

PERANCANGAN DAN IMPLEMENTASI PENGENDALI PID PADA SUBSISTEM AKTUATOR ROBOT MOBIL TIPE SINKRON

Muhammad Ilhamdi Rusydi

Jurusan Teknik Elektro, Universitas Andalas Padang,

Kampus Limau Manis, Padang, Sumatera Barat

Email : rilhamdi@yahoo.com

ABSTRAK

Pada tulisan ini dipaparkan proses perancangan beserta implementasi sistem pengendalian pergerakan robot mobil. Algoritma kendali PID secara digital digunakan dalam sistem kendali lingkaran tertutup yang difungsikan untuk mengendalikan posisi, orientasi dan kecepatan robot mobil dengan tipe pergerakan *Synchronous Steering*. Implementasi dilakukan menggunakan mikrokontroler PIC 16F877 sebagai pengendali. Proses pencarian konstanta pengendali proporsional, integral dan derivatif dilakukan secara eksperimental berdasarkan grafik respon waktu yang diperoleh. Pada tahap akhir pengujian dilakukan dengan menghubungkan pengendali ke komputer, sehingga unjuk-kerja sistem dapat diamati. Hasil pengujian menunjukkan sistem kendali orientasi memiliki ketelitian 1 derajat, sedangkan sistem kendali posisi memiliki galat keadaan tunak sebesar 0,5 cm.

Kata Kunci- Mobile Robot, PID, *Synchronous Steering*

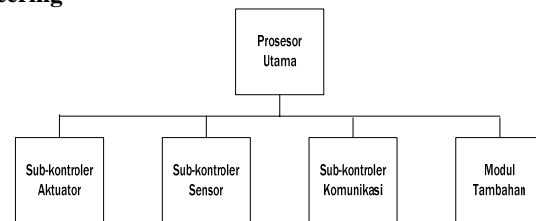
1. PENDAHULUAN

Robotika merupakan suatu bidang kajian multidisiplin yang terus berkembang hingga saat ini. Begitu banyak aplikasi ilmu robotika yang dapat ditemukan dalam kehidupan. Mulai dari proses produksi di pabrik-pabrik, peralatan konstruksi, peralatan operasi medis, penjelajahan ke tempat-tempat yang berbahaya bagi manusia, hingga permainan anak-anak.

Mobile robotics adalah salah satu cabang dari bidang ilmu robotika. Keistimewaan robot mobil adalah kemampuannya untuk dapat bergerak dan berpindah tempat secara bebas tanpa adanya ikatan yang membatasi ruang kerjanya. Karena mobilitasnya itulah robot mobil sering digunakan untuk menjelajahi tempat – tempat yang berbahaya bagi manusia. Pengambilan sampel bebatuan di bulan dan penelusuran keadaan di dasar laut adalah contoh dari aplikasi robot mobil.

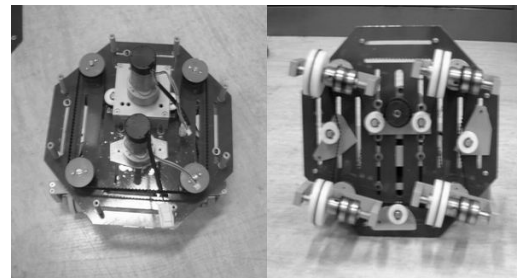
Saat ini Laboratorium Sistem Kendali dan Komputer (LSKK) sedang membangun beberapa *platform* robot mobil yang dapat dikembangkan lebih lanjut. Oleh karena itu, maka dibangun sistem yang bersifat terbuka dan fleksibel terhadap penambahan fungsi ataupun modul di kemudian hari. Arsitektur sistem yang dibangun dapat dilihat pada Gambar 1.

Robot mobil yang sedang dikembangkan di LSCK adalah robot mobil yang beroperasi di darat dan menggunakan roda sebagai alat untuk bergerak dan berpindah tempat. Terdapat beberapa tipe pergerakan untuk robot mobil dengan penggerak roda.



