

## PERANCANGAN SISTEM KONTROL GERAKAN PADA ROBOT *LINE TRACER*

Ahmad Sahru Romadhon<sup>1)</sup>, Muhammmad Fuad<sup>2)</sup>

Program Studi Mekatronika, Fakultas Teknik, Universitas Trujoyo Madura<sup>1,2)</sup>  
Jl. Raya Telang, PO.Box. 2 Kamal Bangkalan - Madura  
Email: s4hru\_0354@yahoo.com<sup>1)</sup>, ibrahim.fuad@gmail.com<sup>2)</sup>

### ABSTRAK

Perancangan gerakan pada robot *line tracer* merupakan suatu sistem yang dapat mengontrol gerakan robot sehingga robot dapat bergerak dengan baik. Untuk itu dibutuhkan kontrol PID yang dapat mengontrol robot *line tracer* dalam melakukan gerakan pada bidang datar berwarna putih dengan garis berwarna hitam dalam lingkungan statis, sehingga robot dapat mengikuti garis hitam tersebut dengan baik. Dari penelitian ini, diperoleh hasil pengujian dalam ruangan dengan cahaya yang redup robot dapat berjalan dengan baik pada lintasan yang lurus dan gagal pada lintasan lengkung, sedangkan pengujian dalam ruangan dengan cahaya yang terang robot dapat berjalan dengan baik pada lintasan yang lurus maupun pada lintasan lengkung.

**Kata Kunci:** Kontrol PID, Robot *Line Tracer*

### ABSTRACT

*Motion control in line tracer robot is a system that capabel to control robot movement in order to move well. To achieve such a nice movement, it needs PID control that can control line traca robot to make a movement in flat white testpad with black line in static environment, so robot can folliw the black line smoothly. In this research, experiments with robot run in room with low intensity light resulting a successful cruise in stright path but failed in curve path. Experiment in room with high intensity resulting a successful navigation both in stright and curve path.*

**Key Words:** Kontrol PID, Robot *Line Tracer*.

### 1. Pendahuluan

Perencanaan Gerakan adalah salah satu komponen kecerdasan dalam suatu Robot Mobil Otomatis (*Autonomous Mobile Robot, AMR*). Karakteristik sebuah robot mobil yang dilengkapi dengan suatu kecerdasan dapat dibagi ke dalam beberapa tingkatan struktur kontrol, yaitu *Planner-Navigator-Pilot-Controller* [1].

Kontroler sebagai tingkatan yang paling dekat dengan perangkat keras merupakan komponen penting yang menerjemahkan informasi trayektori sebagai hasil dari tingkatan di atasnya (*Planner-Navigator-Pilot*) ke dalam bentuk gerakan mekanis melalui kontrol putaran motor.

Salah satu mode kontrol klasik yang cukup banyak digunakan untuk melakukan kontrol pada motor adalah kontroler *Proportional-Integral-Derivative* (PID) [2]. Pengontrol PID (Proporsional-Integral-Derivatif) merupakan salah satu pengontrol yang dikenal luas dan telah terbukti di industri selama lebih dari tujuh dekade. Pengontrol PID ditemukan pada tahun 1922 dan sampai sekarang masih merupakan pengontrol yang paling banyak digunakan, lebih dari 80 % lup kontrol umpan balik di industri saat ini masih didominasi oleh pengontrol PID [3]. Daya tarik utama dari pengontrol PID adalah pengontrol ini mudah digunakan [4] dapat dioperasikan oleh operator yang tidak mempunyai pengetahuan tentang teori atau teknik kontrol otomatis. Operator cukup dibekali pengalaman terhadap karakteristik proses yang dikontrol, berdasarkan pengetahuan ini, operator dapat menentukan parameter-parameter pengontrol PID secara coba-coba dengan hasil yang memuaskan.

Telah banyak penelitian yang mengimplementasikan kontroler PID[5], termasuk didalamnya implementasi PID dalam robot mobil penjejak garis (*line*

*follower mobile robot*). Penelitian ini menggunakan kontroler PID untuk kontrol gerakan pada robot *line tracer* untuk mencapai suatu sasaran tertentu dengan mengikuti jalur yang telah tersedia.

Permasalahan yang akan diselesaikan dalam penelitian ini adalah bagaimana sebuah robot *line tracer* dapat melakukan kontrol gerakan pada bidang datar berwarna putih dengan garis berwarna hitam dalam lingkungan statis, sehingga robot dapat mengikuti garis hitam tersebut dengan baik.

