

Kendali Kecepatan Motor DC Dengan Kontroler PID dan Antarmuka Visual Basic

Reza Muhandian¹, Krismadinata²

¹Teknik Elektro 1, ²Universitas Negeri Padang 2

rezamuhardian.rm@gmail.com

Abstrak

Kecepatan motor DC sering mengalami penurunan akibat dari beban yang ada, sehingga kecepatannya menjadi tidak konstan sehingga diperlukan rancangan kontroler. Dalam artikel ini, mengimplementasikan suatu alat pengendali kecepatan motor DC dengan kontroler PID dan antarmuka *visual basic* berbasis *Arduino* UNO. Pada alat ini *setpoint* dan parameter PID diinputkan melalui *visual basic*. Sensor *encoder* akan membaca kecepatan motor dan kontroler PID akan membandingkan nilai yang terbaca dengan *setpoint* dan selisih tersebut dinamakan *error*. Nilai *error* akan dimasukkan ke persamaan PID dan keluaran dari PID berupa PWM (*pulse width modulation*) yang menjadi masukan untuk *driver* motor L298N melalui mikrokontroler *Arduino* UNO. Selanjutnya terjadi pengulangan proses sampai mencapai nilai *setpoint* yang ditentukan dan kecepatan akan dipertahankan. Penentuan hasil parameter kontroler PID ini didapatkan dengan penalaan *trial and error*. *Trial and error* dilakukan dengan cara meinputkan nilai K_p , K_i , K_d dan melihat kurva respon sistem sampai didapatkan bentuk kurva respon yang diinginkan. Dengan menerapkan kontroler PID kecepatan putaran motor DC dapat tercapai dan stabil sesuai dengan nilai *setpoint* yang ditetapkan.

Abstract

The speed of a DC motor often decreases due to the load, so the speed is not constant so a controller design is needed. In this article, implementing a DC motor speed controller with a PID controller and an Arduino UNO-based visual basic interface. In this tool setpoint and PID parameters are input via visual basic. The encoder sensor will read the motor speed and the PID controller will compare the read value with the setpoint and the difference is called an error. The error value will be entered into the PID equation and the output of the PID is a PWM (pulse width modulation) which is input for the L298N motor driver via the Arduino UNO microcontroller. Then the process is repeated until it reaches the specified setpoint value and the speed will be maintained. The determination of the results of the PID controller parameters is obtained by trial and error tuning. Trial and error is done by inputting the values of K_p , K_i , K_d and looking at the response curve of the system to get the desired response curve shape. By implementing a PID controller the rotation speed of a DC motor can be achieved and is stable in accordance with the setpoint value specified..

Keywords: *ArduinoUno, Visual Basic, Encoder, Motor DC, PID,*

PENDAHULUAN

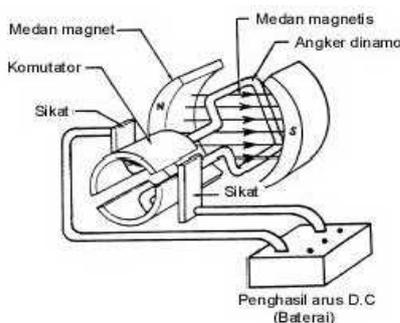
Motor DC (*direct current*) merupakan perangkat yang dapat mengubah energi listrik menjadi energi mekanik [1]. Penggunaan motor DC dalam dunia industri sangat umum digunakan maupun dalam piranti pendukung sistem *instrument elektronik*[2]. Tetapi, kestabilan kecepatan motor DC sulit untuk dikendalikan [3].

Sistem pengendalian kecepatan motor DC diperlukan untuk mengatasi hal ini agar motor dapat berputar stabil sesuai dengan kecepatan yang diinginkan [4]. Salah satunya menggunakan kontroler PID (*proportional-integral-derivative*), dimana kontrol ini parameter settingannya dapat atur sehingga mendapatkan respon system yang diinginkan yaitu kecepatan dengan stabilitas yang baik dengan tingkat error serta overshoot yang kecil sesuai dengan nilai setpoint yang ditentukan [5].

Dalam artikel ini, sistem kontrol PID menggunakan *feedback encoder* sebagai umpan balik yang berbasis Mikrokontroler Arduino UNO dan pengaplikasiannya pada motor DC tanpa beban dengan menggunakan aplikasi *visual studio* sebagai antarmuka yang berfungsi untuk menginputkan nilai *set point* dan parameter PID serta menyajikan data dalam bentuk grafik secara *real time*.