Gambar-1 Arsitektur sistem platform mobile robot yang terbuka

Salah satu robot mobil yang dikembangkan adalah SyncBot. SyncBot memiliki tipe pergerakan *synchronous steering*, sering juga disebut *synchronous drive*. Sesuai dengan namanya, karakteristik dari tipe pergerakan ini adalah beberapa buah roda yang selalu menghadap ke arah yang sama dan berputar secara bersama-sama pula. Realisasi robot mobil tipe ini diperlihatkan oleh Gambar-2.



Gambar- 2 Tampak atas (kiri) dan tampak bawah (tampak kiri) SyncBot.

Agar dapat melakukan pergerakan dengan baik dan terkendali diperlukan suatu subsistem yang secara khusus mengatur dan mengendalikan proses pergerakan pada SyncBot. Proses pengendalian pergerakan SyncBot inilah yang dijadikan topik

pada tulisan ini.

2. TEORI DASAR

2.1 Kendali Proporsional-Integral-Derivatif

Pada umumnya suatu sistem kendali lingkaran tertutup terdiri atas unit pengendali (kontroler), aktuatur dan sensor umpan-balik. Pada Tugas Akhir ini metode pengendalian yang diterapkan adalah kendali posisi dan kecepatan dengan metode PID. Metode PID memperhitungkan besarnya galat, perubahan galat serta akumulasi galat sebagai masukan bagi kontroler untuk menentukan aksi kendali yang diperlukan. Sedangkan masukan bagi sistem kendali lingkaran tertutup adalah *setpoint* posisi dan kecepatan.

Kendali proporsional memberikan keluaran yang besarnya sebanding dengan galat. Pada sistem kendali, kendali proporsional ini berfungsi untuk memperkecil galat keadaan tunak dan mempercepat respon transien. Namun pengendali proporsional memiliki kelemahan karena tidak dapat menghilangkan galat keadaan tunak. Bila konstanta pengendali diset terlalu besar maka aksi kendali akan naik terlalu cepat, menyebabkan *overshoot* pada sistem.

Kendali integral berfungsi untuk menghilangkan galat keadaan tunak. Kendali integral mengambil akumulasi galat sebagai parameter untuk menghasilkan aksi kendali. Nilai aksi kendali dari kendali integral akan tetap ketika galat sudah berharga nol, sehingga menjaga keluaran sistem agar sesuai dengan *setpoint* yang diinginkan. Walaupun dapat menghilangkan galat keadaan tunak, kendali integral memperlambat respon transien dan menyebabkan sistem cenderung berosilasi.

Kendali derivatif memberikan aksi kendali berdasarkan kecepatan perubahan galat. Secara garis besar kendali derivatif meredam akselerasi keluaran sistem kendali pada saat keluaran sistem kendali tersebut sudah mendekati *setpoint*. Dengan demikian kendali derivatif dapat menghilangkan *overshoot* pada sistem dan membuat sistem lebih peka terhadap perubahan output.

Ketiga buah komponen pengendali ini memberikan kontribusi bagi performa sistem secara keseluruhan. Aksi kendali yang dihasilkan adalah gabungan dari aksi kendali dari masing-masing komponen pengendali di atas. Untuk mendapatkan sistem kendali yang dapat memenuhi spesifikasi yang diinginkan, harus ditentukan konstanta pengendali yang sesuai untuk masing-masing komponen proporsional, integral dan derivatif.

Dalam mencari konstanta pengendali dapat digunakan dua buah pendekatan, yaitu pendekatan secara analitis dan pendekatan secara eksperimental. Untuk mencari konstanta pengendali secara analitis diperlukan adanya model yang akurat dari plant yang dikendalikan (dalam hal ini robot mobil). Dalam Tugas Akhir ini akan digunakan pencarian konstanta pengendali secara eksperimental (trial and

error).

2.2 Motor Arus Searah (DC)

Motor DC adalah motor yang paling sering digunakan pada platform robot mobil. Motor DC merupakan motor yang tahan lama dan memiliki jangkauan daya, ketersediaan dan harga yang bervariasi. Selain itu motor DC juga memiliki daya poros (torsi) yang lebih besar dibandingkan jenis motor lainnya.