### 2. Tinjauan Pustaka

Kontroler merupakan salah satu komponen sistem yang berfungsi mengolah sinyal umpan balik dan sinyal referensi menjadi sinyal kontrol sedemikian rupa sehingga performansi dari sistem yang dikendalikannya sesuai dengan spesifikasi performansi yang diinginkan.

#### Kontroler PID

Merupakan kontroler *feed-forward* yang berfungsi mengolah sinyal *error* menjadi sinyal kontrol, di mana hubungan sinyal kontrol terhadap sinyal *error* dapat proporsional, integral, diferensial atau gabungan diantaranya. Berdasarkan bentuk hubungan sinyal *error* dan sinyal kontrol, kontroler ini dibedakan atas beberapa tipe, antara lain:

#### A. Kontroler tipe-P (*Proportional Controller*)

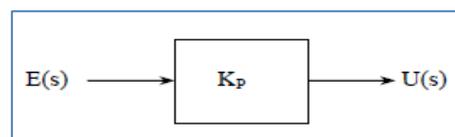
Hubungan sinyal eror dan sinyal kontrol pada kontroler tipe-P dapat dinyatakan pada Persamaan (1):

$$u(t) = K_p e(t) \quad (1)$$

atau dalam bentuk *transfer function* Persamaan (2).

$$\frac{U_s}{U(s)} = K_p \quad (2)$$

Diagram balok Gambar 1 adalah kontroler untuk tipe P.



Gambar 1. Diagram Balok Kontroler tipe-P

**B. Kontroler tipe-PI (Proportional + Integral Controller)**

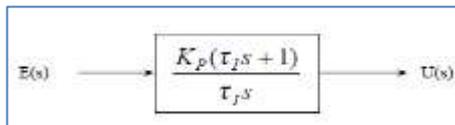
Hubungan sinyal error dan sinyal kontrol pada kontroler tipe-PI dapat dinyatakan pada rumus (3).

$$u(t) = K_p \left[ e(t) + \frac{1}{\tau_i} \int e(t) dt \right] \quad (3)$$

atau dalam bentuk *transfer function*

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{\tau_i s} \right) \text{ atau } \frac{U(s)}{E(s)} = \frac{K_p(\tau_i s + 1)}{\tau_i s} \quad (4)$$

Dalam bentuk diagram balok, kontroler tipe-PI ini ditunjukkan pada Gambar 2.



Gambar 2. Diagram balok kontroler tipe-PI

**C. Kontroler tipe-PD (Proportional + Derivative Controller)**

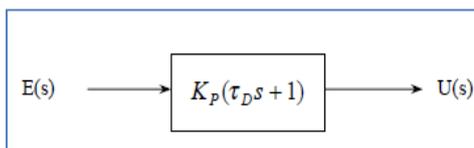
Hubungan sinyal *error* dan sinyal kontrol pada kontroler tipe-PD dapat dinyatakan seperti pada rumus (5).

$$u(t) = K_p \left[ e(t) + \tau_D \frac{d}{dt} e(t) \right] \quad (5)$$

atau dalam bentuk *transfer function*

$$\frac{U(s)}{E(s)} = K_p \left( \tau_D \frac{d}{dt} e(t) \right) \quad (6)$$

Dalam bentuk diagram balok, kontroler tipe-PD ini digambarkan sebagai:



Gambar 3. Diagram Balok Kontroler tipe-PD

**D. Kontroler tipe-PID (Proportional + Integral + Derivative Controller)**

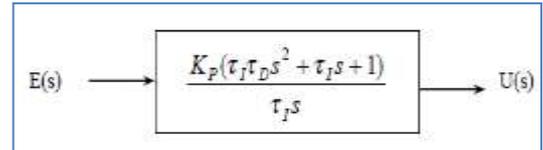
Hubungan sinyal error dan sinyal kontrol pada kontroler tipe-PID standar dapat dinyatakan seperti rumus (7).

$$u(t) = K_p \left[ e(t) + \frac{1}{\tau_i} \int e(t) dt + \tau_D \frac{d}{dt} e(t) \right] \quad (7)$$

atau dalam bentuk *transfer function* pada rumus (8).

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_D s \right) \text{ atau } \frac{U(s)}{E(s)} = \frac{K_p(\tau_i \tau_D s^2 + \tau_i s + \tau_D s)}{\tau_i s} \quad (8)$$

Dalam bentuk diagram balok, kontroler tipe-PID ditunjukkan pada Gambar 4.

