

## ABSTRACT

The world of robotics is growing rapidly. Especially on the surveillance robot. The type of robot used for surveillance robot is tank robot because this robot have the ability to pass through terrain that is quite heavy. However, most people only use the motor torque only for its strenght regardless of other factors. Therefore, I try ro designed a tank robot equipped with PID control. PID control is expected ro provide a stable speed rotation of the dc motor on the robot, so the robot can work optimally.

The purpose of this design is how to designed a PID system for the robot, so the robot can move at a steady pace as it passes through the uphill road, flat, and downhill road. In addition, mechanical and electronics systems on the robot is also very concerned to support the performance of the robot.

PID control is embedded in LabView. The result of the calculation is sent to the microcontroller ATmega 328P wirelessly. This value is the PWM value that useful to drive the dc motor. Based on open loop experiment, the readings from sensor is still not good. There are some errors between 1 RPM until 5 RPM. However, the result of experiment in close loop system, on the up and down terrain, from experiments on seven angle ( $5^{\circ}$ - $35^{\circ}$ ), only six angle can be managed. One angle ( $35^{\circ}$ ), the robot was fail to pass the up and down terrain.

Keywords : PID, robot tank, ATmega 328P, LabView

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini, perkembangan dalam dunia robotika semakin pesat, terutama robot *surveillance* yang berguna untuk membantu manusia dalam menanggulangi bencana alam. Robot *surveillance* merupakan robot yang memiliki kemampuan tinggi yang biasa digunakan untuk misi-misi penyelamatan dengan medan yang cukup berat. Robot *surveillance* ini biasanya menggunakan robot yang berbentuk seperti tank. Robot seperti ini memiliki keunggulan dalam melewati medan yang cukup berat karena memiliki bentuk roda yang berkemampuan untuk mencengkram tanah.

Robot yang berjenis robot tank harus memiliki motor dengan torsi yang besar agar dapat melewati segala medan. Torsi motor ini berpengaruh dalam kekuatan robot untuk menggerakkan robot. Namun, torsi motor yang tinggi saja tidak cukup karena rotasi motor tersebut bisa melambat saat melewati tanjakan dengan lintasan yang cukup panjang akibat tarikan gaya gravitasi bumi.

Oleh karena itu, dirancanglah suatu sistem kontrol kecepatan motor dc pada robot darat yang berjenis robot tank dengan metode PID. Maksud perancangan tersebut tidak lain untuk menyempurnakan pergerakan dari robot *surveillance* agar robot tersebut mampu melewati segala jalur, seperti tanjakan, mendatar, dan turunan, dan dengan perubahan bobot robot dengan kecepatan yang lebih stabil.

Perancangan sistem PID dieksekusi didalam program yang dibuat pada LabVIEW. Hal ini dimaksudkan untuk memberi keringanan kepada mikrokontroler pada robot agar kinerjanya lebih maksimal lagi karena segala macam perhitungan untuk PID dilakukan pada LabVIEW, sehingga mikrokontroler pada robot hanya perlu untuk menggerakkan motor dan membaca sensor saja. Selain itu, apabila mikrokontroler pada robot perlu diberi tambahan program, memori pada mikrokontroler masih tersedia.

## 1.2 Rumusan Masalah

Pada tugas akhir ini terdapat beberapa permasalahan yang muncul, diantaranya :

- a. Bagaimana merancang suatu sistem mekanik dan elektronik pada robot?
- b. Bagaimana merancang perangkat lunak untuk sebagai antar muka pengguna dalam monitoring robot dan sebagai pengontrol PID untuk robot?
- c. Bagaimana mengontrol kecepatan motor yang digunakan pada robot?