Stator motor DC yang berupa magnet permanen memberikan fluksi magnet yang konstan, sehingga tidak ada suplai daya listrik dari luar yang digunakan untuk menghasilkan fluks magnet. Fluks magnet stator bersifat tetap di setiap keadaan arus jangkar, sehingga kurva torsi-kecepatan motor magnet menjadi lebih linier. Motor DC yang digunakan dapat dilihat pada Gambar 3.

2.3 Antarmuka antara motor dengan pengendali (H-Bridge)

Pada sistem kendali lingkaran tertutup ini, diperlukan sebuah antar-muka yang dapat menjembatani perbedaan spesifikasi antara pengendali dengan aktuatur. Agar sinyal kendali yang dikeluarkan pengendali dapat diterjemahkan dan dipahami oleh aktuatur maka diperlukan suatu komponen yang berfungsi menyesuaikan sinyal kendali menjadi sinyal yang dimengerti oleh aktuatur.

Mikrokontroler sebagai pengendali mengeluarkan sinyal kendali berupa pulsa yang lebarnya dapat dimodulasi (Pulse Width Modulation-PWM). Lebarnya pulsa pada sinyal kendali merepresentasikan besarnya amplitudo sinyal kendali yang ingin dikirimkan ke aktuatur. Selain itu juga diinginkan antarmuka yang memungkinkan pengendali untuk mengatur arah putaran motor.



Gambar-3 Motor DC

Antarmuka yang digunakan untuk dapat meneruskan sinyal kendali ke motor pada Tugas Akhir ini adalah rangkaian H-Bridge. Rangkaian ini memungkinkan motor berputar ke dua arah dan juga memungkinkan untuk mengendalikan motor yang

memerlukan tegangan besar dengan sinyal dari mikrokontroler yang bertegangan rendah.

2.4 Pengendali

Mikrokontroler adalah sebuah sistem yang merupakan gabungan dari komponen-komponen digital seperti memori, akumulator, register yang diintegrasikan. Mikrokontroler mampu melakukan manipulasi informasi, mengatur urutan eksekusi intruksi, menerjemahkan intruksi yang diprogram, mengendalikan timing dan operasi pada jalur komunikasi (bus), menyimpan intruksi dan data pada memori yang dimilikinya dan juga berkomunikasi dengan lingkungan di luar sistemnya.

Sebuah mikrokontroler memiliki set instruksi tertentu yang mampu dijalankan. Set instruksi yang dimiliki oleh mikrokontroler jenis tertentu berbeda dengan set instruksi mikrokontroler lain dengan tipe yang berbeda. Untuk tiap instruksi sudah ditentukan bentuk instruksi (format) yang dapat dipahami oleh mikrokontroler, dengan demikian tugas yang diberikan ke mikrokontroler harus berupa urutan instruksi-instruksi yang terdapat pada set instruksinya.

Saat ini mikrokontroler sudah mengalami perkembangan yang sangat pesat. Sebelumnya komunikasi dengan lingkungan di luar sistem hanya berupa komunikasi masukan dan keluaran digital sehingga bila dihubungkan dengan komponen eksternal memerlukan antarmuka yang relatif rumit dan juga program khusus yang secara spesifik menangani proses antarmuka tersebut. Sekarang telah banyak mikrokontroler yang mampu menangani fungsi tambahan seperti komunikasi serial, pembangkitan sinyal PWM, penghitung pulsa secara otomatis bahkan hingga memanipulasi sinyal analog (komparator analog dan melakukan konversi sinyal analog ke digital).

3 PERANCANGAN DAN IMPLEMENTASI

3.1 Spesifikasi sistem kendali SyncBot

Spesifikasi sistem kendali dapat dilihat pada Tabel 1.