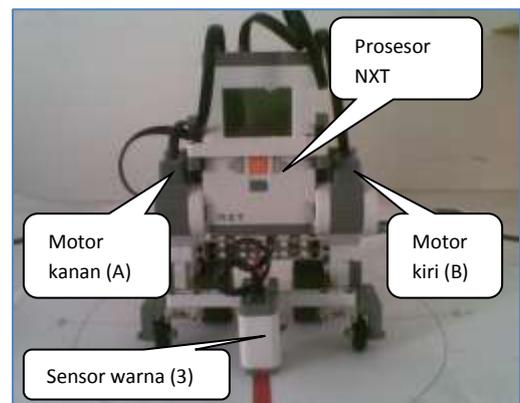


Gambar 4. Diagram Balok Kontroler Standar tipe-PID

**3. Metodologi Penelitian**

Sistem kontrol gerakan robot mobil diimplementasikan ke dalam *plant* berupa robot LEGO Mindstorms NXT (Gambar 1) dengan spesifikasi sebagai berikut:

1. Bricx
  - Pusat pengolahan data yang terdiri dari tiga *output port* A, B, C; empat *input port* 1, 2, 3, 4; sebuah port USB untuk akses data dengan PC; layar LCD.
2. Motor DC
  - Robot ini menggunakan dua motor, yaitu: motor kanan sebagai *output* dihubungkan dengan *port* A dan motor kiri sebagai *output* dihubungkan dengan *port* B. Kedua motor ini dilengkapi dengan roda jenis 43,2 X 22 ZR.
3. Sensor
  - Robot hanya menggunakan sebuah sensor warna yang mampu mengenali 6 warna, yaitu: hitam, biru, hijau, kuning, merah dan putih. Sensor warna sebagai input dihubungkan dengan *port* 3.



Gambar 5. Konstruksi Robot Mobil Penjejak Garis

Algoritma kontroler PID untuk penjejak garis adalah sebagai berikut:

1. Menentukan konstanta proporsional, integral, derivative, waktu sampling dan kecepatan
2. Membaca sensor warna untuk inialisasi *setpoint*
3. Membaca sensor warna untuk menentukan posisi aktual
4. Menghitung selisih antara nilai *setpoint* dan nilai posisi aktual hingga menghasilkan nilai *error*
5. Mengalikan nilai *error* dengan konstanta proporsional dan menyimpannya dalam variabel proporsional
6. Menjumlahkan nilai *error* ke dalam variabel integral
7. Menghitung selisih antara nilai *error* aktual dan nilai *error* sebelumnya kemudian membaginya dengan waktu sampling dan menyimpan nilai ini ke dalam variabel derivative
8. Menghitung jumlah dari nilai variabel proporsional, integral dan derivative
9. Menyimpan nilai *error* aktual sebagai *error* sebelumnya sehingga variabel *error* aktual dapat digunakan untuk menyimpan nilai *error* aktual yang berikutnya
10. Mengatur kecepatan dan orientasi motor kanan dan kiri berdasarkan konstanta kecepatan dan jumlah dari nilai PID
11. Mengulang langkah ke-3 hingga ke-11 sampai batas tertentu

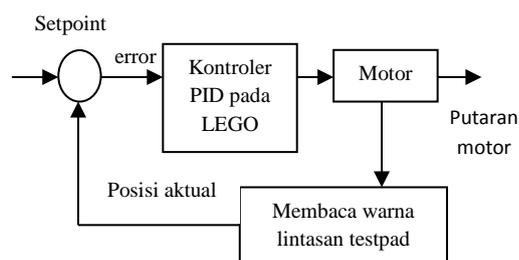
Bidang uji (*testpad*) berupa bidang dua dimensi berisi kumpulan rute berupa garis warna hitam yang dijejak robot *line tracer* yaitu *testpad* 8547 dari LEGO (Gambar 6). *Testpad* 8547 digunakan untuk menguji implementasi algoritma kontroler PID untuk gerakan robot mobil menjejak garis.



Gambar 6. LEGO Mindstorms Testpad 8547

Diagram blok pada Gambar 7 menggambarkan sub sistem robot mobil penjejak garis yang terdiri dari beberapa blok.

1. *Setpoint* menentukan nilai referensi berdasarkan warna yang dijejak pertama kali, dalam studi kasus ini warna hitam sebagai referensi. Nilai ini didapatkan dari sensor warna yang ditempatkan di atas garis hitam pada *testpad*.
2. Posisi aktual berisi pembacaan terhadap warna lintasan yang terbaca sensor warna pada pengambilan data warna yang berikutnya. Nilai posisi aktual didapatkan saat sensor menangkap warna seiring gerakan robot mobil.
3. *Error* didapatkan dari membandingkan *setpoint* dan posisi aktual. Selisih antara *setpoint* dan posisi aktual menghasilkan nilai *error* sebagai masukan bagi sub sistem kontroler PID.
4. Sub sistem kontroler PID mengolah nilai *error* menghasilkan sinyal kontrol yang diumpankan ke motor kanan dan kiri.
5. Putaran motor menggerakkan robot mobil menyusuri lintasan yang warnanya dibaca kembali untuk meng-*update* nilai posisi aktual.