## 1.3 Batasan Masalah

Batasan-batasan masalah dalam tugas akhir ini diantaranya:

- a. Menggunakan robot darat berbentuk *mobile robot*.
- b. Sudut tanjakan dan turunan lintasan maksimum yang digunakan adalah  $35^\circ$ .
- c. Menggunakan metode PID dalam mengontrol kecepatan motor dc.
- d. Alat yang dirancang didesain untuk medan darat yang memiliki jalur yang menanjak, mendatar, dan menurun.
- e. Menggunakan perangkat lunak LabView sebagai pengendali gerak robot, pengontrol PID, dan antar muka dalam monitoring kecepatan rotasi pada motor.
- f. Menggunakan RF Modul sebagai alat komunikasi serial secara *wireless*.
- g. Robot hanya dikontrol untuk gerakan maju dan berhenti.
- h. Lintasan menggunakan medan kayu yang sudah diplitur.
- i. Menggunakan metode *trial and error* untuk menentukan nilai  $k_p$ ,  $k_i$ , dan  $k_d$ .
- j. Tidak ada sistem pengereman pada robot yang dirancang.
- k. Tidak ada sistem peringatan saat baterai sudah mulai lemah.

## 1.4 Tujuan Tugas Akhir

Berdasarkan pada rumusan masalah yang ada, maka dirumuskan tujuan dari tugas akhir ini adalah sebagai berikut:

- a. Merancang sistem mekanik dan elektronik pada robot.
- b. Merancang perangkat lunak sebagai antar muka pengguna dalam memonitoring robot dan sebagai pengontrol PID untuk robot?
- c. Mengontrol kecepatan motor yang digunakan pada robot.

## 1.5 Metoda Penelitian

Langkah yang akan ditempuh dalam menyelesaikan tugas akhir ini diantaranya;

- a. Tahap Studi Literatur  
Pada tahap ini dilakukan pencarian dan pengumpulan literatur-literatur dan kajian-kajian yang berkaitan dengan masalah-masalah yang ada pada tugas akhir ini, baik berupa artikel, buku referensi, jurnal, internet, dan sumber-sumber lain yang terpercaya yang berhubungan dengan masalah tugas akhir.
- b. Tahap Perancangan dan Realisasi Alat  
Pada tahap ini setelah mempelajari literatur yang ada, dilakukan pembuatan rancangan-rancangan yang kemudian direalisasikan rancangan tersebut ke dalam suatu rangkaian dan perangkat.
- c. Tahap Pengujian Sistem  
Berdasarkan standar yang ada, tahap selanjutnya adalah melakukan pengujian sistem untuk melihat kinerja sistem tersebut.

## 1.6 Sistematika Penulisan

Adapun sistematika penulisan pada tugas akhir ini, yaitu:

### **BAB I            PENDAHULUAN**

Pada bab ini menjelaskan latar belakang, rumusan masalah, tujuan masalah, batasan masalah, metoda penelitian, dan sistematika penulisan.

### **BAB II           DASAR TEORI**

Pada bab ini membahas mengenai dasar teori yang berkaitan dengan tugas akhir berupa kontrol PID, *Driver motor H-*

Bridge, mikrokontroler ATmega 328P, robot tank, motor dc, *line tracking sensor*, dan LabView 2014.

### **BAB III PERANCANGAN DAN IMPLEMENTASI**

Pada bab ini membahas model sistem kontrol gerak yang dibuat dan cara kerja sistem serta bentuk realisasinya. Selain itu membahas tentang perangkat lunak yang digunakan pada LabVIEW sebagai antar muka dan pusat pengendali kecepatan.

### **BAB IV PENGUJIAN DAN ANALISA**

Pada bab ini berisi pengujian perangkat lunak yang dibuat pada LabVIEW dan pengujian alat berupa kontrol gerak dan analisa sistem.

### **BAB V PENUTUP**

Bab ini berisi simpulan atas hasil kerja yang telah dilakukan beserta rekomendasi dan saran untuk pengembangan dan perbaikan selanjutnya.

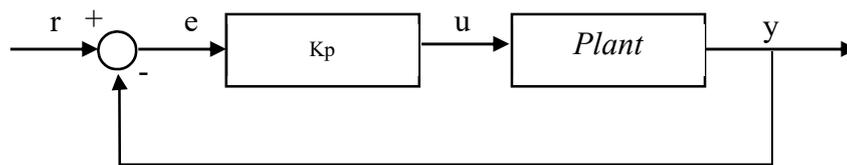
## BAB II DASAR TEORI

### 2.1 Kontrol PID

Kontrol PID merupakan suatu tipe kontrol yang paling klasik. Kontrol PID terdiri atas kontrol *proportional*, *integral*, dan *derivative* dengan masing-masing kontrol memiliki kelebihan dan kekurangan yang berbeda. Kontrol PID ini berpengaruh untuk menutupi semua kekurangan yang dimiliki oleh masing-masing kontrol. Dengan men-*tuning* salah satu nilai dari tiga pengendali, salah satu pengendali akan lebih menonjol dan akan berpengaruh pada pencapaian respon sistem yang diinginkan dengan *error* yang sangat minim.

