) | | | | |
| (Custom) | | | | |
| (Name) | Winsock1 | | | |
| Index | | | | |
| Left | 1320 | | | |
| LocalPort | 0 | | | |
| Protocol | 0 - sckTCPProtoco | | | |
| RemoteHost | | | | |

Gambar 3.10. Pengaturan winsock pada Visual Basic 6.0.

Winsock melakukan komunikasi berbasis TCP/IP, sehingga membutuhkan ip address antar perangkat yang digunakan sebagai alamat tujuan untuk saling berkomunikasi, port address yang akan dibuka untuk melakukan komunikasi antara tujuan dan pengirim. Dalam penelitian ini port address yang digunakan harus diatas dari port 1024 karena port 0-1024 merupakan Well-Know-Port, sehingga penulis menggunakan adalah port 5000 yang merupakan Unknow-Port.

C. Konfigurasi Serial pada Mikrokontroler

Konfigurasi serial atau USART (Universal synchronous asynchronous Receiver Transmitter) pada CVAVR ini harus sesuai dengan pengaturan yang ada pada wiznet, dikarenakan ini adalah pengiriman yang tidak syncronus sehingga apabila terjadi perbedaan kecepatan penerimaan data atau pengiriman data maka data yang diterima tidak akan benar (fail). Konfigurasi serial dari interfaces perangkat ini diatur menjadi kecepatan pengiriman data 9600 bps (BaudRate), dengan jumlah pengiriman data perbitnya yaitu 8 bit, stop bitnya 1 bit, parity dan flow diatur menjadi kosong. Seperti pada gambar berikut.



Gambar 3.11. Konfigurasi USART pada CVAVR.

Penerimaan data *serial* pada mikrokontroller ini akan diinterup, agar pada saat ada suatu perangkat yang mengirimkan data ke mikrokontroller yang melewati komunikasi *serial* maka mikrokontroller akan melakukan penundaan sementara untuk semua proses yang sedang dia lakukan dan melakukan penerimaan data. Setelah data yang dikirimkan telah selesai sepenuhnya barulah kemudian mikrokontroller melanjutkan interuksi yang ada.

D. Perancangan Program PID Pada Mikrokontroler

Pengendali PID adalah suatu sistem pengendali yang merupakan gabungan antara pengendali *proporsional, integral*, dan turunan (*derivative*). Umpan balik pada *triangle omni mobile robot* ini menggunakan parameter pulsa yang dihasilkan oleh *magnetic encoder* sebagai masukkan yang akan dirubah menjadi kecepatan dari motor DC.

Pada modul driver motor L298H memiliki jenis *hybrid driver motor* dimana modul ini memiliki kemampuan untuk melakukan pengereman terhadap motor yang sedang berputar. Modul ini akan melakukan pengereman terhadap motor apabila kaki *input* dari *direction* pada *driver motor* yaitu 1A1 dan 1A2

untuk motor satu serta 2A1 dan 2A2 untuk motor dua diberikan tegangan 5 Volt. Ketika kaki dari direction pada motor satu dan motor dua diberikan tegangan 0 Volt maka rangkaian ini akan memberhentikan motor secara perlahan sehingga akan terjadi lembam terhadap motor yang sedang berputar sebelumnya.

Berdasarkan persamaan pada rumus 2.1 maka hasil penulisan program untuk pengendalian PID ini yaitu sebagai berikut.

```
void PID_M3 (int Sp)
{
    error = Sp - counter3;
    MV = Motor3+(Kp*error)+((Ki/50)*(error+last_error3))+
((Kd/50)*(error-last_error3));

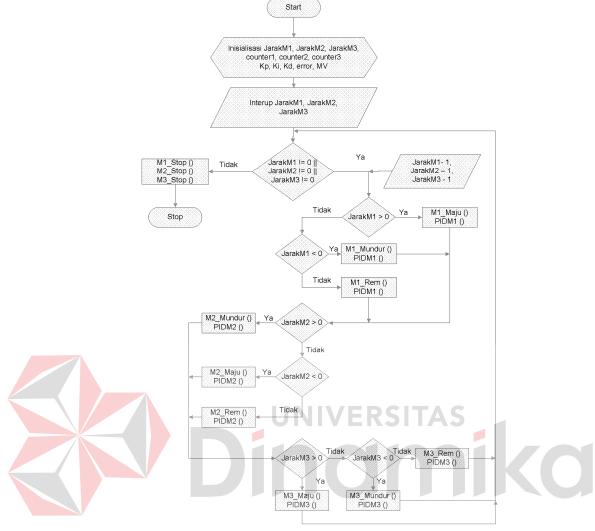
    if (MV >= 255)
    {
        Motor3 = 254;
    }
    else if (MV <= 0)
    {
        M3_Rem ();
        Motor3 = 0;
    }
    else
    {
        Motor3 = error;
        counter3 = 0;
}</pre>
```

Pada potongan program diatas dibuat sebuah *function* untuk mengendalikan motor tiga dengan nama PID_M3. *Variable* Sp merupakan nilai dari *set point* yang diinginkan, *variabel* counter3 merupakan *variabel* yang digunakan untuk menghitung jumlah umpan balik dari *magnetic encoder* pada motor tiga, *variable error* merupakan hasil dari nilai *Set point* yang diinginkan dikurangi dengan nilai counter3 pada saat itu. *Variabel last_error* merupakan *variabel* yang digunakan untuk menampung kesalahan sebelumnya dari umpan balik *magnetic encoder*, dimana nilai dari *last error* ini didapatkan dari nilai

error. Control Proporsional merupakan hasil dari Kp yang dikalikan dengan variabel error, control Integral merupakan hasil dari Ki dibagi dengan time sampling (50 ms) yang kemudian dikalikan dengan nilai error ditambahkan dengan last_error, control D merupakan hasil dari Kd dibagi dengan nilai time sampling dan dikalikan dengan nilai error dikurangi dengan error sebelumnya. MV (Move value) adalah variable yang digunakan untuk menampung semua nilai dari control Proporsional, Integral, dan Derivative yang kemudian ditambahkan dengan kecepatan awal atau kecepatan sebelumnya dari motor D. Karena dalam pengerjaan penelitian kali ini menggunakan Timer 8 bit maka batas atas atau batas maximum dari PWM (pulse witdh modulation) yang dapat dihasilkan oleh mikrokontroller adalah sebesar 255.

E. Perancangan Program pengerak Mobile robot

Algoritma ini akan digunakan untuk mengendalikan gerak robot serta kendali posisi dari robot ini, dimana nantinya algoritma ini akan menghitung putaran dari setiap motor dan serta mengendalikan arah dari putarannya. Input dari mikrokontroler ini yaitu magnetic *encoder* dari motor DC dan jarak untuk motor satu, dua, dan tiga. Program mikrokontroler ini dibuat dengan algoritma seperti pada gambar 3.12.



Gambar 3.12. Flowchart Program Mikrokontroler

Dari gambar 3.12 langkah pertama yang dilakukan adalah inisialisasi untuk nilai awal dari jarak motor 1 (JarakM1), jarak motor 2 (JarakM2), jarak motor 3 (JarakM3), serta *variable counter* dan PID yang akan digunakan sebagai pengendali kecepatan motor.

Mikrokontroler akan menunggu terjadi interup pengiriman data *serial* dari wiznet yang akan merubah *variable* dari jarak pada setiap motor. Adapun penerimaan data dari setiap motor ini dipasangan dengan *header* sebagai berikut.

Tabel 3.4. Format Pengiriman Data *Serial*.

| Α | Theta | В | Jarak1 (| (+arah-) | C | Jarak2 (+arah-) | D | Jarak3 (+arah- | (| Е |
|---|-------|---|----------|----------|---|-----------------|---|----------------|---|---|

Untuk program *interup* dari penerimaan data *serial* ini adalah sebagai berikut.

```
if (data == 'E')
      header = 5;
      element = 0;
else if (header == 4)
{
      USART_Terima3 [element++] = data;
if (data == 'D')
      header = 4;
      element = 0;
else if (header == 3)
      USART_Terima2 [element++] = data;
if (data == 'C')
{
      header = 3;
      element = 0;
else if (header == 2)
      USART_Terima [element++] = data;
if (data == 'B')
{
      header = 2;
      element = 0;
else if (header == 1)
      Theta = data;
if (data == 'A')
      header = 1;
}
```

Setelah penerimaan data dari *serial* telah selesai selanjutnya data tersebut akan masuk kedalam perulangan yang akan dirubah menjadi kendali dari kecepatan dan posisi robot. Adapun potongan program untuk kendali dan posisi dari kecepatan robot ini adalah sebagai berikut.

```
while (Jarak1 != 0 || Jarak2 != 0 || Jarak3 != 0)
{
    if (Jarak1 == 0)
    {
```

Ketika nilai jarak dari setiap motor telah sama dengan nol maka motor DC dari robot ini akan direm.

```
if (Jarak1 == 0)
{
      M1_Rem ();
}
.....
if (Jarak2 == 0)
{
      M2_Rem ();
}
.....
if (Jarak3 == 0)
{
      M3_Rem ();
}
```

UNIVERSITAS

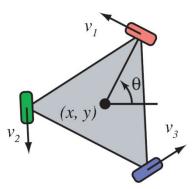
Pada saat semua nilai dari jarak satu hingga jarak ketiga telah nol maka robot akan berhenti untuk menunggu interuksi selanjutnya. Apabila pada saat program sedang berjalan dan terjadi pengiriman *interup* dari pengiriman serial maka setelah *interup* tersebut selesai data yang dikirimkan akan terabaikan oleh program ini. Adapun potongan program dari pengendalian ini adalah sebagai berikut.

```
M1_Stop ();
M2_Stop ();
M3_Stop ();
```

F. Perancangan Program Pengerak Omni-Directional

Pada penelitian ini penulis membuat desain dari *mobile robot* yang menggunakan *triangle omni-directional wheels*. Dimana posisi setiap ban harus

memiliki sudut 120° dari ban yang lainnya atau berbanding 30° dengan sudut siku-suku 90° untuk setiap bannya.