A. Motor DC

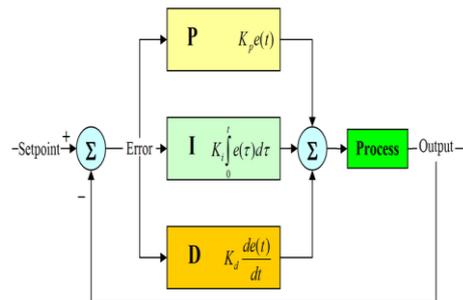
Perangkat yang dapat mengkonversi tenaga listrik menjadi tenaga mekanik dinamakan dengan motor DC. Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya [6].



Gambar 1. Kontruksi Motor DC

B. Kontroler PID

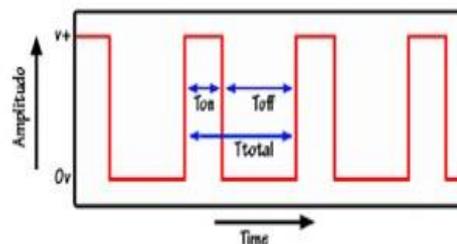
Kontroler PID (*proportional-integral-derivative*) merupakan kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri [7]. Sebuah kontroler PID secara kontinyu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur [8]. Kontroler mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan *variabel kontrol*, seperti posisi keran kontrol, damper, atau daya pada elemen pemanas, ke nilai baru [9].



Gambar 2. Diagram Blok Kontroller PID

C. PWM (pulse width modulation)

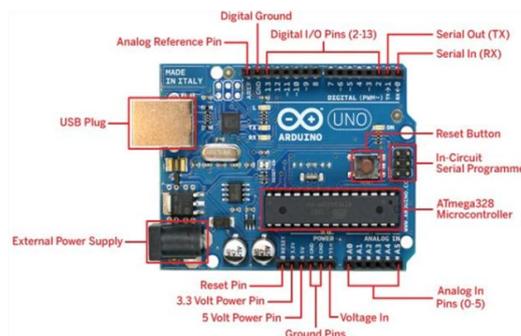
Modulasi lebar pulsa adalah suatu sinyal digital yang berbentuk gelombang kotak (*square wave*) yang lebarnya dapat diatur sesuai kebutuhan dengan mengubah *duty cycle*-nya [10].



Gambar 3. Sinyal PWM (pulse width modulation)

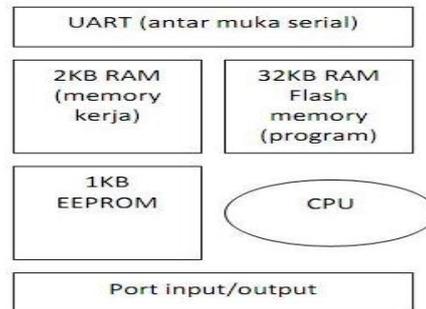
D. Arduino Uno

Arduino merupakan sebuah mikrokontroller Untuk memprogram dan menjalankan arduino kita memerlukan software arduino IDE untuk mengupload program tersebut ke arduino [11]. *Board* Arduino menggunakan *Chip/ IC* mikrokontroler dimana pada tugas akhir ini menggunakan ATmega328. Operasi berbasis waktu dapat dilaksanakan dengan tepat dikarenakan terkandungnya mikroprosesor serta kelengkapan dengan *oscillator* 16MHz serta *regulator* atau *supply* sebesar 5 volt.



Gambar 4. Arduino Uno

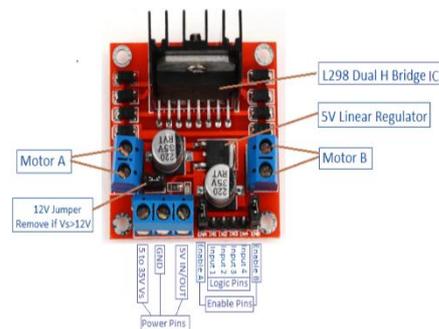
Berikut adalah blok diagram memori arduino uno