Tabel-1 Spesifikasi sistem kendali SyncBot

| | | |
|---|---------------------------------------|-----------|
| 1 | Kecepatan motor tanpa beban | 5500 rpm |
| 2 | Rasio roda gigi untuk pergerakan maju | 72 : 1 |
| 3 | Rasio roda gigi untuk membelok | 27 : 1 |
| 4 | Jumlah pulsa per putaran roda | 864 |
| 5 | Jumlah pulsa tiap 360 derajat | 324 |
| 6 | Diameter roda | 6 cm |
| 7 | Keliling roda | 18,857 cm |

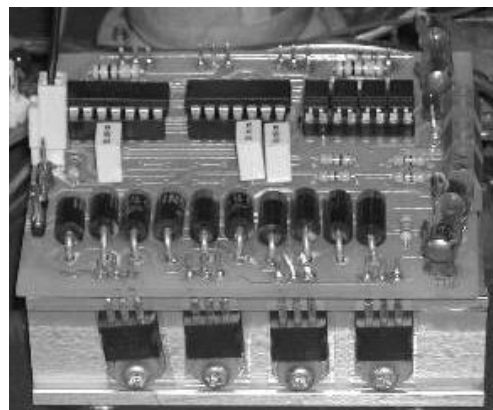
| | | |
|----|---------------------------------------|--------------|
| 8 | Kecepatan maksimum (secara teoritis) | 20.5 cm/s |
| 9 | Sampling rate PID | 20 Hz |
| 10 | Maksimum pulsa setiap <i>sampling</i> | 47 pulsa |
| 11 | Resolusi pengendalian arah robot | 1,11 derajat |

3.2 Perangkat Keras

Perangkat keras sebagai komponen penyusun sistem kendali pergerakan dirancang agar dapat memenuhi spesifikasi sistem kendali pergerakan SyncBot. Selain itu, perangkat keras yang dirancang juga memperhatikan aspek keamanan dan kemudahan dalam penggunaan. Gambar 4 dan Gambar 5 menunjukkan realisasi perangkat keras sistem kendali pergerakan.

Berikut ini spesifikasi perangkat keras yang dirancang sebagai bagian dari sistem kendali pergerakan SyncBot :

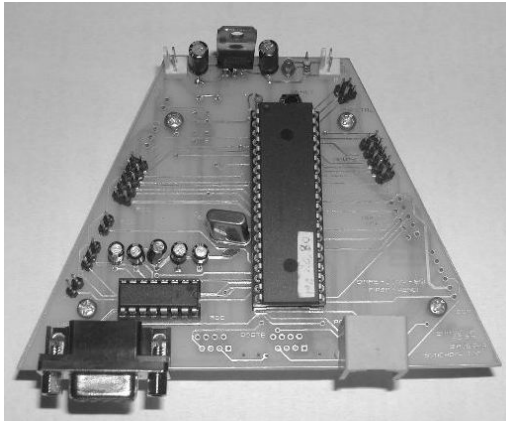
- Antarmuka pengendali dengan motor (*H-Bridge*):
 - Dapat digunakan untuk menggerakkan dua buah motor DC.
 - Menggunakan isolasi secara optik antara sisi tegangan tinggi dengan sisi tegangan rendah.
 - Sinyal masukan berupa arah dan keadaan motor (berputar atau berhenti).
 - Tegangan masukan pada sisi tegangan motor adalah 24 Volt.
 - Tegangan catu untuk rangkaian logika adalah 5 Volt.
 - Arus maksimum 3 Ampere.
 - Memiliki sekering pengamanan yang akan putus jika arus terlalu besar.
 - Memiliki rangkaian peredam yang berfungsi untuk menghilangkan tegangan balik akibat *switching* pada motor.



Gambar-4 Realisasi rangkaian antarmuka

- Pengendali :
 - Memiliki regulator tegangan dengan tegangan masukan 12 V.

- Memiliki tombol reset.
- Memiliki antarmuka untuk komunikasi serial dengan komputer.
- Menggunakan resonator kristal 20 MHz.
- Memiliki konektor output yang dikelompokkan berdasarkan fungsinya dengan tujuan memudahkan proses penghubungan pengendali ke komponen lain.