Gambar 3.13. Desain triangle omni-directional wheels.

1. Koordinat Wilayah Mobile robot

Untuk mengendalikan gerak dari *mobile robot* ini peneliti membagi ruang gerak dari robot ini menjadi empat daerah atau empat kuadran dimana setiap kuadrannya dibagi menjadi 90° setiap kuadrannya. Kuadran pertama berada pada sudut 0° hingga 90°, kuadran kedua berada pada sudut 90° hingga 180°, kuadran ketiga berada pada sudut 180° hingga 270°, dan kuadran keempat berada pada sudut 270° hingga 360°/0°.

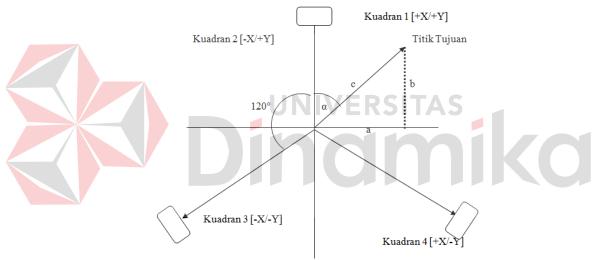
Untuk membagi gerakkan robot menjadi empat kuadran tersebut maka penulis mendeteksi inputan posisi yang diberikan oleh user, apabila koordinat input dari user (X,Y) bernilai sama-sama positif maka robot akan bergerak pada kuadran pertama. Apabila koordinat tujuan dari input user memiliki nilai minus pada salah satunya maka robot akan bergerak ke kuadran dua atau kuadran empat. Adapun perincian dari kendali gerak dan posisi pada *mobile robot* ini dapat dilihat pada tabel 3.5 berikut.

Tabel 3.5. Koordinat Wilayah *Mobile robot*.

| Input koordinat | Kuadran |
|-----------------|---------|
| X, Y | 1 |
| X, -Y | 2 |
| -X, Y | 3 |
| -X, -Y | 4 |

2. Gerak Mobile robot

Mobile robot ini akan bergerak berdasarkan intruksi koordinat yang dikirimkan, dalam pergerakkannya robot ini dibutuhkan jarak dan sudut tempuh yang akan dibentuk oleh robot ini. Adapun bagaimana cara menghitung dari jarak dan sudut tempuh oleh robot ini seperti pada gambar 3.14 berikut.



Gambar 3.14. Jarak dan sudut tempuh dari robot.

Untuk menghitung jarak tempuh dari robot digunakan rumus *phytagoras* dimana jarak tempuh adalah sisi miring dari segitiga dan sisi siku-siku dari segitiga adalah jarak posisi X dan jarak dari posisi Y, dan untuk menghitung sudut yang dibentuk dari phytagoras maka menghasilkan.

Sesuai dengan persamaan 2.15 dan 2.16 maka didapatkan jarak tempuh dan sudut tempuh dari *mobile robot triangle omni-directional* ini, maka robot akan bergerak berdasarkan posisi kuadran yang ada dengan membentuk jarak dan sudut yang sesuai dengan nilainya masing-masing. Apabila tujuan dari robot

tersebut berada pada kuadran satu dan tiga, maka motor kedua yang akan menjadi sudut alpha dari robot, motor satu dan motor dua akan bergerak yang sesuai dengan arah kuadran tersebut (maju/mundur), dan apabila tujuan dari robot tersebut berada pada kuadran satu dan empat, maka motor ke tiga yang akan menjadi sudut alpha dari robot, motor satu dan motor dua akan bergerak dengan arah yang sesuai dengan kuadran yang dituju (maju/mundur).

Adapun potongan program dari penghitungan sudut tempuh dan jarak tempuh dari *mobile robot* ini adalah sebagai berikut.

```
If (CInt(Text1.Text) >= 0) And (CInt(Text2.Text) >= 0) Then

Sudut = CInt(Atn(CInt(Text1.Text) / CInt(Text2.Text)) * (180 / 3.14))

Sudut = Sudut - 60

Putar = Sudut * 3.4611

Jarak = (Sqr((CInt(Text1.Text) ^ 2) + (CInt(Text2.Text) ^ 2))) * 11.3376

Winsock1.SendData ("A" + "3" + "B" + CStr(Jarak) + "C" + CStr(Jarak) + "D" + CStr(Putar * 4) + "E")
```

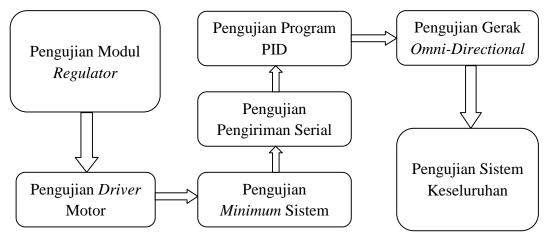
Pada potongan program diatas robot akan bergerak ke kuadran pertama dikarenakan nilai dari text1.text dan nilai dari text2.text lebih besar dari nol dimana nilai dari text1.text adalah sumbu Y dan nilai dari text2.text adalah sumbu X. Hal pertama yang akan dihitung adalah mencari besar sudut Θ tehadap titik 0° Setelah nilai sudut sesuai persamaan dari 2.15 telah didapatkan maka akan dikurangi dengan 60° dikarenaka posisi motor3 berada pada sudut 240° sehingga $(240-180) = 60^{\circ}$. Variable putar yaitu sudut Θ yang akan dikalikan dengan jumlah pulsa per satu derajat putaran badan $mobile\ robot$. Karena keliling lingkaran dari

badan *mobile robot* = 2*3.14*17.5 maka didapatkan nilai sebesar 109.9 cm per 360° putaran penuh dari *mobile r*obot. Dari persamaan ini dapat kita hitung bahwa 1° putaran *mobile robot* sama dengan 109.9 cm dikali 11.3376 *pulsa* per cm dan dibagi sebesar 360° yang menghasilkan nilai sebesar 3.4611 *pulsa* per satu derajat putaran badan *mobile robot*.

Perhitungan yang digunakan untuk menghitung jarak tempuh dari roda pada *mobile robot* ini didapatkan dari keliling roda robot yaitu sebesar 2*3.14*5cm=31.4cm sehingga jarak tempuh untuk 1 cm yaitu sebesar 356ppr/31.4cm=11.3376 *pulsa*. Jarak tempuh total dari robot didapatkan dari persamaan 3.16. setelah semuanya telah selesai dihitung dan mendapatkan hasil dari jarak tempuh total robot dan derajat tempuh dari robot maka data akan dikirim sesuai dengan format pengiriman data serial seperti pada tabel 3.4.

3.3. Langkah Pengujian

untuk menyatakan sebuah sistem ini telah berhasil berjalan atau berkerja dengan baik maka akan dilakukan pengujian terhadap setiap perangkan dan modul yang akan digunakan, Dalam langkah pengujian ini, pengujian dibagi atas beberapa modul atau perangkat secara terpisah terlebih dahulu, yang kemudian akhirnya akan digabungkan menjadi satu pengujian secara keseluruhan. Langkah yang akan dilakukan untuk pengujian ini sesuai dengan gambar blok diagram 3.15 berikut.



Gambar 3.15. Langkah-Langkah Pengujian

Pengujian yang pertama kali dilakukan adalah pengujian terhadap modul regulator yang digunakan untuk memberikan sumber tegangan pada semua perangkat yang akan digunakan nantinya, pengujian driver motor sebagai pengujian kontrol kecepatan dan arah putaran dari motor DC, pengujian minimum sistem sebagai otak dari sistem yang akan digunakan, pengujian pengiriman serial yang akan dilakukan antara mikrokontroler dengan perangkat computer, pengujian program PID untuk mengatur kecepatan putaran dari motor DC agar selalu constant atau tetap, pengujian omni-directional sebagai pengujian gerakan dari mobile robot ini. Yang terakhir adalah pengujian mobile robot ini secara keseluruhan dimana semua sistem dari mobile robot ini akan digunakan seutuhnya.

3.3.1. Pengujian Rangkaian Regulator

Pengujian rangkaian *regulator* ini adalah untuk memastikan hasil luaran rangkaian ini berada pada tegangan operasi dari perangkat *minimum* sistem, *access point, magnetic encoder* dan perangkat lainnya. Prosedur pengujian dari rangkaian *regulator* yang akan digunakan pada penelitian ini adalah sebagai berikut.

- 1. Nyalakan catu daya pada power supply.
- 2. Atur tegangan input dari power supply antara 9 Volt hingga 12 Volt.
- 3. Hubungkan catu daya dengan rangkaian input regulator.
- 4. Gunakan *AvoMeter* pada kaki *output* dari rangkaian untuk melihat besar tegangan keluarannya.

Tabel 3.6. Pengujian Rangkaian Regulator

| Tujuan | Alat dan Bahan | Input | Output yang | Indikator |
|----------------|----------------|-------------------|------------------|----------------|
| | | | diharapkan | Keberhasilan |
| Untuk | 1. Power | Tegangan | Hasil Keluaran | Hasil keluaran |
| mengetahui | Supply | <i>Input</i> dari | yang ditunjukan | yang |
| hasil keluaran | 2. Rangkaian | batteray | oleh AvoMeter | ditunjunkan |
| dari rangkaian | Regulator | sebesar 12V | adalah sebesar 5 | dari |
| regulator | 3. AvoMeter | | Volt | AvoMeter |
| | | | | adalah sebesar |
| | | | | 5 Volt |

3.3.2. Pengujian Rangkaian *Driver* Motor

Tujuan dari pengujian ini adalah untuk mengetahui apakah *driver* motor yang akan digunakan ini dapat menggerakkan motor DC dengan kecepatan yang berbeda dan dapat mengatur arah dari putaran motor DC yang ada. Prosedur pengujian dari *driver* motor yang akan digunakan pada penelitian ini adalah sebagai berikut.