Gambar 5. Diagram Blok Memori Arduino Uno

E. Driver Motor L298N

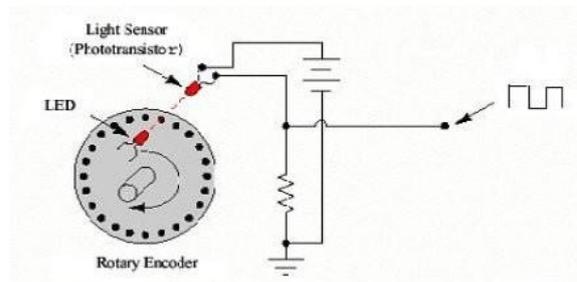
Driver motor L298N merupakan *driver* motor duah H *bridge* yang dapat mengoperasikan 2 buah motor sekaligus, pada dasarnya driver motor mempunyai fungsi yang sama dengan saklar. *Driver* L298N membutuhkan supply 12 volt dan 5 volt dimana kecepatan motor dapat diatur dengan *logic high low* dan modulasi lebar pulsa (PWM) [12].



Gambar 6. Driver Motor L298N [10]

F. Rotary Encoder

Rotary encoder terdiri dari komponen *infrared* sebagai *transmitter* atau pengirim dan phototransistor sebagai *receiver* atau penerima. Prinsip kerjanya cahaya *infrared* melewati lubang piringan dan phototransistor menerima cahaya dari *infrared* sehingga menghasilkan sinyal pulsa yang akan oleh oleh mikrokonroller.



Gambar 7. Prinsip Kerja Rotary Encoder [11]

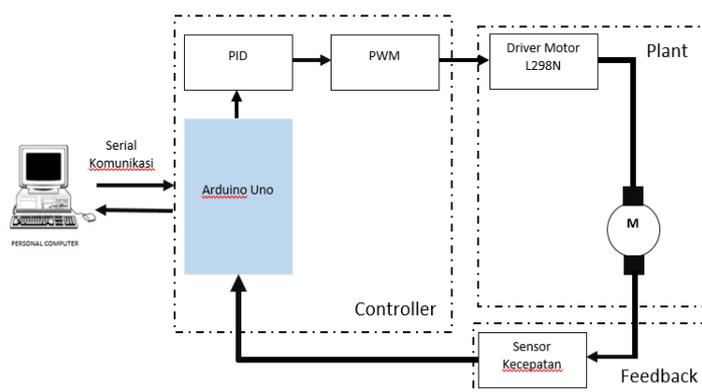
G. Visual Basic

Visual Basic merupakan program yang akan berjalan jika pemakai memberikan respon dengan mengklik atau menekan mouse dan memulai yang ada pada tampilan layar VB tersebut. Bahasa pemrograman dari *Visual Basic* adalah pemrograman *windows* yang berbasis GUI (*Graphical User Interface*) [13]. Sifat bahasa pemrogramannya adalah *Event driven*. Saat *event* terjadi maka kode yang berhubungan dengan *event* akan dijalankan. *Visual Basic* mempunyai suatu jendela yang luas sebagai ruang kerjanya. Pembuatan aplikasi dalam *Visual Basic* kita harus memulai dengan memikirkan kebutuhan, merancang skema dan dilanjutkan dengan *coding* untuk program tersebut [14].

METODE PENELITIAN

A. Blok Diagram

Blok Diagram merupakan salah satu bagian terpenting dalam sebuah perancangan komponen elektronik, dengan adanya diagram blok dapat memberikan kemudahan dalam mengetahui prinsip kerja sebuah alat secara keseluruhan dan juga memberikan kemudahan dalam mengetahui sebuah kesalahan pada alat dengan melakukan pengecekan pada bagian blok diagram.

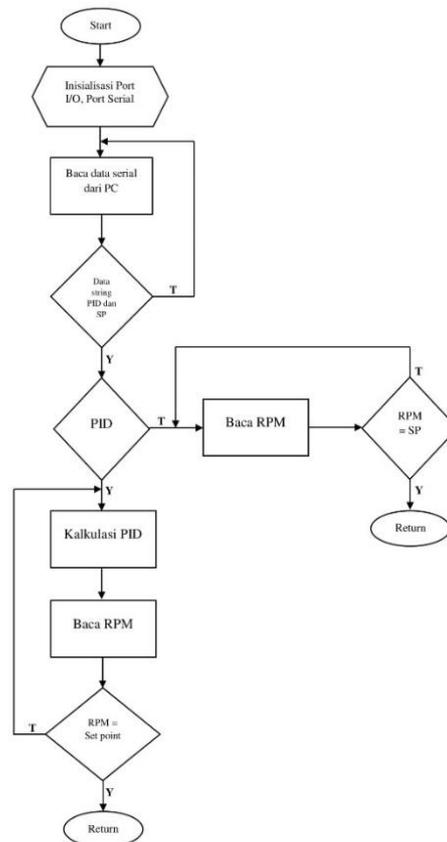


Gambar 8. Blok Diagram Alat

Berikut ini fungsi setiap blok komponen diatas:

