

## **SISTEM PENGATURAN PID MOTOR DC SEBAGAI PENGGERAK MINI CONVEYOR BERBASIS MATLAB**

**Ahmadan Ainul Fikri**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia.  
E-mail : ahmadanfikri@mhs.unesa.ac.id

**Endryansyah**

Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia.  
E-mail : endryansyah@unesa.ac.id

### **Abstrak**

*Conveyor* banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Pada proses produksi saat ini, produk yang dihasilkan memiliki berat yang sangat beragam. Pada *conveyor* yang tidak dilengkapi kontrol kecepatan, apabila mendapat beban berlebih akan mempengaruhi kecepatan motor itu sendiri. Tujuan penelitian ini adalah untuk merancang Sistem Pengaturan PID Motor DC sebagai Penggerak Mini *Conveyor* Berbasis Matlab untuk mengetahui bagaimana mengatur kecepatan motor conveyor pada *plant* mini *conveyor* yang sesungguhnya, dimana faktor-faktor tertentu tidak dianggap 0. Perancangan *hardware* pada mini *conveyor* ini menggunakan besi berongga sebagai rangka dan 2 *roller* sebagai penarik *belt conveyor*. Sistem pengaturan PID menggunakan metode *analytic* yang mencari respon sistem dari motor kemudian menghitung nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang sesuai. Kemudian diaplikasikan sebagai modul kontrol kecepatan di dalam sistem yang menggunakan Matlab dan Arduino sebagai piranti data akuisisi. Sensor kecepatan (*Rotary Encoder*) dipasang pada *roller* agar bisa membaca kecepatan secara tepat. Sensor ini akan mengirim data berupa *Pulse Width Modulation* (PWM) pada setiap *rising*-nya agar dapat menghitung kecepatan secara akurat, serta mengendalikan kecepatan motor sesuai dengan *set point* yang diinginkan. Hasil penelitian ini didapati nilai  $K_p=0.94624747$ ,  $K_i=51.4023958$  dan  $K_d=0.01941504$ . Kontroler PID yang dirancang untuk mengendalikan respon kecepatan motor DC pada penelitian ini berhasil mencapai nilai *setpoint* dan mengurangi *Ess* (*Error Steady State*) pada respon sistem riil dari 47,16% menjadi 1,015188% (tanpa beban) dan 2,2020751% (menggunakan beban besi timbangan).

**Kata Kunci:** Mini *Conveyor*, Kontrol Kecepatan, Metode *Analytic*, PID, Arduino.

### **Abstract**

*Conveyor* is used by mayor industry for moving object in massive amount and continuously. Nowadays, in production process, product that produced has diverse load, from light to heavy. The Speed on uncontrolled conveyor will be affected by the overload weight. The objective of this research is designing PID control DC motor system as the driver mini conveyor based on Matlab for discovering how to controlling speed on real plant of mini conveyor where some factors that effecting the research isn't count as 0. Hardware Design on this mini conveyor using hollow steel as the frame and 2 copy of roller belt for stretching belt conveyor. PID control system using analytical method to get the respond system of the DC motor for next to calculate the best fit of  $K_p$ ,  $K_i$  and  $K_d$ . Then applying those PID control system using Matlab and Arduino as the data acquisition tools. Speed sensor (*Rotary Encoder*) placed on the roller belt to get the reading speed accurately. These sensor will sending data in the PWM form on every rising to be able for calculate the speed accurately and the same time controlling speed base on setpoint that has been set. The result of this research is value of  $K_p=0.94624747$ ,  $K_i=51.4023958$  and  $K_d=0.01941504$ . PID control that designed to controlling respond of motor DC speed on this research has been successfully reached setpoint value and decreasing *Ess* on real respond system from 47,16% to 1,015188% (unloaded) and 2,2020751% (loaded).

**Keywords:** Mini Conveyor, Speed Control, Analytical Method, PID, Arduino.

### **PENDAHULUAN**

*Conveyor* adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari suatu tempat ke tempat yang lain. *Conveyor* banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Pada proses produksi saat ini, produk yang dihasilkan memiliki berat yang sangat beragam. Pada *conveyor* yang tidak dilengkapi kontrol kecepatan, apabila mendapat beban berlebih akan mempengaruhi kecepatan motor itu sendiri.

Pada waktu motor dibebani, arus jangkar akan meningkat seiring dengan kenaikan beban. Kenaikan arus jangkar akan menyebabkan gangguan medan jangkar pada medan utama ikut membesar pula, sehingga titik netral medan utama makin bergeser. Bergesernya medan utama akan menaikan rugi-rugi pada sikat jika posisi sikat tidak ikut disesuaikan.(Ananda, 2003). Nilai efisiensi rata-rata pada motor tanpa kontrol kecepatan pada penelitian yang dilakukan oleh Stephanus Antonius Ananda juga terbilang rendah, sekitar 70%.

Hal ini akan mempengaruhi kecepatan produksi. Seiring dengan kemajuan ilmu pengetahuan dan teknologi ada banyak cara untuk untuk mengontrol kecepatan motor, salah satunya menggunakan kontroler PID.

Penelitian sistem pengaturan kecepatan motor menggunakan kontroler PID sudah pernah dilakukan oleh Eka Bayu (Prinandika, 2013). Dalam penelitian tersebut Eka Bayu Menggunakan robot *line follower* sebagai *plant* dan menghasilkan maksimum *overshoot* yang cukup besar mencapai 10%.