- 1. Nyalakan catu daya pada power supply menjadi 12V
- 2. Hubungkan kabel power dari motor DC 12V ke connector motor dari L298H.
- 3. Hubungkan semua kaki *input* dari rangkaian *driver* motor ke kabel data.
- 4. Untuk menguji putaran motor, berikan *input* 5V dari power supply ke kabel data pada kaki PWM dan 1A1 atau 2A1.
- 5. Untuk menguji putaran motor yang kearah sebaliknya, Berikan *input* 5V dari *power supply* ke kabel data pada kaki PWM dan 1A2 atau 2A2.

- 6. Untuk memberhentikan motor berikan *input* 0V dari power supply ke kabel data pada kaki PWM dan 1A1 dan 1A2.
- Untuk mengatur kecepatan putaran motor, semakin besar tegangan yang ada pada kaki pin PWM maka akan semakin cepat pula perputaran dari motor DC 12V tersebut.

Tabel 3.7. Pengujian Rangkaian *Driver* Motor

| Tujuan | Alat dan | Input | Output yang | Indikator |
|--------------|-------------|----------------|--------------|--------------|
| | Bahan | | diharapkan | Keberhasilan |
| Untuk | 1. Power | 1. Kaki 1A1 | 1. Motor | Motor |
| mengetahui | Supply | diberikan | berputar | berputar dua |
| kerja dari | 2. Rangkaia | tegangan 5V | dua arah. | arah |
| driver motor | Driver | dan 1A2 | 2. Kecepatan | Tampilan |
| | Motor | diberikan 0V | Dapat | RPM meter |
| | 3. RPM | dan sebaliknya | berbeda | berbeda. |
| | Meter | 2. Kaki PWM1 | | |
| | | diberikan | | |
| | | tegangan yang | | |
| | | berbeda | ITAC | |

3.3.3. Pengujian Rangkaian Minimum Sistem

Tujuan dari pengujian rangkaian *minimum* sistem ini adalah untuk mengetahui apakah modul ini dapat digunakan sebagai otak dari pengerak sistem yang ada. Prosedur pengujian dari *minimum* sistem yang akan digunakan dalam penelitian ini adalah sebagai berikut.

- 1. Aktifkan *power supply* dan hubungkan dengan *minimum* sistem.
- 2. Pasangkan Atmega32 pada rangkaian *minimum* sistem.
- 3. Sambungkan *minimum* sistem dengan *downloader* dengan kabel USB to mikrokontroler.
- 4. Selanjutnya nyalakan komputer dan jalankan program *Code Vision AVR*.

- 5. Untuk *download* program yang telah dibuat kedalam *minimum* sistem maka yang harus dilakukan adalah menjalankan menu *Chip Signature programmer* pada *Code Vision AVR*.
- 6. Setelah proses *signature* selesai maka selanjutnya proses *compile project* dengan menekan F9 pada *keyboard* kemudian proses *download* program ke *microcontroler* masuk ke *menu* → *make project* pada *Code Vision AVR*.

Tabel 3.8. Pengujian Rangkaian *Minimum* Sistem

| Tujuan | Alat dan Bahan | | Input | | utput yang | Indikator | | | | |
|------------|----------------|----|------------------|------------|------------|----------------------|--|--|--|--|
| | | | | diharapkan | | Keberhasilan | | | | |
| Untuk | 1. Downloader | 1. | Klik <i>Read</i> | 1. | Read | Tampilan Read | | | | |
| mengetahui | ISP | | Chip pada | | Success | Success | | | | |
| keadaan | 2. Rangkaian | | program | 2. | Load | Tampilan <i>Load</i> | | | | |
| minimum | Minimum | | CVAVR. | | Success | Program | | | | |
| sistem | sistem | 2. | Klik <i>Load</i> | | | Success | | | | |
| | 3. Power | | Program | | | | | | | |
| | Supply | | pada | | | | | | | |
| | | | CVAVR. | - - | TAC | | | | | |

3.3.4. Pengujian Rangkaian MAX232 dan pengiriman serial

Pengujian rangkaian MAX232 bertujuan untuk mengubungkan komunikasi yang terjadi antara mikrokontroler dengan *computer* yang pengirimannya dilakukan secara serial. Prosedur pengujian dari rangkaian MAX232 yang akan digunakan pada *Penelitian* ini adalah sebagai berikut.

- 1. Nyalakan catu daya pada power supply dengan input 5 Volt DC.
- Hubungkan kaki Rx dan Tx dari *minimum* sistem dan wiznet WIZ110SR dengan MAX232 sesuai pada tabel 3.3.
- 3. Nyalakan Komputer dan mobile robot.
- Kirimkan data dari PC/Laptop yang akan di seleksi oleh mikrokontroler untuk menyalakan dan mematikan lampu led. Adapun program pada visual basic yaitu sebagai berikut.

Adapun program pada mikrokontroler adalah sebagai berikut.

5. Kirimkan data dari mikrokontroler ke PC/laptop dengan program penerima pada *Visual Basic 6.0* adalah sebagai berikut.

```
Private Sub Winsockl_DataArrival(ByVal bytesTotal As Long)
Dim data As String
Winsockl.GetData data
Textl.Text = Textl.Text + data
End Sub
```

Potongan untuk program pengiriman yang ada pada mikrokontroler adalah sebagai berikut.

```
Puts ("ADITYA");
While (1)
{
    // Write Your code here
}
```

Tabel 3.9. Pengujian Rangkaian MAX232 dan pengiriman serial

| Tujuan | Alat dan Bahan | Input | Output yang | Indikator |
|------------|----------------|------------|-----------------|---------------|
| | | | diharapkan | Keberhasilan |
| Untuk | 1. Power | Data | Apa yang | Tampilan dari |
| mengetahui | Supply | pengiriman | dikirimkan sama | program |
| hasil | 2. Rangkaian | serial | dengan apa yang | Visual Basic |
| pengiriman | MAX232 | | ditampilkan dan | 6.0 dan |
| serial | 3. Laptop | | lampu bisa | minimum |
| | 4. Kabel RG45 | | menyala dan | sistem mampu |
| | | | mati tergantung | menyalakan |
| | | | interuksi | lampu led |

3.3.5. Pengujian Program PID

Pengujian program PID ini bertujuan untuk menghasilkan kecepatan motor DC yang stabil pada ketiga buah motor DC yang digunakan. Prosedur pengujian dari metode PID yang akan digunakan pada *Penelitian* ini adalah sebagai berikut.

- 1. Hubungkan battray 12V DC dengan mobile robot.
- 2. Hidupkan power yang ada pada *mobile robot*.
- 3. Hubungkan *wifi* pada *mobile robot* dengan laptop untuk dapat saling berkomunikasi.
- 4. Buat *mobile robot* untuk mengirimkan kecepatan dari setiap putaran motor DC yang ada sesuai dengan TS (*Time Sampling*).
- 5. Lakukan tunning pada nilai Kp, Ki, dan Kd pada *mobile robot*.
- 6. Set nilai Kp dari yang terendah hingga terus naik sampai terjadi sinyal osilasi.
- 7. Set nilai Kd dari yang terendah hingga terus naik sampai ESS (*Error Stady State*) dari sinyal pada kecepatan robot menjadi stabil.
- 8. Apabila *control* Kp dan Kd saja sudah membuat pergerakkan dari motor DC ini menjadi stabil maka nilai untuk *Control* Ki menjadi opsional.
- Atur koordinat yang dituju oleh robot dan biarkan robot tersebut berjalan.
 Ketika robot berjalan kecepatan dikirimkan kepada laptop dengan time sampling yang sama dengan robot yaitu 50ms.
- 10. Ambil semua data *sample* tersebut kemudian dibuat menjadi grafik pada *MS*. *Excell*.

Tabel 3.10. Pengujian Program PID

| Tujuan | Alat dan Bahan | Input | Output yang | Indikator | |
|------------|-----------------|------------|------------------------|---------------|--|
| | | | diharapkan | Keberhasilan | |
| Untuk | 1. Mobile | Data | Kecepatan | Tampilan | |
| mengetahui | robot | pengiriman | putaran motor | grafik dari | |
| sinyal | 2. Wifi sebagai | serial | DC sesuai | data pada MS. | |
| keluaran | Access | | dengan | Exell sesuai | |
| kecepatan | Point | | <i>setpoint</i> atau | dengan | |
| pada motor | 3. Laptop | | mengalami | setpoint. | |
| DC. | | | sedikit <i>osilasi</i> | | |
| | | | | | |

3.3.6. Pengujian Gerak Omni-directional

Pengujian gerak *omni-directional* ini bertujuan untuk mengerakkan *mobile robot* sesuai dengan apa yang diharapkan. Prosedur pengujian dari gerak *omni-directional* yang akan dilakukan pada penelitian ini adalah sebagai berikut.

- 1. Nyalakan power *Mobile robot*.
- 2. Koneksikan antara *mobile robot* dengan laptop menggunakan *wifi*.
- 3. Buatlah gambar plus dengan sudut siku-siku 90°.
- 4. Buatlah program agar *mobile robot* berputar pada porosnya dengan sudut tertentu. Pengiriman data sesuai dengan tabel 3.4.
- 5. Letakkan *mobile robot* pada garis tengah dari gambar plus tersebut.
- 6. Kirimkan program untuk membuat *mobile robot* bergerak berputar sesuai poros tengah pada gambar.
- 7. Lihat sudut perputaran pada poros lingkaran dengan busur.
- Lakukan uji coba beberapa kali namun dengan mengunakan beberapa sudut yang berbeda.

Tabel 3.11. Pengujian Omni-Directional

| Tujuan | Alat dan Bahan | Input | Output yang | Indikator | |
|---------------|-----------------|---------------|---------------|---------------|--|
| | | | diharapkan | Keberhasilan | |
| Mengerakkan | 1. Mobile | Besar sudut | Mobile robot | Mobile robot | |
| mobile robot. | robot | yang ingin | berputar | berhasil | |
| | 2. Wifi sebagai | dibentuk dari | sesuai dengan | berputar | |
| | Access | visual basic | sudut yang | sesuai dengan | |
| | Point | | ditentukan | sudut yang | |
| | 3. Laptop | | oleh program | ditentukan | |
| | 4. Busur | | | | |
| | 5. Spidol | | | | |

3.3.7. Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan ini bertujuan untuk mengerakkan *mobile robot* sesuai dengan apa yang diharapkan. Prosedur pengujian dari sistem secara keseluruhan yang akan dilakukan pada penelitian ini adalah sebagai berikut.

