

## MODUL 5 Sistem Kendali PID : Kasus P dan D

### 1. JUDUL PRAKTIKUM

Sistem Kendali PID Kasus P dan D

### 2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PID pada motor DC
2. Mahasiswa dapat membuat program sistem kendali berbasis PID dengan error yang dihubungkan dengan konstanta proporsional dan derivatif

### 3. PARAMETER PENILAIAN

No.	Parameter	Persentase (%)
1.	Lembar Penilaian Praktikum	40%
2.	Jurnal/Laporan Praktikum	60%

### 4. PERALATAN DAN BAHAN

Alat dan Bahan :

1. Robot Kit Line Follower
2. Baterai LiPo 2-Cell 1300 mAh
3. Kabel Mini-USB
4. Arduino Nano
5. Battery Checker
6. Battery Balancer

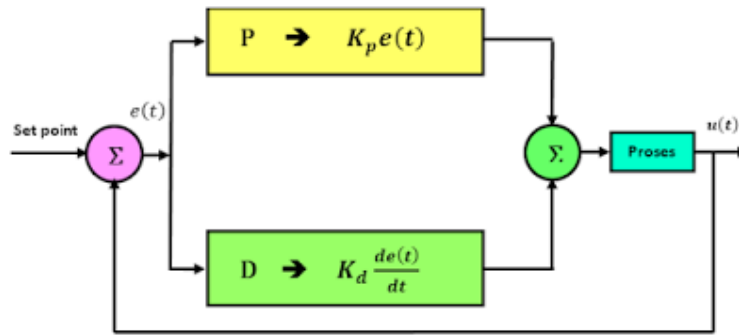
Perangkat Lunak :

1. Software IDE Arduino
2. Software Proteus (untuk simulasi)

### 5. TEORI DASAR

#### 5.1. Sistem Kendali PD

Teknik kendali proporsional-derivatif (PD) adalah pengendali yang merupakan gabungan antara teknik kendali proporsional (P) dengan teknik kendali derivatif (D). Gambar 1 merupakan gambar diagram blok sistem kendali PD.



Gambar 1 Diagram blok sistem kendali PD.

Persamaan hubungan antara keluaran sistem dengan sinyal error pada kombinasi aksi kendali proporsional-derivative adalah sebagai berikut.

$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \quad \dots\dots (1)$$

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad \dots\dots(2)$$

Dalam penerapannya di software, kondisi ideal pada robot adalah bergerak maju lurus mengikuti garis, dengan kata lain error = 0 . Dari sini dapat diasumsikan bahwa Set Point (SP) / kondisi ideal adalah saat SP = 0. Nilai sensor yang dibaca oleh sensor disebut *Process Variable* (PV) / nilai aktual pembacaan. Menyimpangnya posisi robot dari garis disebut sebagai *error* (e), yang didapat dari  $e = SP - PV$ . Dengan mengetahui besar *error*, mikrokontroler dapat memberikan nilai PWM motor kiri dan kanan yang sesuai agar dapat menuju ke posisi ideal (SP = 0). Besarnya nilai PWM ini dapat diperoleh dengan menggunakan kontrol Proporsional (P), dimana  $P = e \times K_p$  ( $K_p$  adalah konstanta proporsional yang nilainya diset sendiri dari hasil *tuning/trial and error*).

Jika pergerakan robot masih terlihat bergelombang, dapat ditambahkan parameter kontrol Derivatif (D). Kontrol D digunakan untuk mengukur seberapa cepat robot bergerak dari kiri ke kanan atau dari kanan ke kiri. Semakin cepat bergerak dari satu sisi ke sisi lainnya, maka semakin besar nilai D. Konstanta D ( $K_d$ ) digunakan untuk menambah atau mengurangi imbas dari derivatif. Dengan mendapatkan nilai  $K_d$  yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional dapat diminimalisasi. Dengan mendapatkan nilai  $K_d$  yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional bisa diminimalisasi. Nilai D didapat dari  $D = K_d/T_s \times rate$ , dimana  $T_s$  adalah time sampling atau waktu cuplik dan  $rate = e(n) - e(n-1)$ . Dalam program, nilai error (SP – PV) saat itu menjadi nilai `last_error`, sehingga `rate` didapat dari `error – last_error`

Ringkasan efek perubahan dari masing-masing parameter PID terhadap keluaran dari robot *line follower* dapat dilihat pada tabel berikut.