Penelitian ini membahas tentang “Sistem Pengaturan PID Motor DC sebagai Penggerak Mini Conveyor Berbasis Matlab” untuk mengetahui bagaimana mengatur kecepatan motor pada *plant* mini *conveyor* yang sesungguhnya, dimana faktor-faktor tertentu tidak dianggap 0. Sistem pengaturan PID yang diaplikasikan sebagai modul kontrol kecepatan di dalam sistem yang menggunakan Matlab dan Arduino sebagai piranti data akuisisi untuk membaca sensor kecepatan serta mengendalikan kecepatan motor untuk disesuaikan dengan *set point* yang diinginkan.

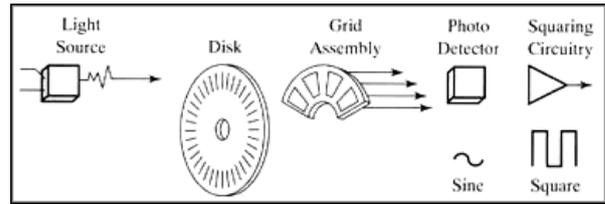
Hasil penelitiann ini diharapkan bisa dijadikan acuan untuk mengontrol kecepatan motor DC dengan menggunakan kontroler PID pada *plant* yang sesungguhnya.

**KAJIAN PUSTAKA**

**Rotary Encoder**

*Rotary encoder* adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. *Rotary encoder* umumnya digunakan pada pengendalian robot dan penggerak motor.

Susunan *rotary encoder* berupa suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Disisi 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 perangkat berputar lainnya 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. Gambar 1. menunjukkan bagan skematik sederhana dari *rotary encoder*. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary encoder* tersebut.



Gambar 1. Blok Penyusunan *Rotary Encoder*  
(Sumber : <http://www.ni.com/white-paper/14805/en/>)

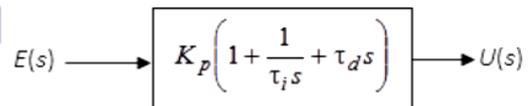
Rangkaian penghasil pulsa yang digunakan umumnya memiliki *output* yang berubah dari + 5V menjadi 0.5V ketika cahaya diblok oleh piringan dan ketika diteruskan ke *photo-transistor*. Karena perangkat ini umumnya bekerja dekat dengan motor DC maka banyak noise yang timbul sehingga biasanya *output* akan dimasukkan ke *low-pass filter* dahulu. Apabila *low-pass filter* digunakan, frekuensi *cut-off* yang dipakai umumnya ditentukan oleh jumlah slot yang ada pada piringan dan seberapa cepat piringan tersebut berputar, dinyatakan dengan:

$$f_c = \frac{s_w n}{60} \tag{1}$$

Keterangan,  $f_c$  : frekuensi *cut-off* filter  
 $s_w$  : kecepatan piringan  
 $n$  : jumlah slot pada piringan.

**Kontrol PID**

Dalam suatu sistem kendali otomatis dikenal adanya beberapa aksi pengendalian, diantaranya aksi kendali *On/Off*, aksi kendali proporsional (P), aksi kendali integral (I) dan aksi kendali differensial (D). Masing-masing aksi kendali tersebut mempunyai keunggulan tertentu. Aksi kendali *On/Off* mempunyai keunggulan waktu naik yang cepat, aksi kendali proporsional mempunyai keunggulan waktu naik yang cepat dan stabil, aksi kendali integral mempunyai keunggulan untuk memperkecil kesalahan, dan aksi kendali differensial memiliki keunggulan meredam kekurangan tanggapan atau kelebihan tanggapan. Untuk mendapatkan suatu sistem kendali dengan hasil pengendalian yang memiliki waktu naik yang cepat, kesalahan yang kecil dan kestabilan yang baik, dapat dilakukan dengan menggabungkan ketiga aksi kendali tersebut menjadi aksi kendali PID (Gunterus, 1994).



Gambar 2. Diagram Blok Kontrol PID  
(Sumber : Katsuhiko Ogata)

$$U(s) = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) E(s) \tag{2}$$

$$K_p = \frac{\tau_i}{K_{xt}} \tag{3}$$

$$K_i = K_p \times \frac{1}{\tau_i} \tag{4}$$

$$K_d = K_p \times \tau_d \tag{5}$$

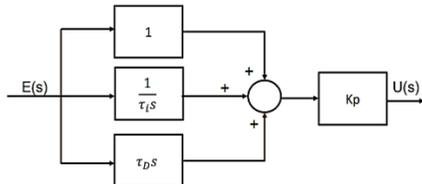
Keterangan,:

$K_p$  = Penguatan Proporsional

$U(s)$  = Sinyal Kendali

- Ki = Penguatan Integral
- E (s) = Sinyal Error
- Kd = Penguatan Derivatif
- $\tau_{is}$  = Konstanta Waktu Integral
- $\tau_{ds}$  = Konstanta Waktu Derivatif

Kontrol PID adalah gabungan dari tiga macam kontroler, yaitu pengontrol proporsional (*Propotional Controller*), pengontrol integral (*Integrall Controller*) dan pengontrolan turunan (*Derivative Controller*) (Doren V.V., 2009).