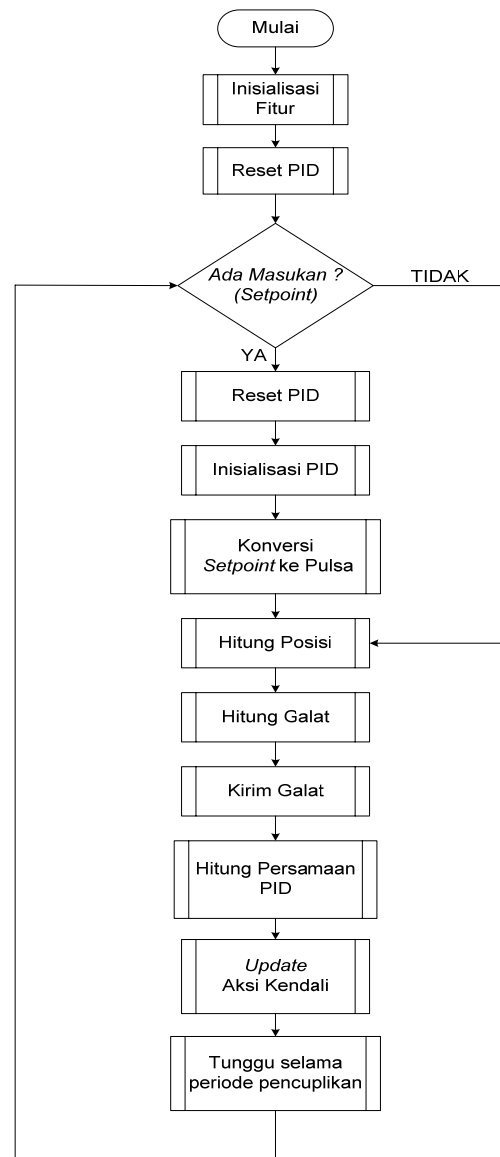


Gambar-5 Mikrokontroler PIC 16F877 sebagai pengendali

3.3 Perangkat Lunak

Perangkat lunak digunakan untuk melakukan perhitungan algoritma kendali PID. Proses pemanggilan fungsi-fungsi dapat dilihat pada Gambar 6. Perangkat lunak yang dirancang memiliki spesifikasi sebagai berikut :

- Melakukan perhitungan PID dalam *sampling rate* yang tetap.
- Menerima instruksi berupa *setpoint* dari komputer.
- Mengeluarkan sinyal galat ke komputer.
- Menerima konstanta PID yang digunakan dari komputer.
- Memberikan sinyal keluaran berupa sinyal PWM.
- Melakukan proses penghitungan pulsa sebagai umpan balik untuk algoritma kendali PID.



Gambar-6 Diagram alir perangkat lunak

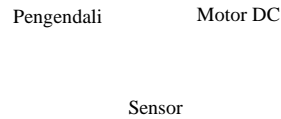
3.4 Strategi Kendali

Metode kendali PID yang diterapkan adalah kendali PID secara digital, dengan komponen mikrokontroler sebagai pengendalinya. Kendali PID secara digital memiliki beberapa perbedaan dengan metode kendali PID secara kontinu. Proses *sampling* dan kuantisasi pada penerapan kendali PID secara digital tidak ditemui pada kendali PID secara kontinu.

Pada dasarnya pengendali menentukan aksi kendali berdasarkan sinyal galat. Pada permulaan proses diberikan input *setpoint* kepada sistem, kemudian dihitung perbedaan antara *setpoint* yang diberikan dengan keluaran sistem sebenarnya. Besarnya keluaran sistem diperoleh dari sensor umpan balik. Skema dasar sistem kendali lingkaran tertutup dapat dilihat pada Gambar 7.

Proses penghitungan di atas dilakukan dalam

bentuk bilangan bulat. Strategi ini dimaksudkan untuk mengurangi waktu yang dibutuhkan oleh pengendali untuk komputasi sehingga dapat tercapai sampling rate yang lebih tinggi. Adanya proses pembulatan (kuantisasi) akan ikut mempengaruhi kinerja pengendali secara keseluruhan.