- 1. Nyalakan power Mobile robot.
- 2. Koneksikan antara *mobile robot* dengan laptop menggunakan wifi.
- 3. Buatlah garis *vertikal* dan *horizontal* sebagai pembagi area sesuai kuadran garis ini akan menjadi garis sumbu X dan sumbu Y.
- 4. Letakkan *mobile robot* tepat pada posisi koordinat (0,0) pada garis ke-3.
- Kirimkan koordinat tujuan kepada robot sebanyak 10 kali untuk setiap kuadran yang ada.
- 6. Catat setiap hasil gerakkan *mobile robot* terhadap sumbu X dan sumbu Y.

Tabel 3.12. Pengujian Omni-Directional

| | | - 6 · J · · · | | |
|-----------------|-----------------|---------------|---------------|--------------|
| Tujuan | Alat dan Bahan | Input | Output yang | Indikator |
| | | | diharapkan | Keberhasilan |
| Mengerakka | 1. Mobile | Koordinat | Mobile robot | Mobile robot |
| n <i>mobile</i> | robot | Tujuan dari | bergerak | berhasil |
| robot. | 2. Wifi sebagai | laptop | sesuai dengan | mencapai |
| | Access | | koordinat | koordinat |
| | Point | | yang | yang |
| | 3. Laptop | | diinginkan | diinginkan |
| | 4. Pengaris | | | |



BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian Perangkat Keras

Sesuai dengan langkah pengujian perangkat keras yang ada sebelumnya maka didapatkan hasil dari pengujian pada setiap modul sebagai berikut.

4.1.1. Hasil Pengujian Rangkaian Regulator

A. Hasil Pengujian

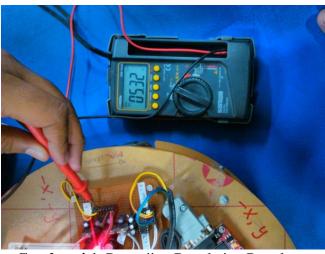
Sesuai dengan langkah pengujian yang dilakukan, didapatkan hasil tegangan dari rangkaian regulator ini seperti tabel 4.1 berikut ini.

Tabel 4.1. Hasil Pengujian Rangkaian Regulator

Pengujian Ke Luaran

| Pengujian Ke | Luaran |
|--------------|-----------------|
| 11151 | Tegangan (Volt) |
| 1 UN | 5,32 Volt |
| 2 | 5,31 Volt |
| 3 | 5,28 Volt |
| 4 | 5,30 Volt |
| 5 | 5, 29 Volt |
| Rata-rata | 5,30 |

Dapat dilihat pula hasil dari pengujian ini pada gambar 4.1 yang ada berikut ini.



Gambar 4.1. Pengujian Rangkaian Regulator

B. Pembahasan

Sesuai dengan hasil pengujian yang sudah dilakukan sebanyak lima kali dan telah sesuai dengan langkah pengujian yang ada, didapatkan bahwa hasil keluaran tegangan dari rangkaian regulator ini adalah sebesar 5.3 volt. Hasil tegangan keluaran ini telah sesuai dengan range kerja dari rangkaian mikrokontroller dan modul lainya maka rangkaian ini dapat dinyatakan telah berhasil dan dapat digunakan lebih lanjut.

4.1.2. Hasil Pengujian Rangkaian Driver Motor

A. Hasil Pengujian

Dari langkah pengujian yang telah dilakukan maka didapatkan hasil dari pengujian untuk mengendalikan putaran motor DC 12Volt ini kearah kiri adalah sebagai berikut.



Gambar 4.2. Motor DC Berputar Kearah Kiri

Dari langkah pengujian yang telah dilakukan maka didapatkan hasil dari pengujian untuk mengendalikan putaran motor DC 12Volt ini kearah kanan adalah sebagai berikut.



Gambar 4.3. Motor DC Berputar Kearah Kanan

Dengan mengikuti langkah pengujian untuk pengaturan kecepatan putaran motor DC 12Volt ini didapatkan hasil dari pengujiannya seperti gambar 4.4 berikut.





Gambar 4.4. Perbedaan Kecepatan Putaran Motor

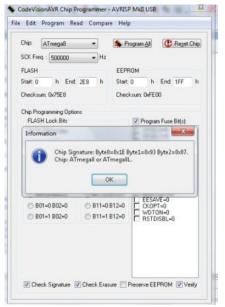
B. Pembahasan

Dari hasil pengujian ingin didapatkan bahwa motor DC 12V telah dapat berputar ke dua arah yang berlawanan dengan kecepatan yang dapat berubah-ubah sensuai dengan input tegangan pada pin PWM. Pada pengujian ini dilakukan dalam kurun waktu kurang lebih 5 menit.

4.1.3. Hasil Pengujian Rangkaian *Minimum* Sistem

A. Hasil Pengujian

Setelah mengikuti langkah pengujian rangkaian *minimum* sistem maka didapatkan hasil *chip signature programmer* seperti gambar 4.5 berikut ini.



Gambar 4.5. Tampilan Chip Signature

Untuk hasil dari donwload program ke rangkaian minimum sistem dapat dilihat seperti gambar 4.6 berikut ini.



B. Pembahasan

Dari percobaan pada gambar 4.5 menu chip signature programmer, dan gambar 4.6 download program berhasil dilakukan maka minimum sistem dapat bekerja dengan baik. Tampilan dari program chip signature pada Code Vision AVR yang akan digunakan untuk menuliskan program dan melakukan percobaan terhadap minimum sistem. Pada Gambar 4.6 menunjukan bahwa minimum sistem telah berhasil melakukan download program ke mikrokontroler sehingga program telah berhasil dijalankan.

4.2. Hasil Pengujian Perangkat Lunak

Sesuai dengan langkah pengujian perangkat lunak yang ada sebelumnya maka didapatkan hasil dari pengujian pada setiap percobaan sesuai dengan langkah pengujian adalah sebagai berikut.

4.2.1. Hasil Pengujian Progam Pengirimanan Serial

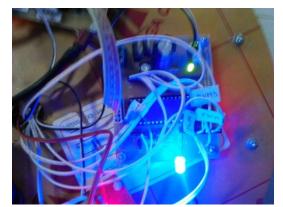
A. Hasil Pengujian

Hasil pengujian dari pengiriman serial ini dapat dilihat pada gambar 4.7 berikut.



Gambar 4.7. Pengiriman dari mikrokontroler ke Laptop

Setelah mikrokontroler mengirimkan data ke laptop pengujian selanjutnya yaitu pengiriman data dari laptop ke mikrokontroler untuk menyalakan lampu led sesuai dengan prosedur pengujian yang ada. Adapun hasil dari pengiriman data tersebut seperti gambar 4.8 dan gambar 4.9 berikut ini.



Gambar 4.8. Hasil Pengiriman Intruksi Data Satu



Gambar 4.9. Hasil Pengiriman Intruksi Data Dua

Apabila tombol "Lampu *On*" ditekan maka *microcontroller* akan menyalakan lampu led berwarna biru seperti pada gambar 4.8 dan pada saat tombol "Lampu *Off*" ditekan maka lampu led berwarna biru tersebut akan mati seperti gambar 4.9.

B. Pembahasan

Dari hasil pengujian ini didapatkan bahwa PC/laptop dapat menerima data yang dikirimkan oleh mikrokontroler dan begitu juga sebaliknya mikrokontroler dapat menerima pengiriman yang dilakukan oleh komputer.

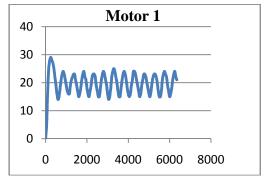
4.2.2. Hasil Pengujian metode PID

Penentuan nilai Kp, Ki, dan Kd pada penelitian ini ditentukan dengan metode *Ziegler-Nichols* yang pertama yaitu metode osilasi dimana awal mulanya melakukan tunning terhadap nilai Kp terlebih dahulu kemudian nilai Ki, dan Kd. Adapun hasil dari pengujian tersebut akan disajikan dalam bentuk grafik pada gambar 4.10 hingga 4.14. Untuk pengujian semua percobaan ini digunakan setpoint dari PID yaitu sebesar 20 *pulsa* untuk setiap *time sampling* (50ms).

A. Kontrol P (*Proporsional*)

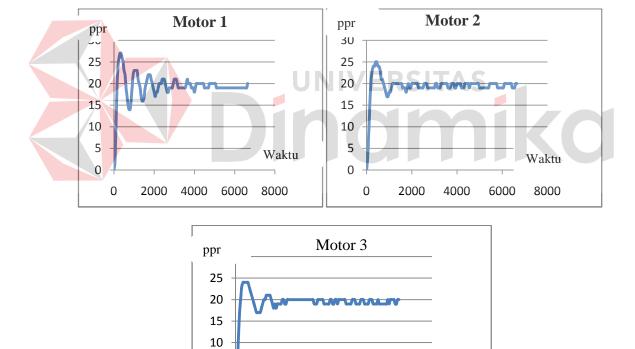
Sesuai dengan metode penentuan nilai PID oleh *Ziegler-Nichols* yang pertama yaitu *osilasi*, nilai yang pertama kali ditunning adalah nilai dari kontrol P

hingga terjadi *osilasi* terus menerus seperti gambar 4.10 berikut ini. Nilai untuk *set point* dari semua pengujian ini yaitu 20 *pulsa* per 50 ms.



Gambar 4.10. Sinyal Berosilasi Terus Menerus

Hasil sinyal yang ditunjukan pada gambar 4.11 berikut ini merupakan *tunning* dari nilai Kp sebesar 3,5.



Gambar 4.11. Control P (Proporsional).