Gambar. 3. Struktur Kontrol PID (Sumber : Katsuhiko Ogata)

Gambar 3. menunjukkan struktur kontrol PID ideal. Struktur kontrol PID ideal adalah struktur kontrol PID yang sering dijumpai.

Metode umum untuk mendapatkan nilai awal kontrol PID adalah untuk merancang sistem kontrol PID. Hal ini disebabkan karena parameter  $K_p$ ,  $K_i$  dan  $K_d$  tidak *independent*. Untuk mendapatkan aksi kontrol yang baik diperlukan langkah mencoba kombinasi antara P, I dan D sampai ditemukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  seperti yang diinginkan (Ali, 2004).

### Metode PID

Metode pembelajaran yang digunakan pada penelitian ini adalah Metode *Analitic* dengan pendekatan system Orde-1. Pada metode ini respon motor secara *open loop* tanpa kontroller akan digunakan untuk mencari respon system yang kemudian akan dihitung menggunakan rumus dibawah ini untuk mendapatkan *transfer function* motor, serta nilai  $K_p$ ,  $K_i$  dan  $K_d$  (Pratama, 2018)

$$K = \frac{Y_{ss}}{X_{ss}} \quad (6)$$

$$C\tau = Y_{ss} \cdot 0,632 \quad (7)$$

$$\tau_s (5\%) = 3\tau \quad (8)$$

$$\tau_r (10-90\%) = \tau \ln 9 \quad (9)$$

$$\tau_d = \tau \ln 2 \quad (10)$$

$$E_{ss} = \frac{X_{ss} - Y_{ss}}{X_{ss}} \times 100\% \quad (11)$$

Keterangan :

- K = Gain Overall
- $X_{ss}$  = Setpoint (RPM)
- $Y_{ss}$  = Rata-rata respon sistem saat *steady state* (RPM)
- $C\tau$  = Nilai saat respon sistem mencapai 63,2% dari *steady state* (RPM)
- $\tau$  = Waktu saat respon sistem mencapai 63,2% dari *steady state* (s)
- $\tau^*$  = Kecepatan respon sistem yang diinginkan

- $\tau_s (5\%)$  = Waktu tunak  $\pm 5\%$  dari *steady state* (s)
- $\tau_r (10-90\%)$  = Waktu naik dari 10-90% dari *steady state* (s)
- $\tau_d$  = Waktu tunda dari 0-50% dari *steady state* (s)

### PWM (*Pulse Width Modulation*)

PWM adalah singkatan dari *Pulse Width Modulation*, yaitu teknik yang biasa digunakan untuk mengontrol daya ke perangkat listrik, dibuat praktis dengan *switch* daya elektronik. Metode PWM merupakan metode untuk pengaturan kecepatan dengan cara mengatur persentase lebar pulsa *high* terhadap periode dari suatu sinyal persegi dalam bentuk tegangan periodik yang akan diproses mikrokontroller lalu diteruskan ke *driver motor* sebagai pengatur kecepatan.

### Arduino Uno R3

Arduino Uno R3 adalah *prototyping platform* sebuah paket berupa papan (*board*) elektronik (*hardware*) dan lingkungan pengembangan (*software*) yang memanfaatkan kemampuan mikrokontroller jenis tertentu. Mikrokontroller yang digunakan pada Arduino Uno adalah jenis Atmel seri ATmega 328 (Wardana, 2015).



Gambar 4. Papan Arduino Uno R3

(Sumber : <https://www.arduino.cc/en/Main/Products>)

*Prototyping platform* ini berkomunikasi secara serial dengan perangkat kabel USB yang digunakan untuk *upload* program dari *software* IDE Arduino. Penelitian ini menggunakan pin-pin data *analog* dan digital. Untuk keperluan PWM (*Pulse Width Modulation*) terdapat pada pin 3,5,6,9,10 dan 11.

### METODE PENELITIAN

#### Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif. Tujuan Penelitian ini adalah menggunakan dan mengembangkan model matematis, teori-teori atau hipotesis fisika. Proses pengukuran dalam metode ini berperan sentral pada penelitian kuantitatif karena hal ini memberikan hubungan yang fundamental antara pengamatan empiris dan ekspresi matematis dari hubungan-hubungan kuantitatif (John W. Creswell, 2003).

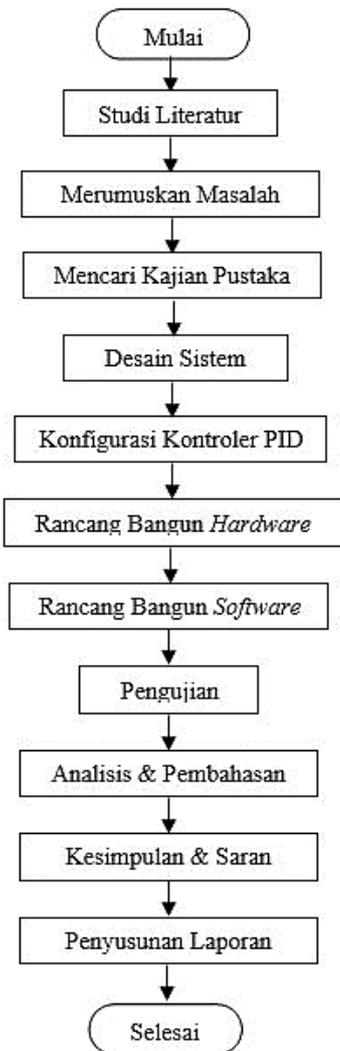
Pada penelitian ini menggunakan *software* Arduino IDE dan Matlab untuk simulasi menjalankan sistem yang akan digunakan beserta rancang bangun plant yang akan menghasilkan respon kecepatan terhadap beban pada plant *mini conveyor*.