Kontrol PID hanya dapat bekerja pada sistem yang cenderung linier terhadap waktu. Artinya kondisi model tidak boleh berubah dalam rentang waktu pengontrolan, walaupun dalam kenyataannya kondisi yang benar-benar linier tidak ada. Sehingga kontrol PID merupakan kontrol yang paling tepat untuk kondisi model yang ideal dengan mengabaikan kondisi-kondisi non linier yang dianggap sulit untuk dimodelkan secara matematis.

#### 2.1.1 Kontrol P (*Proportional*)



**Gambar 2.1** Diagram Blok Kontrol *Proportional* <sup>[1]</sup>

**Gambar 2.1** merupakan diagram blok dari kontrol P. Kontrol *proportional* merupakan penguatan murni nilai  $K_p$ . Berdasarkan blok diagram diatas, persamaan output kontroler  $u$ ,

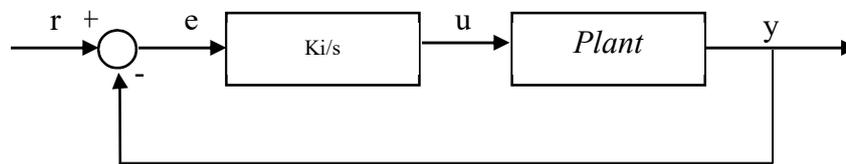
$$u = K_p \cdot e \dots\dots\dots(1)$$

Pertambahan nilai  $K$  akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar respon dan mengurangi *error steady state*. Dengan

menggunakan kontrol P saja, hasil kurang memuaskan karena dengan penambahan nilai K selain membuat sistem lebih sensitif, juga cenderung mengakibatkan ketidakstabilan sistem.

Nilai output bergantung perkalian *error*, yakni kecepatan referensi dikurangi kecepatan aktual, dengan konstanta  $K_p$ . Jika *error* positif berarti kecepatan aktual lebih kecil daripada kecepatan referensi. Jika *error* bernilai negatif berarti kecepatan aktual lebih besar daripada kecepatan referensi.

### 2.1.2 Kontrol I (*Integral*)



**Gambar 2.2** Diagram Blok Kontrol *Integral* <sup>[1]</sup>

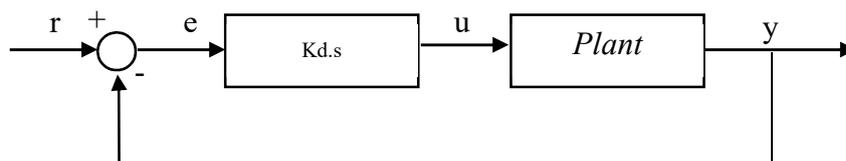
**Gambar 2.2** merupakan diagram blok dari kontrol I. Jika dimisalkan  $G(s)$  adalah kontrol I, maka

$$u(t) = \left[ \int_0^t e(T) dT \right] K_i \dots \dots \dots (2)$$

dengan  $K_i$  adalah konstanta integral, sehingga  $G(s)$  dapat dinyatakan  $G(s) = \frac{K_i}{s}$ . Jika  $e(t)$  mendekati konstanta (bukan nol), maka  $u(t)$  akan menjadi sangat besar, sehingga diharapkan dapat memperbaiki *error*. Jika  $e(t)$  mendekati nol, maka efek kontrol *integral* akan semakin kecil.

Kontrol *integral* dapat memperbaiki *error steady state*, namun apabila pemilihan nilai  $K_i$  tidak tepat dapat mengakibatkan ketidakstabilan sistem karena nilai  $K_i$  dapat menyebabkan respon transien yang tinggi.