6000

4000

Waktu

8000

5

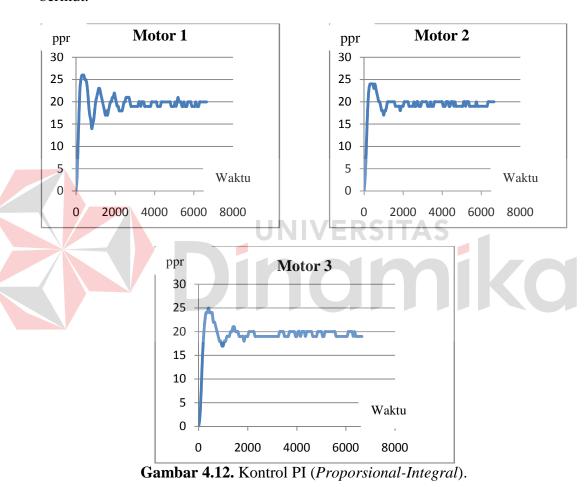
0

0

2000

B. Kontrol PI (Proporsional-Integral)

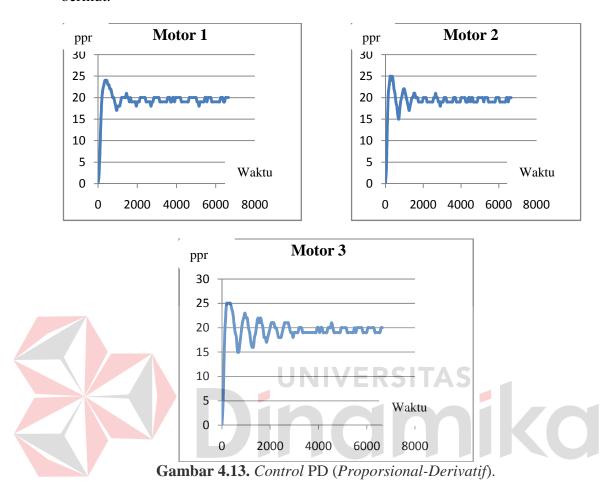
Hasil tunning untuk sinyal control P dan I sesuai dengan gambar 4.12 berikut, dimana nilai Kp yang digunakan adalah sebesar 2,8 dan Ki adalah sebesar 5,6. Nilai *set point* dari pengujian ini adalah sebesar 20 *pulsa* per 50ms *time sampling* yang digunakan sehingga menghasilkan grafik kontrol PI sebagai berikut.



C. Kontrol PD (*Proporsional-Derivative*)

Hasil tunning untuk sinyal kontrol P dan D sesuai dengan gambar 4.13 berikut, dimana nilai Kp yang digunakan adalah sebesar 7 dan Kd adalah sebesar 5. Nilai *set point* dari pengujian ini adalah sebesar 20 *pulsa* per 50ms *time*

sampling yang digunakan sehingga menghasilkan grafik kontrol PD sebagai berikut.



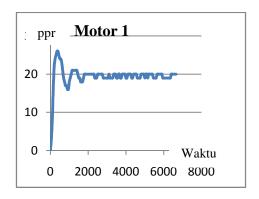
D. Kontrol PID (*Proporsional-Integral-Derivative*)

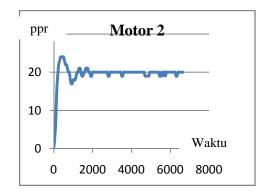
Pada Tugas Akhir kali ini penulis menggunakan nilai dari Kp = 2.8, Kd = 0.84 dan Ki = 5.6 pada ketiga buah motor DC yang ada. Adapun data yang akan ditampilkan berikut ini hanya satu sampel saja diambil dari mikrokontroler, data tersebut dapat dilihat pada tabel 4.2 berikut.

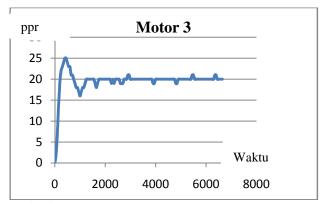
Tabel 4.2. Sampel Data PID Motor 1

| No | Waktu | PPR |
|----|-------|-----|----|-------|-----|----|-------|-----|----|-------|-----|
| 1 | 0 | 0 | 21 | 1000 | 18 | 41 | 2000 | 20 | 61 | 3000 | 19 |
| 2 | 50 | 3 | 22 | 1050 | 19 | 42 | 2050 | 20 | 62 | 3050 | 19 |
| 3 | 100 | 8 | 23 | 1100 | 20 | 43 | 2100 | 20 | 63 | 3100 | 20 |
| 4 | 150 | 16 | 24 | 1150 | 21 | 44 | 2150 | 20 | 64 | 3150 | 19 |
| 5 | 200 | 22 | 25 | 1200 | 21 | 45 | 2200 | 20 | 65 | 3200 | 19 |
| 6 | 250 | 24 | 26 | 1250 | 21 | 46 | 2250 | 20 | 66 | 3250 | 19 |
| 7 | 300 | 25 | 27 | 1300 | 21 | 47 | 2300 | 20 | 67 | 3300 | 19 |
| 8 | 350 | 26 | 28 | 1350 | 21 | 48 | 2350 | 19 | 68 | 3350 | 20 |
| 9 | 400 | 26 | 29 | 1400 | 21 | 49 | 2400 | 19 | 69 | 3400 | 20 |
| 10 | 450 | 25 | 30 | 1450 | 20 | 50 | 2450 | 19 | 70 | 3450 | 20 |
| 11 | 500 | 24 | 31 | 1500 | 19 | 51 | 2500 | 20 | 71 | 3500 | 19 |
| 12 | 550 | 24 | 32 | 1550 | 19 | 52 | 2550 | 20 | 72 | 3550 | 19 |
| 13 | 600 | 23 | 33 | 1600 | 18 | 53 | 2600 | 20 | 73 | 3600 | 20 |
| 14 | 650 | 21 | 34 | 1650 | 18 | 54 | 2650 | 20 | 74 | 3650 | 20 |
| 15 | 700 | 19 | 35 | 1700 | 18 | 55 | 2700 | 20 | 75 | 3700 | 19 |
| 16 | 750 | 18 | 36 | 1750 | 19 | 56 | 2750 | 20 | 76 | 3750 | 19 |
| 17 | 800 | 17 | 37 | 1800 | 20 | 57 | 2800 | 19 | 77 | 3800 | 20 |
| 18 | 850 | 17 | 38 | 1850 | 20 | 58 | 2850 | 19 | 78 | 3850 | 20 |
| 19 | 900 | 16 | 39 | 1900 | 20 | 59 | 2900 | 19 | 79 | 3900 | 20 |
| 20 | 950 | 16 | 40 | 1950 | 20 | 60 | 2950 | 19 | 80 | 3950 | 20 |

Hasil pada tabel 4.2 ini diambil dari *mobile* robot pada saat sedang bergerak menju suatu target sehingga hasil grafik dari data yang ada diatas adalah sebagai gambar 4.14. Nilai *set point* dari pengujian ini adalah sebesar 20 *pulsa* per 50ms *time sampling* yang digunakan sehingga menghasilkan grafik kontrol PID sebagai berikut.







Gambar 4.14. Control PID (Proporsional-Integral-Derivatif).

E. Pembahasan

Proses pengambilan kecepatan putaran motor seperti gambar grafik 4.10 hingga 4.14 diatas yaitu dengan mengirimkan data PPR (*Pulsa Per Rotation*) dari putaran motor pada mikrokontroler, setiap 50 ms sesuai dengan *time sampling* PID yang ada. Pengiriman tersebut ditujukan kepada program *Visual Basic* 6.0 di laptop dengan menggunakan perangkat *wireleas* dan *wiznet*, kemudian data PPR tersebut akan ditampilkan pada Visual Basic 6.0 dan dipindahkan ke *Microsoft Excel* untuk dapat dikelolah dan dirubah menjadi grafik.

Sesuai dengan hasil pengujian diatas dengan menggunakan kontrol P saja akan mengakibatkan sistem menjadi banyak berosilasi sampai detik ketiga pada motor satu, hasil ini menyatakan bahwa pengendalian motor dengan kontrol P saja masih kurang dikarenakan waktu yang dibutuhkan hingga sinyal luaran dari motor ini stabil (*rise time*) masih terlalu lama. Dengan ditambahkannya kontrol PD atau PI sistem menjadi sedikit lebih baik daripada hanya menggunakan kontrol P saja namun, hasil luaran dari outputnya masih sedikit berosilasi sehingga percobaan ini memerlukan kontrol PID untuk memperbaiki *rise time* dan luaran dari sistem ini.

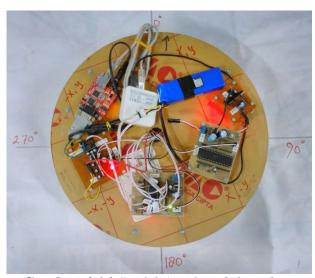
Pengujian ini dilakukan berdasarkan rumusan metode *Ziegler-Nichols* yang melakukan pengujian secara bertahap dimulai dari P kemudian PI dan

terakhir PID untuk berdasarkan beberapa pengumpulan informasi yang peneliti dapatkan maka peneliti melakukan pengujian dengan menggunakan kontrol PD saja, peneliti mengharapan sistem ini bisa memiliki luaran yang stabil dan *rise time* yang baik sehingga tidak diperlukan penambahan kontrol PID. Namun dari hasil pengujian pada gambar 4.10 hingga gambar 4.14 maka didapatkan hasil pengujian yang paling stabil untuk mengendalikan kecepatan putaran dari motor DC 12 Volt ini yang sesuai pada *setpoint* yaitu dengan nilai Kp = 2.8, Kd = 0.84 dan Ki = 5.6. Sistem ini akan dikatakan stabil ketika kecepatan dari putaran motor DC ini konstant atau tetap secara terus menerus.