**Waktu dan Tempat Penelitian**

Penelitian ini dilaksanakan di Laboratorium Sistem Pengaturan dan Kendali Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester gasal 2018/2019.

**Rancangan Penelitian**

Rancangan penelitian ini bertujuan untuk menjawab permasalahan dalam rangka merumuskan kesimpulan, seperti dijelaskan pada diagram alir sebagai berikut:



Gambar 5. Diagram Alir Langkah-Langkah Penelitian (Sumber: Data Primer, 2017)

Penelitian ini banyak mengacu pada jurnal-jurnal terdahulu untuk dimasukkan kedalam kajian pustaka. Desain sistem dirumuskan berdasarkan fenomena industri yang ada. Konfigurasi PID menggunakan metode analitic. Dalam merancang *hardware* dan *software* dikonsultasikan dengan teknisi yang berpengalaman. Pengujian penelitian ini menggunakan beban sebagai *disturbance*. Kesimpulan dan saran murni dari hasil penelitian yang kemudian disusun dalam sebuah laporan untuk mendukung penelitian-penelitian selanjutnya.

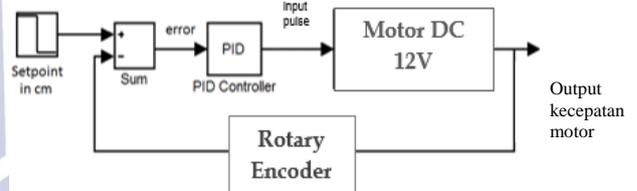
**Rancang Bangun Hardware**

Berikut skema rangkaian keseluruhan *hardware* untuk sistem pengaturan kecepatan motor pada rancang bangun mini conveyor berbasis PID.



Gambar 6. Skema Rangkaian *Hardware* Desain Sistem Pengaturan PID Motor DC sebagai Penggerak Mini Conveyor Berbasis Matlab. (Sumber: Data Primer, 2017)

Pada Gambar 6, komputer sebagai piranti pemroses data, penerima, pengirim program melalui kabel USB ke Arduino UNO R3, menampilkan GUI (*Graphical User Interface*) dari Matlab untuk pembacaan data dalam bentuk grafik. Software Matlab yang digunakan adalah jenis Matlab 2009a yang sudah terintegrasi dengan Arduino untuk membuat program pengendalian kekecepatan motor pada mini conveyor berbasis PID. Tampilan monitoring berasal dari Matlab yang didapat dari pembacaan sensor dengan beberapa data dalam bentuk grafik/nilai angka.

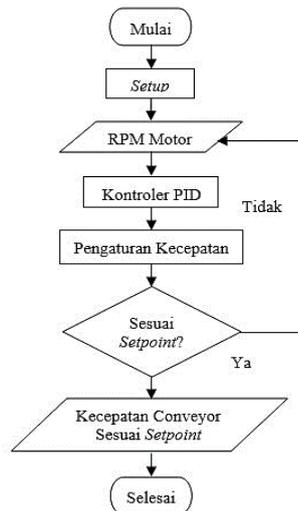


Gambar 7. Diagram Blok Sistem Pengaturan Kecepatan Motor (Sumber: Data Primer, 2017)

Pada Gambar 7. merupakan diagram blok dari sistem pengaturan kecepatan motor pada rancang bangun mini conveyor. Nilai masukan (*Set point*) sistem berupa kecepatan yang dikehendaki. Hasil dari keluaran kontrol PID berupa pergerakan motor yang mendapat pulsa dari PWM untuk mengatur daya yang masuk ke motor. Sensor kecepatan (*Rotary Encoder*) yang akan digunakan untuk mengukur laju pada belt mini conveyor dengan skala m/s (meter per sekon) dan selanjutnya hasil dari nilai pembacaan sebagai nilai umpan balik.

**Rancang Bangun Software**

Rancang bangun *software* yang dimaksud adalah sebuah program yang dimasukkan ke dalam mikrokontroler Arduino. Pemrograman dibuat menggunakan bahasa pemrogram khusus oleh Arduino pada Arduino IDE (*Integrated Development Environment*) versi 1.6.12. Perangkat ini bersifat *open source*, maka banyak *library* yang dikembangkan oleh personal atau komunitas di luar *developer* Arduino itu sendiri.



Gambar 8. Flowchart Rancangan Software (Sumber: Data Primer, 2017)

Penelitian ini dimulai dengan melakukan *Setup* nilai *Setpoint* yang diinginkan. Kemudian motor akan melaju. *Rotary Encoder* yang menempel pada motor akan memberikan *pulse* RPM ke kontroler PID yang sudah dihitung menggunakan metode analitic untuk memberikan *feedback* agar kecepatan motor bisa diatur. Apabila belum sesuai *setpoint* maka proses tersebut akan diulang sampai mendapatkan *setpoint* yang diinginkan.

**HASIL DAN PEMBAHASAN**

Pengujian dan analisis penelitian meliputi; 1) analisis dan pengujian rancangan rangkaian sistem kontrol *plant* untuk PID, 2) Pengaplikasian konfigurasi kontrol PID pada *plant*, 3) Hasil pengujian kontrol PID.