Gambar-7 Diagram blok sistem kendali

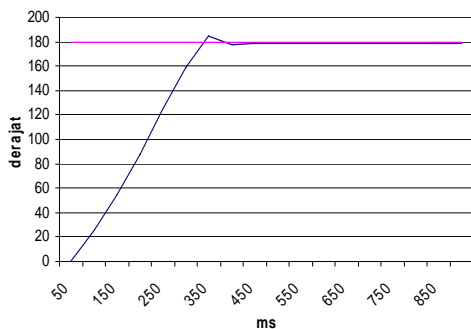
Dalam pemilihan frekuensi pencuplikan (sampling rate) juga diperhatikan kemampuan sensor umpan-balik dalam mengukur keadaan aktual dari keluaran sistem. Frekuensi pencuplikan yang terlalu tinggi menyebabkan sensor umpan balik belum mampu mengukur keadaan keluaran sistem dan pada akhirnya akan menimbulkan kesalahan operasi. Sebaliknya, periode pencuplikan yang terlalu besar juga tidak baik. Pada saat interval waktu antar pencuplikan, keadaan sistem dianggap tidak berubah oleh pengendali, walaupun pada kenyataannya perubahan keadaan tetap terjadi. Jika waktu antar pencuplikan terlalu besar maka pengendali akan terlambat dalam merespon perubahan yang terjadi pada sistem.

4 PENGUJIAN DAN ANALISA

4.1 Sistem Kendali Orientasi

Pengujian ini dimaksudkan untuk mengetahui performa dari sistem kendali lingkaran tertutup yang telah dirancang dan diimplementasikan. Sistem Kendali Orientasi

Setelah melakukan pengujian dengan nilai konstanta yang berbeda-beda, didapatkan nilai yang memberikan hasil sesuai spesifikasi adalah $KP = 1000$ dan $KD = 100$. Gambar 8 memperlihatkan respon waktu dari sistem kendali orientasi.



Gambar- 8 Respon waktu dengan $KP = 1000$ dan $KD = 100$

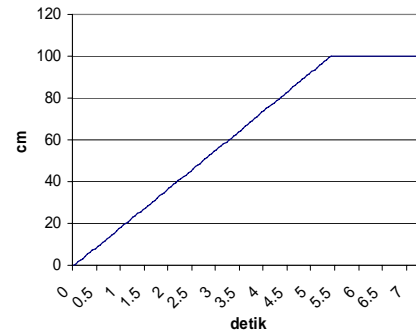
Hasil pengujian sudah memenuhi spesifikasi sistem kendali orientasi yang diinginkan. Galat keadaan tunak yang mendekati nol memungkinkan

robot mobil untuk bergerak ke arah yang dituju secara akurat. Respon transien yang cepat juga akan memudahkan pengendalian robot mobil oleh prosesor utama.

4.2 Sistem Kendali Posisi

Pada prinsipnya sistem kendali posisi SyncBot memiliki karakteristik yang sama dengan sistem kendali orientasi. Hal ini disebabkan karena proses pengendalian hanya melibatkan satu buah motor. Tetapi terdapat sedikit perbedaan yang terletak pada beban yang diterima oleh motor pada saat berputar. Secara keseluruhan metode yang dipakai pada bagian sebelumnya dapat juga diterapkan pada bagian ini.

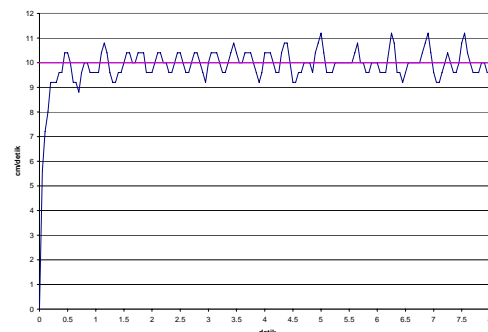
Setelah dilakukan pengujian untuk beberapa nilai konstanta pengendali proporsional ternyata respon waktu sistem kendali posisi sudah menunjukkan galat yang sangat kecil. Pada Gambar 9 diperoleh galat keadaan tunak sebesar 0,5 cm. Pengujian dilakukan dengan memerintahkan SyncBot bergerak sejauh satu meter.