Gambar 7. Diagram blok sistem kontrol penjejak garis

#### 4. Hasil dan Pembahasan

Algoritma kontroler PID pada robot *line tracer* diimplementasikan dengan menggunakan bahasa NXC (*Not eXactly C*) yang dikembangkan dengan editor *BricxCC* (*Bricx Command Center*) [6].

Implementasi algoritma kontroler PID untuk robot mobil *line tracer*, terdiri dari beberapa sub sistem.

1. Definisi konstanta
2. Pembacaan sensor warna sebagai *setpoint*
3. Pembacaan sensor warna sebagai posisi aktual
4. Menghitung *error* dari *setpoint* dan *actual\_position*
5. Perhitungan nilai proporsional, integral, derivative (PID). Selanjutnya *output* dari proses ini menjadi sinyal kontrol untuk motor
6. Sinyal kontrol untuk motor didapatkan dari kombinasi antara variabel *speed* dan *output* PID

### 7. Pengujian

Pengujian pada robot mobil *line tracer* dibagi dalam dua studi kasus. Studi kasus pertama menguji pada ruangan dengan pencahayaan yang redup dan studi kasus kedua menguji algoritma kontroler PID untuk penjejak garis pada ruangan dengan pencahayaan yang terang.

Pada waktu awal, robot mobil diletakkan pada lintasan dengan garis hitam untuk pembacaan *setpoint* dan selanjutnya membandingkan nilai *setpoint* ini dengan nilai posisi aktual hasil pembacaan sensor pada waktu yang berikutnya.

#### A. Studi Kasus 1

Pengujian pertama kontroler PID untuk robot mobil penjejak garis menggunakan *testpad* LEGO Midstorms NXT 8547. *Testpad* diletakkan pada bidang datar dalam ruangan dengan pencahayaan yang redup. Hasil dari percobaan dengan kondisi ini adalah robot mobil mampu menyusuri lintasan garis hitam dengan baik namun gagal pada lintasan lengkung.

#### B. Studi Kasus 2

Pengujian kedua dilakukan pada kondisi ruangan dengan cahaya yang terang. Hasil dari pengujian kedua adalah robot mobil mampu menjejak garis hitam pada lintasan lurus maupun lengkung dengan baik seperti ditunjukkan pada Gambar 8.



Gambar 8. Hasil Pengujian Studi Kasus 2

### 8. Kesimpulan

1. Kontrol PID adalah kontrol yang memanfaatkan umpan balik untuk memperbaiki kinerja sistem atau sebuah proses.
2. Kontrol PID menghitung nilai kesalahan (*error*) antara nilai acuan (*setpoint*) dan nilai keluaran (*output*) yang terukur oleh sensor.
3. Kontrol PID meminimalisir nilai kesalahan (*error*) dengan cara mengatur sinyal kontrol masukan (*input*) yang diberikan ke dalam sistem.
4. Pada pengujian dalam ruangan dengan cahaya yang redup robot dapat berjalan dengan baik pada lintasan yang lurus dan gagal pada lintasan lengkung.
5. Pada pengujian dalam ruangan dengan cahaya yang terang robot dapat berjalan dengan baik pada lintasan yang lurus maupun pada lintasan lengkung.

**Daftar Pustaka**

- [1] Meystel A. [1991]. *Autonomous Mobile Robot Vehicles with Cognitive Control*, Singapore: World Scientific Publishing.
- [2] Johnson, C.[1993], *Process Control Instrumentation Technology, Chapter 9 Controller Principles*, halaman 386, New Jersey: Prentice Hall.
- [3] Takatsu, H., Itoh, T. & Araki, M.[1998], *Future needs for the control theory in industries: Report and Topics of the Control Technology Survey in Japanese Industry*, J. Process Control, 8(5-6), 363-374.
- [4] M. M'Saad, M. Bouslimani, and M. A. Latifi. [1993], "PID Adaptive Control Of Exothermic Stirred Tank Reactors", *Second IEEE Conference on Control Applications*, Vol.1, pp. 113 – 117, September.
- [5] B. Kumar, and R. Dhiman. [2011], "Optimization Of Pid Controller For Liquid Level Tank System Using Intelligent Echniques", *Canadian Journal on Electrical and Electronics Engineering*, Vol. 2, No. 11, pp. 531-535, November.
- [6] Hansen, J. [2010], *NXC Programmer's Guide Version 1.2.1 r3*, June 8.