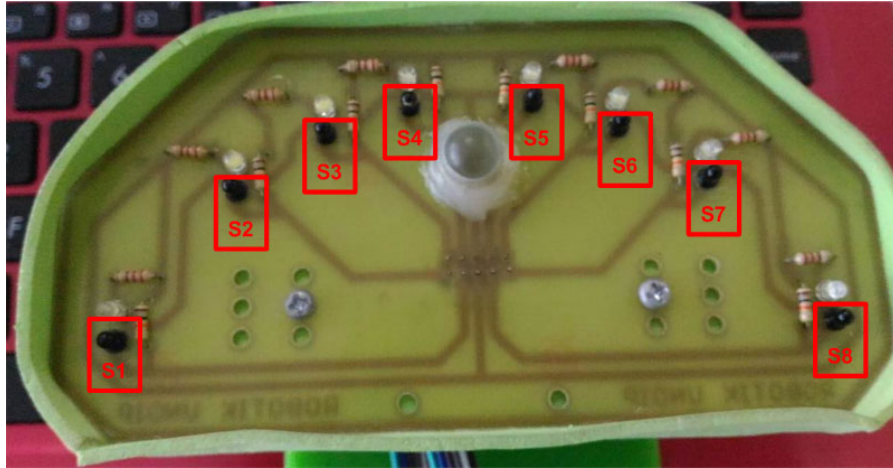
*Tabel 1 Efek Perubahan Konstanta terhadap Parameter PID terhadap Keluaran Sistem*

<b>Parameter</b>	<b><i>Rise Time</i></b>	<b><i>Overshoot</i></b>	<b><i>Settling Time</i></b>	<b><i>Steady State Error</i></b>
$K_p$	Menurun	Meningkat	Perubahan kecil	Menurun
$K_i$	Menurun	Meningkat	Meningkat	Menurun signifikan
$K_d$	Sedikit turun	Sedikit turun	Sedikit turun	Tidak ada efek

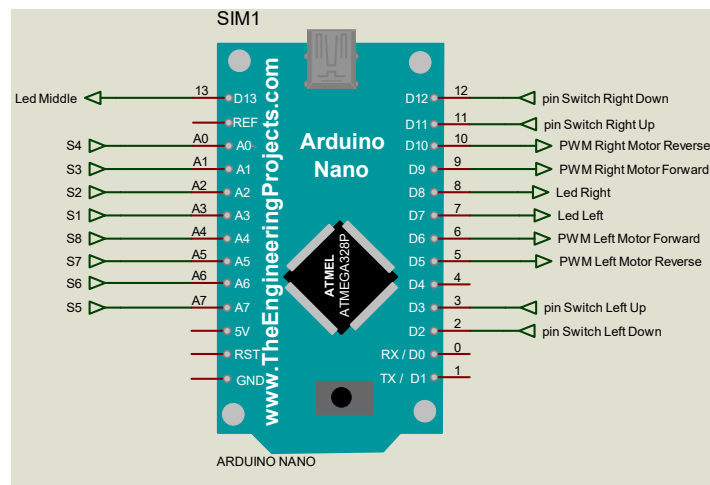
## PROSEDUR PRAKTIKUM

### A. Percobaan dalam praktikum

#### 1. Kasus Percobaan



Gambar 2 Contoh susunan dan urutan sensor pada robot line follower.



Gambar 3 Pin Layout Arduino pada Robot Line Follower.

- a. Modifikasi program sistem kendali pada praktikum sebelumnya dengan menambahkan sebuah sub program untuk melakukan kalibrasi sensor secara otomatis (*auto calibration*). Kalibrasi ini berfungsi untuk menentukan nilai pembacaan sensor untuk warna hitam dan putih atau warna lainnya. Apabila push button yang terhubung dengan pin D2 (pin switch Left Down (Kiri bawah) pada Gambar 2) ditekan, eksekusi program berikut :
- Tambahkan variabel **sensor[0]** sampai **sensor[7]**
  - Tambahkan variabel **NilaiMaxSensor[0]** sampai **NilaiMaxSensor[7]** dengan nilai awal 1023
  - Tambahkan variabel **NilaiMinSensor[0]** sampai **NilaiMinSensor[7]** dengan nilai awal 0
  - Tambahkan variabel **NilaiTengahSensor[0]** sampai **NilaiTengahSensor[7]**

- Baca kondisi sensor 1 sampai 8 dan simpan di variabel **sensor[0]** sampai **sensor[7]**. Untuk nilai *i* dari 0 hingga 7 :
  - o Jika nilai variabel **sensor[i] > NilaiMinSensor[i]** maka **NilaiMinSensor[i]=sensor[i]**
  - o Jika nilai variabel **sensor[i] < NilaiMaxSensor[i]** maka **NilaiMaxSensor[i]=sensor[i]**
  - o **NilaiTengahSensor[i]** =  

$$(\text{NilaiMinSensor[i]} + \text{NilaiMaxSensor[i]}) / 2$$
  - o Saat proses kalibrasi dilakukan, LED LEFT (Pin D7) dan LED RIGHT (Pin D8) akan berkedap-kedip (*blinking*) sebagai indikator bahwa proses kalibrasi sedang berjalan.
  