Gambar- 9 Respon waktu kendali posisi dengan $KP = 5000$

4.3 Sistem Kendali Kecepatan

Pada sistem kendali kecepatan komponen pengendali yang digunakan adalah proporsional dan integrator. Komponen proporsional berfungsi untuk mempercepat respon transien sistem. Sedangkan komponen integrator memegang peranan pada saat sistem sudah mendekati *setpoint*. Gambar 10 memperlihatkan respon waktu sistem kendali kecepatan dengan nilai $KP = 500$ dan nilai $KI = 10000$ untuk *setpoint* sebesar 10 cm/detik dan 15 cm/detik.



Gambar-10 Respon waktu dengan $KP = 1000$ dan $KD = 100$

Dari grafik respon waktu yang diperoleh terlihat masih adanya riak kecepatan sekitar ± 1 cm/detik. Namun bila dilihat kecepatan rata-rata sistem pada kondisi tunak maka akan diperoleh kecepatan tempuh yang sesuai dengan *setpoint* yang diberikan. Hal ini juga telah dibuktikan dengan melakukan pengamatan secara langsung dengan cara mengukur jarak dan waktu tempuh dari robot mobil.

Dengan pertimbangan bahwa penambahan komponen derivatif akan memperkuat derau, maka komponen derivatif tidak digunakan dalam sistem kendali kecepatan.

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui proses perancangan, implementasi dan pengujian sistem pengendalian pergerakan lingkaran tertutup pada robot mobil SyncBot, dapat ditarik beberapa buah kesimpulan:

- Algoritma kendali PID dapat diimplementasikan secara digital menggunakan perangkat keras berupa mikrokontroler. Mikrokontroler berfungsi sebagai pengendali pada sistem pengendalian pergerakan lingkaran tertutup.
- Penerapan algoritma kendali PID secara perangkat lunak memerlukan beberapa modifikasi khusus
- Proses pencarian konstanta pengendali PID dapat dilakukan secara eksperimental berdasarkan grafik respon waktu dari sistem.
- Algoritma kendali PID telah berhasil diterapkan pada sistem kendali posisi, orientasi dan kecepatan pada robot mobil.

5.2 Saran

Sistem pengendalian pergerakan SyncBot masih memiliki beberapa batasan. Agar diperoleh sistem pengendalian pergerakan yang lebih baik, sebaiknya penelitian tentang sistem pengendalian ini dilanjutkan dengan memperhatikan beberapa hal, antara lain :

- Pengujian terhadap sistem pengendalian pergerakan lingkaran tertutup pada SyncBot baru dilakukan menggunakan sensor internal. Ada baiknya dilakukan juga pengujian sistem pengendalian pergerakan dengan menggunakan sensor eksternal.
- Sistem kendali lingkaran tertutup yang dirancang tidak mampu mengoreksi keadaan sistem jika terdapat gangguan dari luar sistem. Pada saat pengoperasian robot mobil gangguan dari luar sistem merupakan suatu hal yang sangat mungkin terjadi. Oleh karena itu perlu ditambahkan beberapa komponen pada sistem.
- Untuk mempermudah pengembangan, implementasi perangkat lunak dapat diperbaiki dengan menggunakan RTOS (*Real Time*

Operating System) sebagai kerangka dasar perangkat lunak. Dengan menggunakan RTOS pengembangan perangkat lunak dapat dilakukan hanya dengan menambahkan fungsi-fungsi yang diinginkan.

DAFTAR PUSTAKA

- [1] M. I. Ribeiro dan P. Lima, "Kinematics models of mobile robots" April 2002.
- [2] Dennis Clark, Michael Owings, *Robot DNA: Building Robot Drive Trains*, New York: McGraw-Hill, 2003
- [3] Myke Predko, *Programming Robot Controller*, New York: McGraw-Hill, 2003 C. C.

BIODATA

Muhammad Ilhamdi Rusydi ST, MT, lahir di Padang, 22 Mei 1982. Menyelesaikan kuliah sarjana (S1) dan master (S2) di Teknik Elektro ITB dengan spesifikasi pada bidang kendali.