**1) Pengujian Rancangan Rangkaian**

Pengujian rancangan rangkaian meliputi catu daya, sensor *rotary encoder*, mikrokontroler, dan *driver* motor. Berikut tiap blok pengujian rangkaian:

**Catu Daya**

Catu daya pada sistem ini memiliki peran yang penting yaitu sebagai sumber tegangan pada sistem. Pada sistem ini, digunakan 2 buah catu daya. Catu daya yang pertama memiliki output 5V DC untuk memberi tegangan pada blok mikrokontroler dan sensor. Catu daya yang kedua memiliki output 12V DC untuk memberi tegangan pada driver motor. Berikut adalah tabel hasil pengukuran catu daya untuk sistem.

Tabel 1. Hasil Pengujian Catu Daya (Sumber : Data Primer, 2018)

No.	Pengukuran	Tegangan (V)
1.	Input Catu Daya	232
2.	Output Catu Daya 5V	5,1
3.	Output Catu Daya 12V	11,98
4.	Output Catu Daya 9V	8,99
5.	Output Catu Daya 15V	14,96

**Blok Sensor Rotary Encoder**

Proses kalibrasi sensor *rotary encoder* dilakukan agar didapatkan nilai RPM pada Motor DC yang digunakan. Alat bantu kalibrasi yang digunakan adalah *tachometer digital* tipe DT-2234C yang ada di Lab Kendali Universitas Negeri Surabaya. Tabel kalibrasi ditunjukkan pada Tabel 4.2.

Tabel 2. Data Kalibrasi Sensor Rotary Encoder. (Sumber : Data Primer, 2018)

No.	Pembacaan Sensor Rotary Encoder (RPM)	Pembacaan Tachometer Digital (RPM)	Power Supply (V)
1.	47,55	49	12
2.	57,07	54,30	12
3.	91,50	89,40	12

**Blok Mikrokontroler**

Pada penelitian ini, mikrokontroler yang digunakan adalah Arduino Uno R3. Pengujian mikrokontroler dilakukan dengan menghubungkan sensor ke mikrokontroler untuk menghitung kecepatan motor yang kemudian ditampilkan pada PC melalui *software* Arduino. Data dari blok mikrokontroler sama seperti Tabel 2. karena mikrokontroler yang memproses data yang diinput oleh sensor *rotary encoder*.

**Blok Driver Motor**

Pengujian blok motor driver dilakukan dengan pemberian variasi *duty cycle* PWM 0%, 25%, 50%, dan 75% pada driver motor kemudian dilakukan pengukuran kecepatan motor DC tanpa dipasang ke *conveyor*. Hasil pengujian *driver* motor ditampilkan pada Tabel 3 dibawah ini.

Tabel 3. Hasil Pengujian Driver Motor (Sumber : Data Primer, 2018)

No,	Duty Cycle PWM (%)	Kecepatan Motor DC (RPM)
0.	0	464
1.	10	11
2.	20	21
3.	30	33
4.	40	50
5.	50	65
6.	60	76
7.	70	85
8.	80	91
9.	90	96
10.	100	97

Agar lebih mudah dipahami, Tabel 3 dapat disajikan dengan tampilan grafik. Berikut adalah grafik *duty cycle* terhadap kecepatan motor DC:



Gambar 9. Grafik Pengujian Driver Motor DC (Sumber: Data Primer, 2018)

2) **Pengaplikasian Konfigurasi Kontrol PID**

Pada penelitian ini metode yang digunakan untuk mencari pemodelan system adalah dengan menggunakan metode pendekatan karakteristik sistem orde-1 dengan parameter yang dibutuhkan adalah  $K$  dan  $\tau$ . Metode pendekatan karakteristik sistem orde-1 ini dilakukan dengan mendapatkan respon sistem yang akan digunakan untuk mendapatkan model atau fungsi alih dari suatu sistem. Fungsi alih tersebut akan digunakan sebagai perbandingan antara respon riil dengan respon simulasi (Maulana, 2018). Pada penelitian ini dilakukan pemodelan dengan *setpoint* 90 rpm di tegangan 12V.

Berikut cara mencari nilai  $K$  dan  $\tau$  pada setpoint 90 di tegangan 12V,

$$X_{SS} = 9$$

$$Y_{SS} = 47,5563$$

$$K = \frac{Y_{SS}}{X_{SS}} = \frac{47,5563}{9} = 0,528403$$

$$C\tau = Y_{SS} \times 0,632 = 47,5563 \times 0,632 = 30,05558$$

selanjutnya kita mencari nilai  $\tau$  dengan memasukkan parameter diatas ke rumus :

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_2} \quad (15)$$

$$\frac{x - 1}{31 - 1} = \frac{30,05558 - 10}{41 - 10}$$

$$\frac{x - 1}{30} = \frac{20,05558}{31}$$

$$31x - 31 = 601,6674$$

$$31x = 570,6674$$

$$x = \tau = 18,4086258 \text{ mS}$$

$$x = \tau = 0,0184086258 \text{ S}$$

Dari parameter tersebut, didapatkan fungsi alih dengan *setpoint* 90 pada tegangan 12V yaitu :

$$G(s) = \frac{K}{\tau_s + 1}$$

$$G(s) = \frac{0,528403}{0,0184086258 s + 1}$$

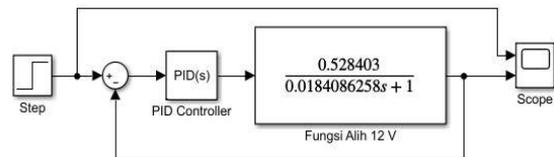
Keterangan :