Prinsip kerja system secara garis besar adalah mengendalikan kecepatan motor DC dengan controller PID dan ditampilkan secara real time pada *personal computer* dalam bentuk grafik kecepatan motor DC menggunakan aplikasi *visual basic*. Pada system ini nilai set point (kecepatan motor DC dalam satuan RPM) serta parameter-parameter PID diinputkan melalui aplikasi *visual basic*, yang kemudian akan di kirimkan ke arduino melalui komunikasi serial. Sensor kecepatan akan membaca kecepatan motor yang kemudian akan di umpan balik. Kemudian terjadi perbandingan antara kecepatan yang terbaca dengan nilai *set point*, proses ini disebut dengan *error*. Perbedaan antara nilai kecepatan yang terbaca dengan nilai set point dimasukan ke dalam persamaan PID. Keluaran dari pengendali PID digunakan sebagai *duty cycle* pada PWM. Semakin besar *error* yang terjadi, semakin besar *duty cycle* yang dihasilkan. PWM akan mengeluarkan pulsa-pulsa sesuai nilai *duty cycle* keluaran dari pengendali PID. Sinyal pulsa ini digunakan untuk menswitching transistor pada *driver* motor. Nilai PWM yang berubah-ubah menyebabkan tegangan rata-rata yang menyuplai motor juga berubah-ubah. Bila kecepatan yang terbaca sudah sesuai dengan nilai setpoint maka kontroller PID akan mempertahankan kecepatan tersebut.

B. Flowchart Sistem



Gambar 9. Flowchart Alat

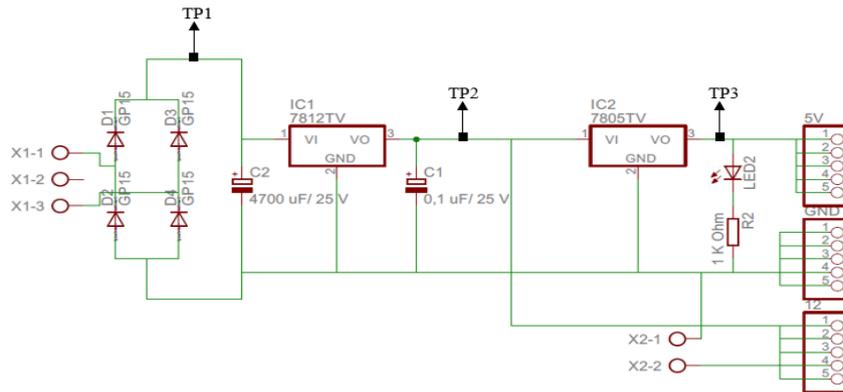
HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk memperoleh data-data setiap bagian alat bekerja dengan baik

A. Pengujian Catu Daya

Pengujian ini dilakukan dengan cara mengukur tegangan keluaran setelah penyearah jembatan, keluaran *regulator* LM7805 dan regulator LM7812. Tegangan yang dihasilkan

catu daya akan menjadi input untuk modul *driver* motor L298N.



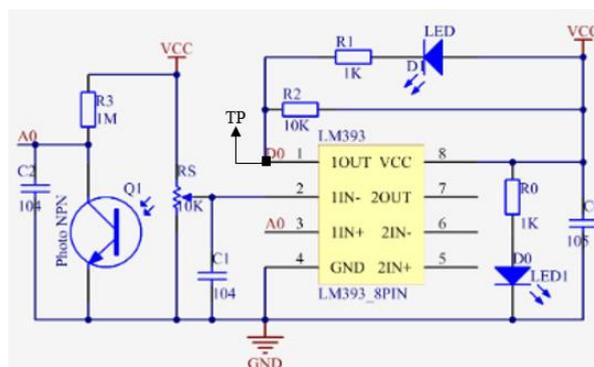
Gambar 10. Titik Pengukuran Catu Daya

Tabel 1. Pengujian Catu Daya

No	Titik Pengukuran (TP)	Tegangan
1	TP 1	13,6 volt
2	TP 2	4,98 volt
3	TP 3	12,11 volt