### 2.1.3 Kontrol D (*Derivative*)



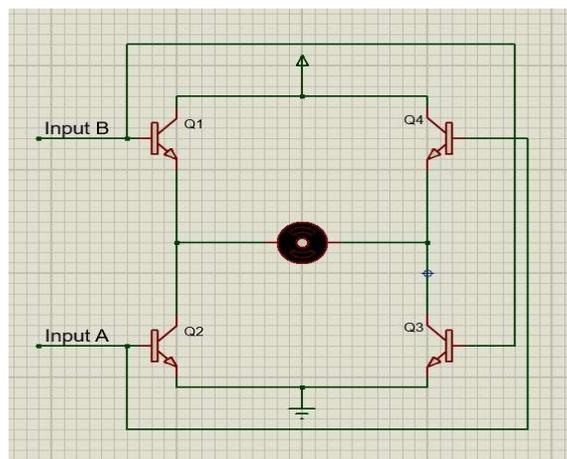
**Gambar 2.3** Diagram Blok Kontrol *Derivative* <sup>[1]</sup>

**Gambar 2.3** merupakan diagram blok dari kontrol D. Kontrol D ini dapat digunakan untuk memperbaiki respon transien dengan memprediksi *error* yang akan terjadi. Kontrol D hanya dapat berubah saat terjadi perubahan *error*, sehingga ketika  $\Delta error$  minim kontrol ini tidak akan beraksi. Hal ini menyebabkan kontrol D tidak dapat berdiri sendiri.

Sinyal kontrol  $u$  yang dihasilkan oleh kontrol D dapat dinyatakan sebagai  $u = Kd \cdot e$  atau  $u = Kd(\Delta e/\Delta t)$ , sehingga jika dimisalkan  $G(s)$  merupakan kontrol D, maka  $G(s) = s \cdot Kd$ . Dari persamaan tersebut dapat dilihat bahwa sifat dari kontrol ini bermain dalam konteks “kecepatan” atau *rate* dari *error*. Dengan sifat ini dapat digunakan untuk memperbaiki *error* yang akan terjadi. Umpan balik yang diberikan adalah sebanding dengan kecepatan perubahan  $e(t)$ , sehingga kontroler dapat mengantisipasi *error* yang akan terjadi.

## 2.2 *Driver Motor H-Bridge*

H-Bridge adalah sebuah perangkat berupa rangkaian yang berfungsi untuk menggerakkan motor. Prinsip kerja pada *driver motor* ini adalah dengan mengatur mati-hidupnya pada transistor. Pada rangkaian H-Bridge terdiri dari 4 buah transistor NPN seperti pada **Gambar 2.4** di bawah ini.



**Gambar 2.4** Rangkaian H-Brigde

Terdapat beberapa kondisi pada rangkaian H-Bridge diatas, diantaranya :

- a. Saat *input* A dan B = 0  
Saat *input* pada A dan B memiliki logika nol, maka transistor Q2 dan Q3 tidak akan mendapat catuan pada kaki basis, sehingga transistor bersifat *cut-off* atau transistor bersifat saklar terbuka. Pada gambar, kaki kolektor transistor Q1 dan Q4 bergantung pada Q2 dan Q3. Sehingga apabila tidak ada arus pada Q2 dan Q3, maka arus pada Q1 dan Q4 juga tidak ada. Hal tersebut menyebabkan motor tidak berputar.
- b. Saat *input* A = 1 dan B = 0  
Saat *input* A berlogika 1 dan B berlogika 0, maka Q2 akan saturasi dan Q3 akan *cut-off*. Sehingga Q4 juga akan mengalami saturasi. Akibatnya, arus akan mengalir dari  $V_{dc} - Q4 - \text{Motor DC} - Q2 - \text{Ground}$ . Motor pun akan bergerak berlawanan arah jarum jam.
- c. Saat *input* A = 0 dan B = 1  
Saat *input* A berlogika 0 dan B berlogika 1, maka Q3 akan saturasi dan Q2 akan *cut-off*. Sehingga Q1 juga akan mengalami saturasi. Akibatnya, arus akan mengalir dari  $V_{cc} - Q1 - \text{Motor DC} - Q3 - \text{Ground}$ . Motor pun akan bergerak searah jarum jam.
- d. Saat *input* A dan B = 1  
Saat kedua *input* berlogika 1, maka semua transistor akan mengalami saturasi. Akibatnya tidak ada perbedaan potensial pada konektor motor DC. Secara logika, apabila pada motor dc tidak terdapat beda potensial, maka motor tidak akan berputar. Namun, hal ini dapat menyebabkan transistor mengalami panas berlebih dan akan mengakibatkan transistor menjadi rusak.