$x_1$  = Waktu sebelum respon mencapai sistem mencapai 63,2% dari *steady state*

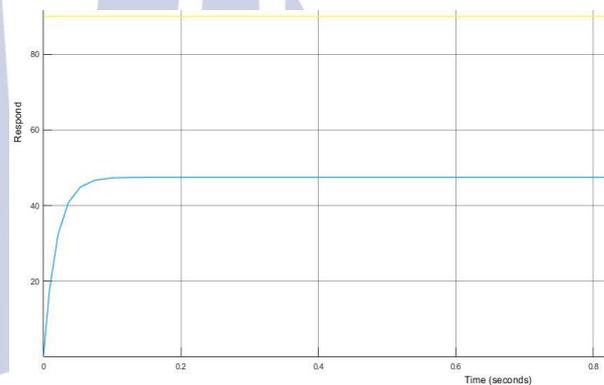
$x_2$  = Waktu sebelum respon mencapai sistem mencapai 63,2% dari *steady state*

$y_1$  = Respon pada saat  $x_1$

$y_2$  = Respon pada saat  $x_2$



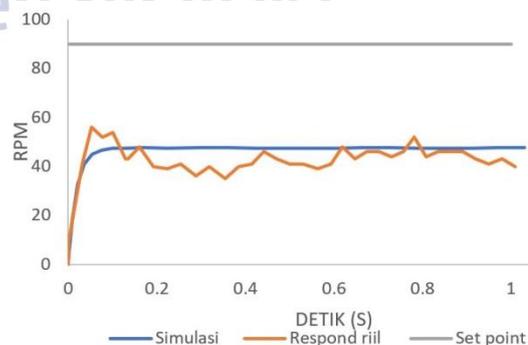
Gambar 10. Blok Simulink Simulasi Fungsi Alih 12V (Sumber : Data Primer, 2018)



Gambar 11. Grafik Respon *Open Loop* Simulasi Fungsi Alih 12 V

(Sumber : Data Primer, 2018)

Pada Gambar 11 nilai  $K_p = 1$  sehingga respon motor masih jauh dari nilai setpoint. Sedangkan perbandingan respon dari simulasi dengan respon sistem (riil) ditunjukkan pada Gambar 13 :



Gambar 12. Grafik Perbandingan Respon Simulasi dengan Respon Riil.

(Sumber: Data Primer, 2018)

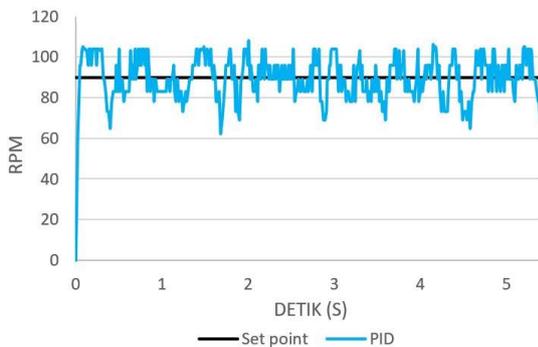
Berdasarkan grafik respon, dapat dihitung parameter respon riil secara *open loop* berupa  $\tau_s$  (*Steady Time*),  $\tau_r$  (*Rise Time*),  $\tau_d$  (*Time Delay*) dan  $E_{ss}$  (*Error Steady State*) dengan nilai  $\tau = 0,0184086258$  detik melalui persamaan (8), (9), (10), (11) secara berurutan dengan hasil seperti yang ditampilkan pada Tabel 5.

Menghitung nilai Kp, Ki, dan Kd menggunakan Persamaan (3), (4) dan (5). Dalam penelitian ini, penulis menentukan *settling time* 2 kali lebih lama, sehingga perhitungan parameter kontroler PID ditunjukkan pada Table 4. berikut ini:

Tabel 4. Hasil Perhitungan Parameter PID (Sumber : Data Primer, 2018)

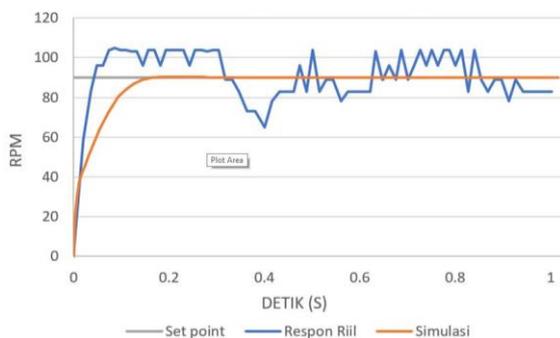
Parameter	Nilai
Kp	0,94624747
Ki	51,4023958
Kd	0,01941504

Dengan memasukkan nilai Kp, Ki dan Kd pada *plant mini conveyor* maka akan didapati grafik respon riil fungsi alih *setpoint* 90 pada tegangan 12 V seperti ditunjukkan pada Gambar 13 berikut.



Gambar 13. Grafik Respon Riil Fungsi Alih 12V dengan PID. (Sumber: Data Primer, 2018)

Perbandingan respon dari simulasi dengan respon sistem (riil) ditunjukkan pada Gambar 14.