- b. Apabila proses kalibrasi sudah selesai dan tombol push button yang terhubung dengan pin D3 (pin switch Left Up/kiri atas pada Gambar 2) ditekan, maka eksekusi program berikut ini :
  - Inisialisasi variabel dengan tipe **int Kp** dengan nilai awal 15, **int Kd** dengan nilai awal : 5, **int moveControl** dengan nilai awal 0, **int error** dengan nilai awal 0, **int lastError** dengan nilai awal 0. **int kecepatanMotorKanan** dengan nilai awal 0, **int kecepatanMotorKiri** dengan nilai awal 0, **int kecepatanSetPoint** dengan nilai awal 150 dan **int rate** (selisih **error** dengan **lastError**).
  - Program harus dapat mendeteksi perubahan nilai pada sensor dan mengirimkannya ke serial monitor dengan ketentuan sebagai berikut.
    - Jika kondisi sensor "10000000", error = -7, print di serial monitor error = -7,
    - Jika kondisi sensor "11000000", error = -6, print di serial monitor error = -6,
    - Jika kondisi sensor "01000000", error = -5, print di serial monitor error = -5,
    - Jika kondisi sensor "01100000", error = -4, print di serial monitor error = -4,
    - Jika kondisi sensor "00100000", error = -3, print di serial monitor error = -3,
    - Jika kondisi sensor "00110000", error = -2, print di serial monitor error = -2,
    - Jika kondisi sensor "00010000", error = -1, print di serial monitor error = -1,
    - Jika kondisi sensor "00011000", error = 0, print di serial monitor error = 0,
    - Jika kondisi sensor "00001000", error = 1, print di serial monitor error = 1,

- Jika kondisi sensor "00001100", error = 2, print di serial monitor error = 2,
- Jika kondisi sensor "00000100", error = 3, print di serial monitor error = 3,
- Jika kondisi sensor "00000110", error = 4, print di serial monitor error = 4,
- Jika kondisi sensor "00000010", error = 5, print di serial monitor error = 5,
- Jika kondisi sensor "00000011", error = 6, print di serial monitor error = 6,
- Jika kondisi sensor "00000001", error = 7, print di serial monitor error = 7,

Jelaskan fungsi dari variabel yang telah ditambahkan pada program di atas terhadap mekanisme sistem kendali pada robot *line follower*!

- c. Kemudian tambahkan kode program dengan ketentuan sebagai berikut :
- 1 Simpan nilai error saat ini dengan variabel **Error** dan nilai error sebelumnya dengan variable **lastError**.
  - 2 Simpan nilai selisih antara **lastError** dengan **Error** dalam variabel **rate**.
  - 3 Simpan nilai **moveControl** = perkalian **Kp** dan **Error** ditambah hasil perkalian **Kd** dan **rate**.
  - 4 Simpan nilai **kecepatanMotorKanan** = **kecepatanSetPoint** dikurang **moveControl**
  - 5 Simpan nilai **kecepatanMotorKiri** = **kecepatanSetPoint** ditambah **moveControl**
  - 6 Kecepatan Motor Kiri dengan nilai analog sebesar **kecepatanMotorKiri**
  - 7 Kecepatan Motor Kanan dengan nilai analog sebesar **kecepatanMotorKanan**

Variasikan nilai konstanta Kp dan Kd pada program kemudian ujicoba dan amati hasil perubahan kedua konstanta tersebut!

Jelaskan fungsi dari **Kp** dan **Kd** pada kode program di atas!

- d. *Screenshoot* keluaran serial monitor untuk setiap kondisi. Cetak dan tempelkan pada buku jurnal praktikum.
- e. Isi tabel kebenaran dari sistem pada Tabel 2 dan tuliskan pada buku jurnal praktikum.

Tabel 2 Tabel Kebenaran Sistem Kendali PD

[illegible]

## 6. Jurnal Praktikum

- a. Jurnal pada Buku Praktikum harus memuat konten sebagai berikut :
- Judul Praktikum :
  - Maksud dan Tujuan Praktikum :
  - Peralatan dan Bahan Praktikum :
  - Dasar Teori
  - Foto Peralatan dan Bahan Praktikum :
  - Hasil Praktikum (Tulis tangan kode program yang telah diberi komentar/penjelasan beserta foto hasil percobaan yang telah diberi nama dan NIM anggota kelompok)
  - Kesimpulan Praktikum