4.2.3. Hasil Pengujian Gerak Omni-Directional

A. Hasil Pengujian

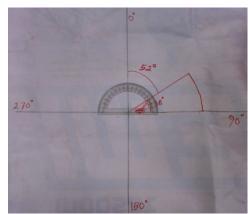
Pengujian ini dilakukan dengan memutar badan dari *Mobile robot Triangle Omni-directional* ini untuk mengetahui hasil dari sudut atau derajat perputaran yang dilakukan. Hasil dari percobaan ini dapat dilihat seperti pada gambar 4.16 hingga gambar 4.17 berikut.



Gambar 4.16. Posisi Awal Mobile Robot.

Pada gambar 4.16 posisi awal robot berada pada 0° kemudian robot bergerak hingga posisi seperti gambar 4.17 dengan sudut seperti gambar.





Gambar 4.17. Posisi Setelah *Mobile Robot* Berputar.

Dari gambar 4.17 diatas dapat dilihat bahwa *mobile robot* ini telah dapat bergerak berputar pada porosnya dengan sudut 52° dari titik awal dia berada.

B. Pembahasan

Dari hasil pengujian ini didapatkan bahwa mobile robot dapat bergerak berputar ditempat sejauh 52° dari posisi awalnya. Dimana pada program visual basic mengrimkan sudut putar sebesar 51° sehingga dapat dikatakan kesalahan yang ada adalah rata-rata sebesar 1°.

4.3. Hasil Pengujian Sistem Secara Keseluruhan

Hasil pengujian keseluruhan dari sistem ini terbagi atas empat kuadran, dimana keempat kuadran tersebut dilakukan ujicoba sebanyak 10 kali percobaan dengan titik koordinat tujuan yang berbeda-beda setiap percobaannya.

4.3.1. Gerak *Mobile Robot* pada Kuadran Pertama

A. Hasil Pengujian

Adapun hasil dari gerak *mobile* robot pada kuadran pertama ini dapat dilihat pada tabel 4.3 berikut. Pada setiap pengujian gerak ini data dimasukkan dengan koordinat tujuan yang berbeda-beda.

Tabel 4.3. Gerak *Mobile* Robot pada Kuadran Pertama.

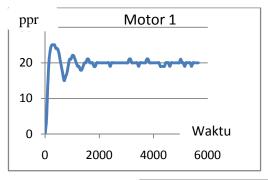
| Uji Coba | Input 1 | Koordi | nat (Cm) | Koordi | nat Aktu | ator (Cm) | Ke | esalahan (Cm) | |
|-------------|---------|--------|----------|--------|----------|-----------|------|---------------|---------|
| Ke | X | Y | Ө | X | Y | Ө | X | Y | Ө |
| 1 | 100 | 100 | 45.022° | 100 | 100 | 45.022° | 0 | 0 | 0° |
| 2 | 150 | 100 | 56.338° | 140 | 105 | 59.515° | -10 | 5 | 3.181° |
| 3 | 100 | 150 | 33.707° | 106 | 148.5 | 31.877° | 6 | -1.5 | -1.83° |
| 4 | 110 | 125 | 41.368° | 105 | 105 | 41.219° | 5 | 5 | -0.149° |
| 5 | 200 | 200 | 45.022° | 193.5 | 102 | 39.605° | -6.5 | 2 | -5.417° |
| 6 | 180 | 170 | 46.660° | 181 | 160 | 44.772° | 1 | -10 | -1.888° |
| 7 | 190 | 170 | 48.204° | 197 | 155 | 44.578° | 7 | -15 | -3.626° |
| 8 | 120 | 190 | 32.292° | 120 | 103 | 32.697° | 0 | 3 | 0.405° |
| 9 | 140 | 110 | 51.869° | 131 | 102 | 54.243° | -9 | 2 | 2.374° |
| 10 | 135 | 170 | 38.473° | 136 | 170 | 38.267° | 1 | 0 | -0.206° |

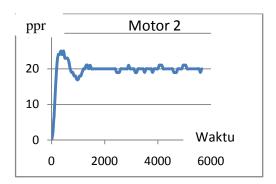
Dari hasil pengujian pada tabel pengujian gerak pada kuadran pertama ini didapatkan hasil kesalahan pada setiap pengujiannya yang kemudian dapat dirubah menjadi presentasi kesalahan seperti pada tabel berikut ini.

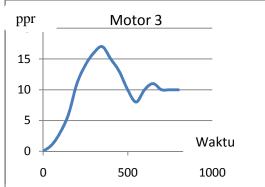
Tabel 4.4. Presentasi Kesalahan Gerak Robot Pada Kuadran Pertama

| Llii Cobo | Preser | ntasi Kesalah | an Terhadap |
|----------------|----------|---------------|-----------------|
| Uji Coba Ke | Sumbu X | Sumbu Y | Sudut Orientasi |
| Ke | (%) | | (%) |
| 1 | 0 | 0 | 0 |
| 2 | 6.66667 | 5 | 5.646278 |
| 3 | 6 | 1 | 5.429139 |
| 4 | 4.545455 | 4 | 0.360182 |
| 5 | 3.25 | 1 | 12.0319 |
| 6 | 0.555556 | 5.882353 | 4.046292 |
| 7 | 3.684211 | 8.823529 | 7.522197 |
| 8 | 0 | 1.578947 | 1.254181 |
| 9 | 6.42857 | 1.818182 | 4.576915 |
| 10 | 0.740741 | 0 | 0.53544 |
| Rata-Rata | 3.18712 | 2.910301 | 4.140252 |

Pada hasil pengujian pergerakkan diatas diambil juga hasil sinyal keluaran PID pada ketiga buah motor DC yang ada, dan hasilnya dapat dilihat sebagai berikut.







Gambar 4.18. Sinyal PID pada Kuadran Pertama

B. Pembahasan

Dari h<mark>asi</mark>l uji coba yang telah dilakukan maka didapatkan rata-rata kesalahan yang terjadi pada kuadran pertama dapat dihitung sebagai berikut.

Rata-rata kesalahan sumbu
$$X = \frac{0 \ 10 \ 6 \ 5 \ 6.5 \ 1 \ 7 \ 0 \ 1}{10} = 4,55 cm$$

Rata-rata kesalahan sumbu
$$Y = \frac{0\ 5\ 1.5\ 5}{10} = \frac{10\ 15\ 3}{10} = 4,35cm$$

Rata-rata kesalahan orientasi sudut =
$$\frac{1.076}{10}$$
 = 1,9076°

Pada hasil pengujian sistem secara keseluruhan ini nilai Θ didapatkan dari perhitunggan secara matematis yang sesuai dengan persamaan 2.16 untuk semua sudut tempuh Θ yang ada.

4.3.2. Gerak *Mobile Robot* pada Kuadran kedua

A. Hasil Pengujian

Adapun hasil dari gerak *mobile* robot pada kuadran kedua ini dapat dilihat pada tabel 4.5. Pada pengujian ini sesuai dengan langkah pengujian sebelumnya nilai dari koordinat tujuan dari setiap percobaan dibuat berbeda-beda sebanyak 10 kali percobaan.

Tabel 4.5. Gerak Mobile Robot pada Kuadran Kedua.

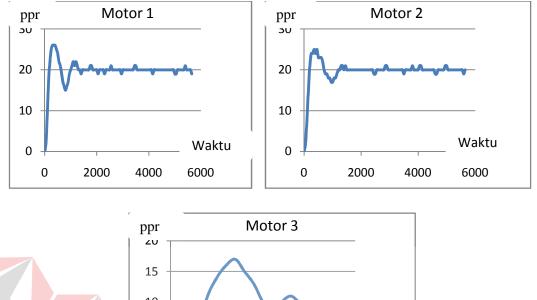
| Uji Coba | Input | Koordi | inat (Cm) | Koord | linat Aktu | ator (Cm) | Kesalahan (Cm) | | |
|-------------|-------|--------|-----------|-------|------------|-----------|----------------|-----|---------|
| Ke | X | Y | θ | X | Y | θ | X | Y | θ |
| 1 | -100 | 100 | -45.022° | -108 | 93 | 45.332° | 8 | -7 | -0.31° |
| 2 | -100 | 150 | -33.707° | -105 | 140 | 33.237° | 5 | -10 | 0.47° |
| 3 | -150 | 100 | -56.338° | -156 | 94 | 55.783° | 6 | -6 | 0.555° |
| 4 | -125 | 110 | -48.676° | -126 | 110 | 48.904° | 1 | 0 | -0.228° |
| 5 | -200 | 200 | -45.022° | -213 | 188 | 45.175° | 13 | -12 | -0.153° |
| 6 | -170 | 180 | -43.385 | -170 | 184 | 44.014° | 0 | 4 | -0.629° |
| 7 | -190 | 120 | -57.753° | -203 | 100 | 54.941° | 13 | -20 | 2.812° |
| 8 | -110 | 140 | -38.176° | -105 | 146 | 38.107° | -5 | 6 | 0.069° |
| 9 | -170 | 135 | -51.572° | -168 | 127 | 49.558° | -2 | -8 | 2.014° |
| 10 | -170 | 190 | -41.841° | 0 | / 185 | 41.08° | 0 | -5 | 0.761° |

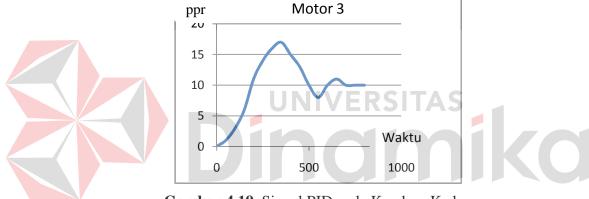
Dari hasil pengujian pada tabel pengujian gerak pada kuadran kedua ini didapatkan hasil kesalahan pada setiap pengujiannya yang kemudian dapat dirubah menjadi presentasi kesalahan seperti pada tabel berikut ini.