### 2.3 Mikrokontroler ATmega 328P

Mikrokontroler ATmega 328P merupakan salah satu IC mikrokontroler 8-bit keluarga AVR. Mikrokontroler ini memiliki performa yang baik dengan energi yang rendah. Salah satu sistem minimum mikrokontroler yang menggunakan ATmega 328P adalah Arduino UNO.

Arduino UNO merupakan suatu *board* mikrokontroler *open source* yang menggunakan IC ATMEga 328P yang sudah dilengkapi dengan fitur *self programming* melalui serial *Bootloader* yang terhubung dengan USB. Karena telah dilengkapi dengan USB to serial, Arduino dapat langsung melakukan komunikasi secara langsung melalui komputer dan juga sebagai *downloader* untuk memasukkan program yang telah dibuat.

Arduino dapat diaktifkan melalui catu daya eksternal, seperti USB, adaptor, atau baterai. *Board* Arduino dapat bekerja pada rentang tegangan antara 6 VDC sampai 20 VDC. Namun rentang tegangan yang dianjurkan yaitu antara 7 VDC sampai 12 VDC. Jika tegangan dibawah 6 VDC, tegangan pada *board* Arduino tidak akan stabil. Jika tegangan lebih dari 12 VDC, regulator tegangan bisa panas dan dapat merusak *board* Arduino.



**Gambar 2.5** *Board* Mikrokontroler Arduino UNO<sup>[7]</sup>

## 2.4 Robot Tank

Robot tank merupakan salah satu jenis robot darat yang biasa dipakai untuk menghadapi medan yang cukup berat. Robot ini memiliki ciri khusus pada bagian roda yang berbentuk layaknya roda pada kendaraan tank sesungguhnya. Roda jenis ini dapat mencengkram tanah yang licin, sehingga robot dapat melewatinya dengan mudah.

Robot jenis ini biasanya menggunakan sistem penggerak AWD (*All Wheel Drive*). Sistem penggerak *All Wheel Drive* atau sering disebut AWD adalah sistem penggerak pada robot darat beroda yang tiap rodanya dikendalikan oleh sebuah motor. Misalkan sebuah robot beroda memiliki empat buah roda, maka motor yang digunakan juga empat buah. Sehingga, tiap motor dapat dikendalikan secara

terpisah. Berbeda dengan sistem penggerak 4WD atau 2WD yang penggeraknya menggunakan satu buah motor. Akibat penggunaan motor yang lebih banyak sistem penggerak AWD dapat menghasilkan tenaga yang lebih besar, sehingga dapat menyebabkan robot bergerak dengan mudah pada medan yang berat. Akan tetapi, sistem penggerak AWD memerlukan pasokan listrik yang lebih banyak untuk menggerakkan keempat buah motornya. **Gambar 2.6** merupakan salah satu contoh robot tank yang biasa digunakan.



**Gambar 2.6** Salah Satu Contoh Robot Tank<sup>[9]</sup>

Sistem penggerak pada robot tank merupakan kombinasi dari tiga buah bagian, yaitu :

1. *Drive Sprocket*

*Drive sprocket* merupakan bagian roda yang terhubung langsung dengan motor. Roda ini biasanya memiliki ukuran lebih besar dibandingkan dengan *idler sprocket*.

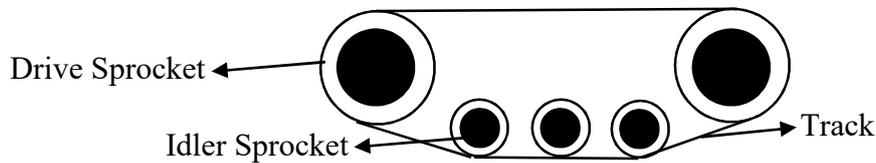
2. *Idler Sprocket*

*Idler sprocket* merupakan bagian roda yang tidak terhubung dengan motor dan berfungsi untuk membantu pergerakan robot tank. Roda ini berukuran lebih kecil dibandingkan dengan *drive sprocket*.