Gambar 14 Grafik Perbandingan Respon Simulasi dengan Respon Riil pada Fungsi Alih 12V. (Sumber: Data Primer, 2018)

Gambar 14 menunjukkan respon sistem pada pengujian fungsi alih 12V dengan PID. Perhitungan parameter respon sistem dengan  $\tau = 0,02058345$  detik dapat ditunjukkan dalam bentuk tabel pada Tabel 5 berikut,

Tabel 5. Parameter Respon Sistem Riil. (Sumber : Data Primer, 2018)

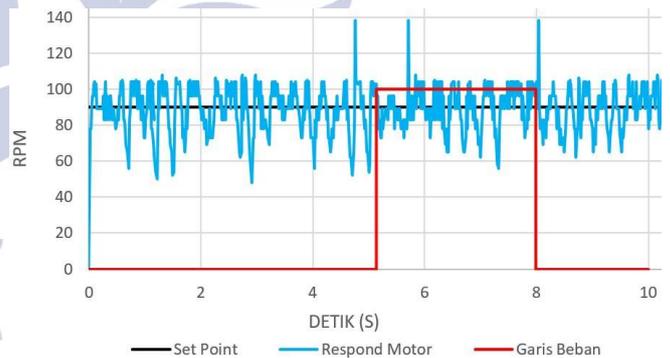
Parameter	Tanpa PID (detik)	Dengan PID (detik)
Waktu tunak (5%)	0,055225	0,061750
Waktu naik (10-90%)	0,040447	0,045226
Waktu tunda ( $\tau_d$ )	0,012759	0,014267
<i>Error Steady State</i>	47,16 %	1,02 %

### 3) Hasil Pengujian Kontrol PID

Setelah mendapatkan nilai Kp, Ki dan Kd, selanjutnya adalah mengimplementasikan parameter tersebut ke dalam kontroler PID. Pengujian dilakukan 3 kali dengan beban 1Kg, 2Kg dan 3Kg.

#### 1. Pengujian dengan beban 1 Kg.

Pengujian dilakukan dengan tanpa meletakkan beban 1 Kg berupa besi pejal pada detik ke-5 sampai detik ke-8 di atas *belt conveyor*. Respon sistem yang dihasilkan ditunjukkan pada Gambar 15 berikut:



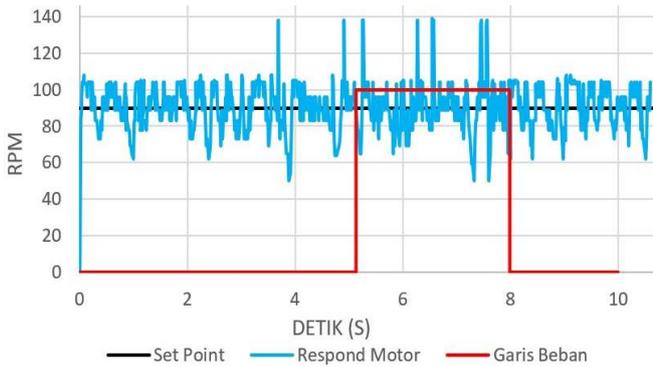
Gambar 15 Respon Sistem Pengujian Beban 1Kg (Sumber: Data Primer, 2018)

Gambar 15 di atas menunjukkan respon sistem pada pengujian fungsi alih PID 12V dengan beban 1 Kg. Perhitungan parameter respon sistem dengan  $\tau = 0,01179991$  detik ditunjukkan dalam bentuk tabel pada Tabel 6.

#### 2. Pengujian dengan Beban 2 Kg.

Pengujian dilakukan dengan tanpa meletakkan beban 2 Kg berupa besi pejal pada detik ke-5 sampai detik ke-8 di atas *belt*

conveyor. Respon sistem yang dihasilkan ditunjukkan pada Gambar 16 berikut:

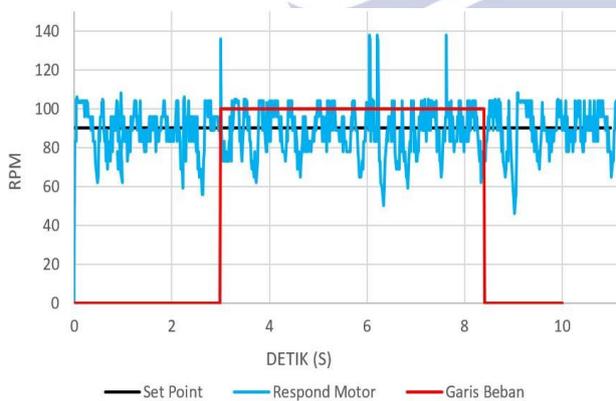


Gambar 16. Respon Sistem Pengujian Beban 2Kg (Sumber: Data Primer, 2018)

Gambar 16 menunjukkan respon sistem pada pengujian fungsi alih PID 12V dengan beban 2 Kg. Perhitungan parameter respon sistem dengan  $\tau = 0,01289025$  detik ditampilkan pada Tabel 6 .