Tabel 4.6. Presentasi Kesalahan Gerak Robot Pada Kuadran Kedua

| Uji Coba | Presentasi Kesalahan Terhadap | | | | | |
|-----------|-------------------------------|----------|-----------------|--|--|--|
| Ke Ke | Sumbu X | Sumbu Y | Sudut Orientasi | | | |
| Ke | (%) | | (%) | | | |
| 1 | 8 | 7 | 0.688552 | | | |
| 2 | 5 | 6.666667 | 1.394369 | | | |
| 3 | 4 | 6 | 0.985125 | | | |
| 4 | 0.8 | 0 | 0.468403 | | | |
| 5 | 6.5 | 6 | 0.339834 | | | |
| 6 | 0 | 2.222222 | 1.44981 | | | |
| 7 | 6.842105 | 16.66667 | 4.869011 | | | |
| 8 | 4.545455 | 4.285714 | 0.180742 | | | |
| 9 | 1.176471 | 5.925926 | 3.90522 | | | |
| 10 | 0 | 2.631579 | 1.81879 | | | |
| Rata-Rata | 3.686403 | 5.739877 | 1.609986 | | | |

Pada hasil pengujian pergerakkan diatas diambil juga hasil sinyal keluaran PID pada ketiga buah motor DC yang ada, dan hasilnya dapat dilihat sebagai berikut.





Gambar 4.19. Sinyal PID pada Kuadran Kedua

B. Pembahasan

Rata-rata kesalahan yang terjadi pada kuadran pertama dihitung sebagai berikut.

Rata-rata kesalahan sumbu
$$X = \frac{8 \ 5 \ 6 \ 1 \ 13 \ 0 \ 13 \ 5 \ 0}{10} = 5,3 \text{ cm}$$

Rata-rata kesalahan sumbu
$$Y = \frac{7 \cdot 10 \cdot 6 \cdot 0 \cdot 1}{10} = 7.8 \text{ cm}$$

Rata-rata kesalahan orientasi sudut =
$$\frac{8.001}{10}$$
 = 0.8001°

Pada hasil pengujian sistem secara keseluruhan ini nilai Θ didapatkan dari perhitunggan secara matematis yang sesuai dengan persamaan 2.16 untuk semua sudut tempuh Θ yang ada.

4.3.3. Gerak Mobile Robot pada Kuadran ketiga

A. Hasil Pengujian

Adapun hasil dari gerak *mobile* robot pada kuadran kedua ini dapat dilihat pada tabel 4.6. Pada pengujian ini sesuai dengan langkah pengujian sebelumnya nilai dari koordinat tujuan dari setiap percobaan dibuat berbeda-beda sebanyak 10 kali percobaan.

Tabel 4.7. Gerak *Mobile* Robot pada Kuadran Ketiga.

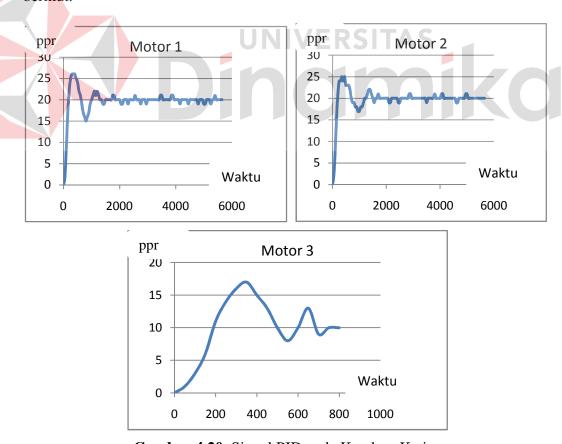
| Tubel 4.7. Gerak Moone Rooot pada Raddran Retiga. | | | | | | | | | |
|---|------|---------|-----------|-------------------------|------|----------|----------------|----|---------|
| Uji Coba | Inpu | t Koord | inat (Cm) | Koordinat Aktuator (Cm) | | | Kesalahan (Cm) | | |
| Ke | X | Y | θ | X | Y | θ | X | Y | Ө |
| 1 | 200 | -200 | -45.022° | 194 | -198 | -47.693° | -6 | 2 | -2.671° |
| 2 | 110 | -120 | -42.532° | 108 | -120 | -43.056° | -2 | 0 | -0.524° |
| 3 | 180 | -110 | -56.338° | 178 | -116 | -55.739° | -2 | -6 | 0.599° |
| 4 | 190 | -150 | -51.736° | 194 | -159 | -52.784° | 4 | -9 | -1.048° |
| 5 | 170 | -100 | -59.564° | 162 | -100 | -60.785° | -8 | 0 | -1.221° |
| 6 | 180 | -110 | -58.600° | 170 | -100 | -62.39° | -10 | 10 | -3.790° |
| 7 | 190 | -115 | -58.844° | 183 | -115 | -61.35° | -7 | 0 | -2.506° |
| 8 | 200 | -100 | -63.467° | 190 | -97 | -65.366° | -10 | 3 | -1.899° |
| 9 | 100 | -200 | -26.578° | 100 | -199 | -26.692° | 0 | 1 | -0.114° |
| 10 | 160 | -180 | -41.654° | 157 | -174 | -43.121° | -3 | 6 | -1.467° |

Dari hasil pengujian pada tabel pengujian gerak pada kuadran ketiga ini didapatkan hasil kesalahan pada setiap pengujiannya yang kemudian dapat dirubah menjadi presentasi kesalahan seperti pada tabel berikut ini.

Tabel 4.8. Presentasi Kesalahan Gerak Robot Pada Kuadran Ketiga

| Liii Cobo | Presentasi Kesalahan Terhadap | | | | |
|----------------|-------------------------------|----------|-----------------|--|--|
| Uji Coba Ke | Sumbu X | Sumbu Y | Sudut Orientasi | | |
| Ke | (%) | (%) | (%) | | |
| 1 | 3 | 1 | 5.932655 | | |
| 2 | 1.818182 | 0 | 1.232014 | | |
| 3 | 1.111111 | 5.454545 | 1.063226 | | |
| 4 | 2.105263 | 6 | 2.025669 | | |
| 5 | 4.705882 | 0 | 2.049896 | | |
| 6 | 5.555556 | 9.090909 | 6.467577 | | |
| 7 | 3.684211 | 0 | 4.258718 | | |
| 8 | 5 | 3 | 2.992106 | | |
| 9 | 0 | 0.5 | 0.428926 | | |
| 10 | 1.875 | 3.333333 | 3.521871 | | |
| Rata-Rata | 2.88552 | 2.837879 | 2.997266 | | |

Pada hasil pengujian pergerakkan diatas diambil juga hasil sinyal keluaran PID pada ketiga buah motor DC yang ada, dan hasilnya dapat dilihat sebagai berikut.



Gambar 4.20. Sinyal PID pada Kuadran Ketiga

B. Pembahasan

Dari hasil pengujian pada tabel 4.6 rata-rata kesalahan yang terjadi pada kuadran pertama dihitung sebagai berikut.

Rata-rata kesalahan sumbu
$$X = \frac{6}{10} \frac{8 \ 10 \ 7 \ 10 \ 0 \ 3}{10} = 5,4 \text{ cm}$$

Rata-rata kesalahan sumbu
$$Y = \frac{0.6}{10.0310} = 3.7 \text{ cm}$$

Rata-rata kesalahan orientasi sudut =
$$\frac{15.38}{10}$$
 = 1,5389°

Pada hasil pengujian sistem secara keseluruhan ini nilai Θ didapatkan dari perhitunggan secara matematis yang sesuai dengan persamaan 2.16 untuk semua sudut tempuh Θ yang ada.

4.3.4. Gerak Mobile Robot pada Kuadran Keempat

A. Hasil Pengujian

Adapun hasil dari gerak mobile robot pada kuadran keempat ini dapat dilihat pada tabel 4.9. Pada pengujian ini sesuai dengan langkah pengujian sebelumnya nilai dari koordinat tujuan dari setiap percobaan dibuat berbeda-beda sebanyak 10 kali percobaan.

UNIVERSITAS

Tabel 4.9. Gerak Mobile Robot pada kuadran ke Empat.

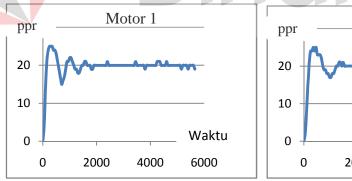
| Uji Coba | Koordinat Tujuan | | | Koordinat Hasil | | Orientasi Objek | | | |
|-------------|------------------|------|---------|-----------------|------|-----------------|-----|----|---------|
| Ke | X | Y | Ө | X | Y | Ө | X | Y | Ө |
| 1 | -100 | -100 | 45.022° | -104 | -96 | 47.314° | 4 | -4 | -2.292° |
| 2 | -150 | -100 | 56.338° | -149 | -104 | 55.113° | -1 | 4 | 1.225° |
| 3 | -100 | -150 | 33.707° | -92 | -156 | 30.545° | -8 | 6 | 3.162° |
| 4 | -110 | -125 | 41.368° | -112 | -123 | 42.341° | 2 | -2 | -0.973° |
| 5 | -200 | -200 | 45.022° | -190 | -210 | 42.158° | -10 | 10 | 2.864° |
| 6 | -190 | -170 | 48.204° | -192 | -168 | 48.838° | 2 | -2 | -0.634° |
| 7 | -120 | -190 | 32.292° | -130 | -182 | 35.555° | 10 | -8 | -3.263° |
| 8 | -140 | -110 | 51.869° | -132 | -120 | 47.75° | -8 | 10 | 4.119° |
| 9 | -135 | -170 | 38.473° | -144 | -165 | 41.132° | 9 | -5 | -2.659° |
| 10 | -170 | -190 | 41.841° | -178 | -183 | 44.228° | 8 | -7 | -2.387° |

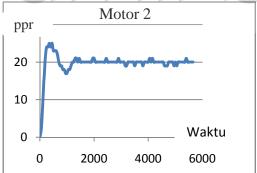
Dari hasil pengujian pada tabel pengujian gerak pada kuadran ke empat ini didapatkan hasil kesalahan pada setiap pengujiannya yang kemudian dapat dirubah menjadi presentasi kesalahan seperti pada tabel berikut ini.