3. *Track*

*Track* merupakan bagian yang bersentuhan langsung dengan tanah. Dengan bentuk yang tidak rata, bagian ini sangat berguna untuk mencengkram tanah yang

licin, sehingga robot tidak terjadi *slip*. Bagian ini juga berfungsi sebagai penghubung antara *drive sprocket* dan *idler sprocket*.



**Gambar 2.7** Bagian-bagian pada roda tank

## 2.5 Motor DC

Motor DC adalah sebuah peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang desain awalnya diperkenalkan oleh Michael Faraday. Karena motor DC sering digunakan pada sistem kendali untuk tujuan analisis, maka diperlukan model matematis motor DC untuk diaplikasikan pada kendali menggunakan rangkaian ekivalen untuk merepresentasikan motor DC dengan magnet permanen. Untuk analisis linier diasumsikan bahwa torsi motor yang dihasilkan motor sebanding dengan fluks pemisah udara dan arus jangkar. Medan konstan, fluks konstan, dan torsi mempunyai arah sesuai kumparan magnet.

$$T_m(t) = K_T I_a \dots \dots (3)$$

Ketika konduktor bergerak pada medan magnet, suatu tegangan dibangkitkan melintasi ujung-ujungnya. Tegangan balik ini yang sebanding dengan kecepatan motor, berlawanan dengan aliran arus. Hubungan tegangan balik ( $e_b$ ) dengan kecepatan motor ( $\omega_m$ ) rad/detik adalah :

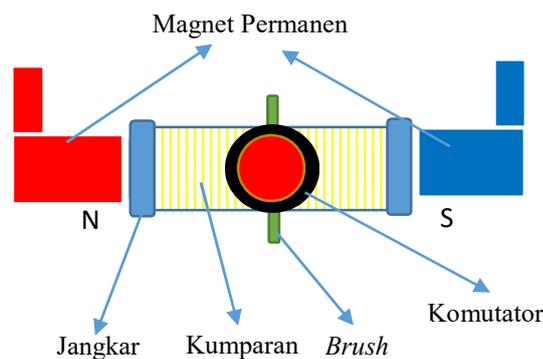
$$e_b = K_b \omega_m(t) \dots \dots (4)$$

Cara kerja motor DC berdasarkan arus yang masuk pada motor DC tersebut. Arus yang melalui kumparan jangkar akan menyebabkan jangkar memiliki medan magnet. Kutub magnet pada jangkar ditarik oleh kutub magnet permanen yang polaritasnya berbeda sehingga jangkar berputar. Jika kutub jangkar satu garis dengan kutub medan magnet permanen, *brush* berada pada celah komutator dan tidak ada arus yang mengalir. Gaya tarik magnet berhenti dan kelembaman membawa jangkar ke titik netral. Komutator membalik arus jangkar dan medan

magnet permanen. Hal ini menyebabkan polaritas pada jangkar berubah. Kutub yang sama antara kutub jangkar dan medan magnet permanen membuat keduanya saling menolak sehingga jangkar kembali berputar dan begitu seterusnya.



**Gambar 2.8** Motor DC



**Gambar 2.9** Ilustrasi Bagian Motor DC

## 2.6 LabVIEW

LabVIEW merupakan sebuah *software* pemrograman yang dikembangkan oleh perusahaan National Instrument. Berbeda dengan *software* pemrograman lain yang berbasis teks, bahasa pemrograman yang digunakan LabVIEW berbasis grafis atau diagram blok. Alur data yang diproses pada LabVIEW bekerja berdasarkan alur dari diagram blok yang disusun oleh *programmer*. LabVIEW juga disebut sebagai VI atau *Virtual Instrument* karena program yang dibuat pada LabVIEW dapat menyerupai instrumen sesungguhnya, seperti *knob*, *push button*, *led*, *grafik*, *level meter*, dan lain-lain. Pada LabVIEW terdapat tiga komponen utama, yaitu :