**3. Pengujian dengan Beban 3 Kg.**

Pengujian dilakukan dengan tanpa meletakkan beban 3 Kg berupa besi pejal pada detik ke-3 sampai detik ke-8,4 di atas belt conveyor. Respon sistem yang dihasilkan ditunjukkan pada Gambar 17 berikut:



Gambar 17. Respon Sistem Pengujian Beban 3Kg (Sumber: Data Primer, 2018)

Gambar 17 di atas menunjukkan respon sistem pada pengujian fungsi alih PID 12V dengan beban 2 Kg. Perhitungan parameter respon sistem dengan  $\tau = 0.01170323$  detik ditunjukkan pada Tabel 6 berikut ini:

Tabel 6. Parameter Pengujian Respon Sistem (Sumber: Data Primer, 2018)

Parameter	Beban 1Kg (detik)	Beban 2Kg (detik)	Beban 3Kg (detik)
Waktu tunak (5%)	0,035399	0,038671	0,035109
Waktu naik (10%-90%)	0,025927	0,028323	0,025714
Waktu tunda ( $\tau$ )	0,008179	0,008935	0,008112
Error Steady State	1,133211 %	2,20207 %	0,60197 %

**PENUTUP**

**Simpulan**

Berdasarkan penelitian dan pengujian yang dilakukan, maka didapatkan simpulan bahwa dalam mengatur kecepatan motor bisa menggunakan metode *analytic* dimana kita mencari respon riil motor lalu menghitung nilai  $K_p$ ,  $K_i$  dan  $K_d$ . Merujuk dari Tabel 4 dalam penelitian ini didapati nilai  $K_p=0.94624747$ ,  $K_i=51.4023958$  dan  $K_d=0.01941504$ .

Mikrokontroler yang digunakan untuk mengaplikasikan PID pada Penelitian ini adalah Arduino Uno R3. Mikrokontroler tersebut dihubungkan dengan sensor *Rotary Encoder* agar bisa mengatur kecepatan pada *Conveyor*.

Merujuk dari Tabel 5 Kontroler PID yang dirancang untuk mengendalikan respon kecepatan motor DC pada penelitian ini berhasil mencapai nilai *setpoint* dan mengurangi *Ess (Error Steady State)* pada respon sistem riil dari 47,16% menjadi 1,02% (tanpa beban) dan 2,2020751% (menggunakan beban besi timbangan).

**Saran**

Berdasarkan hasil penelitian yang telah dilakukan terdapat beberapa aspek yang dapat diperbaiki dan dikembangkan agar memberikan hasil yang lebih maksimal, salah satunya menggunakan bahan yang lebih berkualitas dan presisi agar bisa mendapatkan hasil yang optimal, menggunakan motor dan gearbox yang lebih besar agar dapat memiliki range tenaga yang lebar, menggunakan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini, menggunakan sensor *Rotary Encoder* dan *Optocoupler* yang lebih akurat dan presisi agar respon kecepatan yang didapat lebih baik

Sistem ini masih dapat dikembangkan dengan menggunakan metode kontroler yang lain seperti kontroler PI, PD, *fuzzy logic*, *model predictive control (MPC)*, dan *Linear Quadratic Regulator control (LQR)* untuk mendapatkan perbandingan hasil respon kecepatan yang lebih baik.

## DAFTAR PUSTAKA

- Ali, Muhammad. 2004. *Pembelajaran Perancangan sistem kontrol PID dengan software MATLAB*. Yogyakarta: Universitas Negeri Yogyakarta
- Ananda, Stephanus Antonius; Soewangsa, Edhi Tanaka. 2003. *Studi Karakteristik Motor DC Penguat Luar Terhadap Posisi Sikat*. Surabaya : Universitas Kristen Petra.
- Creswell, John W. 2003. *Desain Penelitian, Pendekatan Kualitatif & Kuantitatif*. Depok : Universitas Indonesia
- Doren, V.V.,. 2009. *Sorting Out PID Controller Differences*. Control Engineering.
- Gunterus, Frans.1994. *Falsafah Dasar : Sistem Pengendalian Proses*. Jakarta: PT. Elex Media Komputindo
- Maulana, Achmad Rizal., dkk 2018. *Desain Sistem Pengendalian Kecepatan Motor DC pada Rancang Bangun Mini Konveyor Berbasis Fuzzy Logic Controller*. Surabaya: Universitas Negeri Surabaya.
- Ogata, Katsuhiko. 2002. *Teknik Kontrol Otomatik*. Jakarta: Penerbit Erlangga.
- Pratama, Fauzi Yuda. 2018. *Rancang Bangun Pengendalian Kecepatan Brushless DC Motor Tipe A2212/10T 1400KV Menggunakan Kontroller PID Berbasis Labview*. Surabaya: Universitas Negeri Surabaya.
- Prinandika, Eka Bayu. 2014. *Sistem Pengaturan Kecepatan Motor pada Robot Line Follower Berbeban Menggunakan Kontroler PID*.
- Salamena, Vicky. 2012. *Simulasi Karakteristik Arus dan Kecepatan Motor DC terhadap Masukan Penyearah Gelombang Penuh di Simulink-MATLAB*. Ambon : Universitas Pattimura.
- UNESA. 2000. *Pedoman Penulisan Artikel Jurnal*, Surabaya: Lembaga Penelitian Universitas Negeri Surabaya.
- Wardana, I Nyoman Kusuma. 2015. *Teknik Antarmuka MATLAB dan Arduino*. Denpasar: Vaikutha International Publication
- [https://www.arduino.cc/en/Main/ Products](https://www.arduino.cc/en/Main/Products).  
Diakses pada 15 Mei 2017.
- <http://www.ni.com/white-paper/14805/en/>.  
Diakses pada 13 Mei 2017.