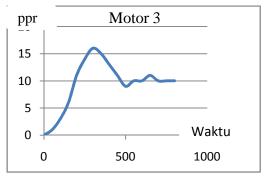
Tabel 4.10. Presentasi Kesalahan Gerak Robot Pada Kuadran Ke empat

| Hii Cobo | Presentasi Kesalahan Terhadap | | | | | |
|----------------|-------------------------------|----------|-----------------|--|--|--|
| Uji Coba Ke | Sumbu X | Sumbu Y | Sudut Orientasi | | | |
| Ke | (%) | (%) | (%) | | | |
| 1 | 4 | 4 | 5.090844 | | | |
| 2 | 0.666667 | 4 | 2.174376 | | | |
| 3 | 8 | 4 | 9.380841 | | | |
| 4 | 1.818182 | 1.6 | 2.35206 | | | |
| 5 | 5 | 5 | 6.361334 | | | |
| 6 | 1.052632 | 1.176471 | 1.315244 | | | |
| 7 | 8.333333 | 4.210526 | 10.10467 | | | |
| 8 | 5.714286 | 9.090909 | 7.941159 | | | |
| 9 | 6.666667 | 2.941176 | 6.91134 | | | |
| 10 | 4.705882 | 3.684211 | 5.704931 | | | |
| Rata-Rata | 4.595765 | 3.970329 | 5.73368 | | | |

Pada hasil pengujian pergerakkan diatas diambil juga hasil sinyal keluaran PID pada ketiga buah motor DC yang ada, dan hasilnya dapat dilihat sebagai berikut.







Gambar 4.21. Sinyal PID pada Kuadran Keempat

B. Pembahasan

Rata-rata kesalahan yang terjadi pada kuadran keempat dihitung sebagai berikut.

Rata-rata kesalahan sumbu
$$X = \frac{1 \ 8 \ 10 \ 10 \ 8 \ 8}{10} = 7,2 \text{ cm}$$

Rata-rata kesalahan sumbu Y =
$$\frac{6}{10}$$
 $\frac{10}{10}$ $\frac{8}{10}$ $\frac{10}{5}$ $\frac{7}{7}$ = 5,8 cm

Rata-rata kesalahan orientasi sudut =
$$\frac{53588}{10}$$
 = 2.3588°

Pada hasil pengujian sistem secara keseluruhan ini nilai Θ didapatkan dari perhitunggan secara matematis yang sesuai dengan persamaan 2.16 untuk semua sudut tempuh Θ yang ada.

4.3.5. Pembahasan Keseluruhan

Dari hasil pengujian seperti pada tabel 4.4 sampai 4.10 maka didapatkan nilai rata-rata kesalahan pada kuadran pertama 4,55 cm pada sumbu X dan 4,35 cm pada sumbu Y dengan sudut orientasi kesalahan 1,907°, kuadran kedua 53 cm pada sumbu X dan 7,8 cm pada sumbu Y dengan sudut orientasi kesalahan 0,8°, pada kuadran ketiga 5,4 cm pada sumbu X dan 3,7 cm pada sumbu Y dengan sudut orientasi kesalahan 1,539°, kuadran keempat 7.2 cm pada sumbu X dan 5,8 cm pada sumbu Y dengan sudut orientasi kesalahan 2,358°. Maka didapatkan rata-rata kesalahan gerak total untuk *mobile robot* ini adalah sebagai berikut.

Rata-rata kesalahan sumbu
$$X = \frac{.55 + 5.3 + 5. + 7.}{.} = 5,6125 \text{ cm}$$

Rata-rata kesalahan sumbu
$$Y = \frac{.35 \ 7.8 \ 3.7 \ 5.8}{... \ 5.4125 \ cm} = 5,4125 \ cm$$

Rata-rata kesalahan orientasi sudut =
$$\frac{1.0760.80011.538}{1.538} = 1,6513$$

Dari rata-rata penghitungan hasil kesalahan terhadap gerak dari *mobile robot* ini dapat kita rubah nilai kesalahannya menjadi presentasi kesalahan gerak *mobile robot* sebagai berikut ini.

Persentase kesalahan terhadap sumbu $X = \frac{3.187 \ 3.686 \ 2.885 \ 4.596}{2.885 \ 4.596} = 3.5885\%$ Persentase kesalahan terhadap sumbu $Y = \frac{2.910 \ 5.740 \ 2.838 \ 3.970}{2.838 \ 3.970} = 3.8645\%$ Persentase kesalahan terhadap sudut $\Theta = \frac{4.140 \ 1.610 \ 2.997 \ 5.734}{2.897 \ 5.734} = 3.62052\%$

Pada hasil pengujian sistem secara keseluruhan ini nilai θ didapatkan dari perhitunggan secara matematis yang sesuai dengan persamaan 2.16 untuk semua sudut tempuh θ yang ada. Rata-rata kesalahan sebesar 5,6 cm terhadap sumbu koordinat yang diinginkan ini didasarkan terhadap kecepatan putaran dari motor DC yang terlalu tinggi dan motor DC ini memiliki ppr sebesar 356 sehingga mengakibatkan sistem ini mengalami loss data.

Dimana apabila kecepatan putaran atau *set point* dari motor DC ini diatur sebesar 20ppr setiap 50 ms maka kecepatan putaran dari motor dapat dihitung sebesar 20*(50*20)/356 = 56.1798 RPS (*rotation per second*). Ketika kecepatan putaran dari motor ini sebesar 56.1798 RPS maka pulsa yang akan dikirimkan oleh *rotary encoder* kepada mikrokontroller adalah sebesar 56.1798*356 = 20000 Hz = 20Khz per motor yang digunakan. Ketika dua buah motor DC berputar bersamaan maka kecepatan *input* yang diterima oleh mikrokontroller menjadi lebih tinggi lagi, sehingga *loss* data *input* dari *rotary encoder* akan yang terjadi menjadi besar.



BAB V

PENUTUP

Berdasarkan hasil pengujian pada perangkat keras dan perangkat lunak yang dipergunakan dalam perancangan *mobile robot* dengan *triangle omnidirectional wheels* ini, maka dapat diambil beberapa kesimpulan dan saran sebagai berikut.

5.1. Kesimpulan

Adapun kesimpulan yang dapat dituliskan setelah melakukan analisa dari hasil sistem yang telah dibuat antara lain sebagai berikut :

- 1. Pengujian gerak *omni-directional wheels* menunjukan bahwa *mobile robot* ini dapat bergerak kesegala arah sesuai dengan koordinat yang dikirmkan oleh program pada Visual Basic 6.0.
- 2. Rata-rata keseluruhan dari kesalahan pergerakkan *mobile robot* ini adalah sebesar 3.5885% terhadap sumbu X, 3.8645% terhadap sumbu Y, dan sebesar 3.6205% terhadap orientasi sudut yang dibuat.
- 3. Penentuan nilai Kp, Ki, dan Kd dengan menggunakan metode *Ziegler-Nichols* yang pertama yaitu "*osilasi*" menghasilkan sinyal keluaran motor DC yang cukup stabil pada ketiga motor DC yang digunakan, dari hasil pengujian ini didapatkan nilai PID yang stabil untuk ketiga buat motor DC ini adalah Kp = 2.8, Ki = 5.6, dan nilai dari Kd = 0.84.

4. Dengan menggunakan metode PID kecepatan putaran roda dari *mobile robot* ini dapat diatur dengan baik dan memiliki kecepatan yang stabil sesuai dengan sinyal hasil keluaran mikrokontroller, Kecepatan dari putaran roda *mobile robot* ini terbagi atas dua yaitu kecepatan menuju target dan kecepatan membuat sudut gerak.

5.1 Saran

Saran yang dapat diberikan untuk mengembangkan sistem ini sebagai berikut:

- 1. Dalam perancangan *minimum* sistem sebagai pusat pengendali *mobile robot* ini, selanjutnya diharapkan untuk menggunakan mikrokontroler ARM Cortex dikarenakan kecepatan exekusi program pada ARM Cortex memiliki kecepatan yang lebih tinggi dari ATmega32.
- 2. Terdapat beberapa cara yang mungkin bisa dicoba saat menentukan posisi tujuan.
- 3. Koordinat tujuan yang akan dicapai selanjutnya bisa lebih dari satu koordinat tujuan saja sehingga robot dapat bergerak lebih banyak lagi.

Pada program Visual Basic 6.0 dapat ditambahkan sebuah gambar yang memberikan *feedback* terhadap posisi dan arah pada *mobile robot* pada saat bergerak.

DAFTAR PUSTAKA

- Alaydrus, Riza. 2012. Rancang Bangun Self Tuning PID Kontrol pH Dengan Metode Pencarian Akar Persamaan Karakteristik. Institut Teknologi Sepuluh Nopember (ITS): Surabaya.
- Ardila, Fernando. 2011. *Path tracking pada mobile robot dengan umpan balik odometry*. Electronic engineering polytechnic institute of Surabaya (PENS): Surabaya.
- Bachri, Samsul. 2004. Sistem Kendali Hybrid Pid Logika Fuzzy Pada Pengaturan Kecepatan Motor Dc. Universitas Jember: Jember.
- Diegel, Olaf. 2002. *Improved Mecanum Wheel Design For Omni-Directional Robots*. Massey University: Auckland.
- Fahmi, Ibnu. 2012. Kendali PID dan Logika Fuzzy untuk Optimalisasi Pergerakan Mobile Robot. STIKOM Surabaya:Surabaya.
- Firdaus, Akhmad. 2010. Implementasi Pengaturan Kecepatan Motor Induksi Satu Phase dengan PID Fuzzy pada Pompa Air. STIKOM Surabaya: Surabaya.
- Liu, Yong. 2007. Omni-directional mobile robot controller based on trajectory linearization. Ohio University: Athens.
- Muthusubramanian, R. 2000. *Basic Electrical, Electronics, and Computer Engineering*. Tata McGraw-Hill: New Delhi.
- Syam, Rafiuddin. 2012. Rancang Bangun Omni Wheels Robot Dengan Roda Penggerak Independent. Universitas Hasanudin (UNHAS): Makassar.
- Pasila, Felix. 2000. Pengendalian pengaturan kecepatan motor DC menggunakan kendali hybrid PID-Fuzzy. Institut Teknologi Sepuluh Nopember (ITS): Surabaya.