B. Pengujian Rangkaian Sensor Kecepatan

Dalam pengujian sensor kecepatan diukur tegangan keluaran sensor saat mendeteksi cahaya



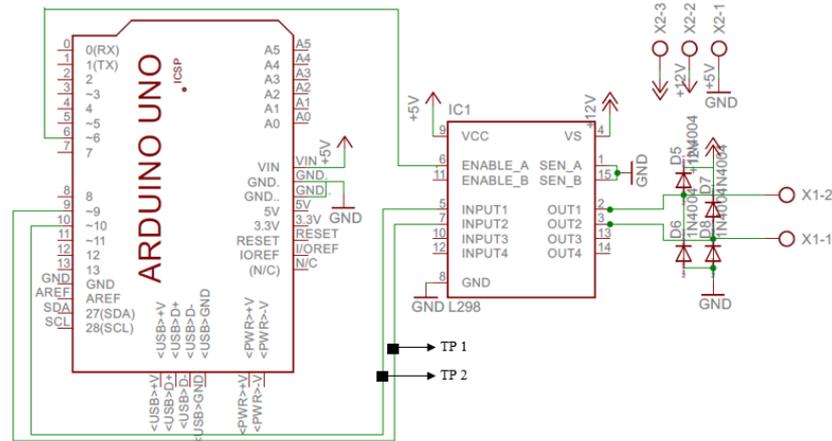
Gambar 11. Rangkaian Pengujian Sensor Kecepatan

Tabel 2. Hasil Pengujian dan Pengukuran Rangkaian Sensor Kecepatan

Kondisi	Tegangan
Saat mendeteksi infarled	152,1 mV
Saat tak mendeteksi infraled	4,74

C. Pengujian Driver Motor L298N

Pengujian *driver* motor l298n dilakukan dengan cara mengukur keluaran dari pin *logic* arduino yang terhubung ke modul l298n, dimana pada pengujian ini pin *logic* arduino menggunakan pin 9 dan pin 10 disaat motor dalam kondisi berputar.



Gambar 12. Rangkaian Pengukuran Driver Motor L298N

Table 3. Hasil Pengujian Driver Motor L298N

Titik Pengukuran (TP)	Tegangan
TP 1	4,74 volt
TP 2	0,01 volt

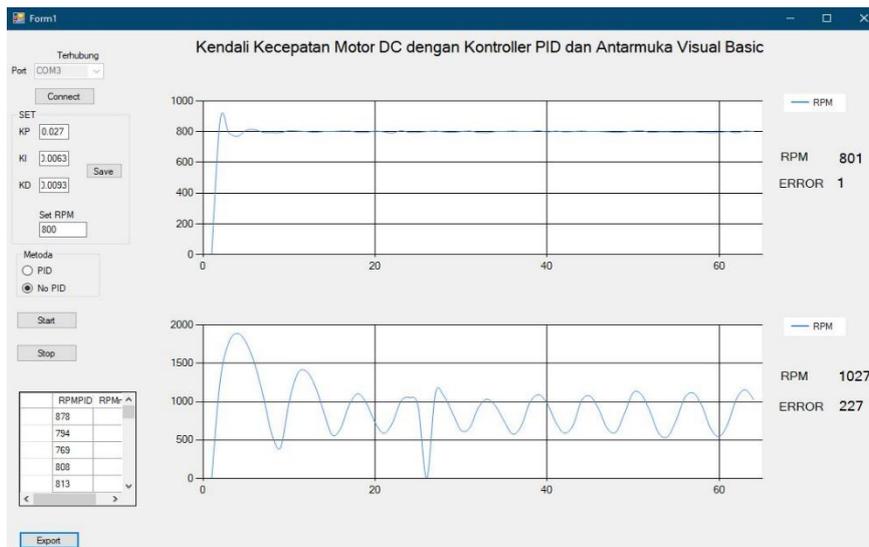
D. Pengujian Kontroler PID (*propotional-integral-derivative*)

Pada pengujian controller PID dilakukan dengan cara memasukkan nilai setpoint (rpm) dan parameter PID. Pada pengujian ini nilai setpoint ditetapkan dari 700 rpm -1200 rpm dan nilai $K_p = 0,027$; $K_i = 0,0063$; dan $K_d = 0,0093$ dimana nilai tersebut didapatkan dengan pendekatan trial and error.

Table 4. Hasil Pengukuran Respon Motor

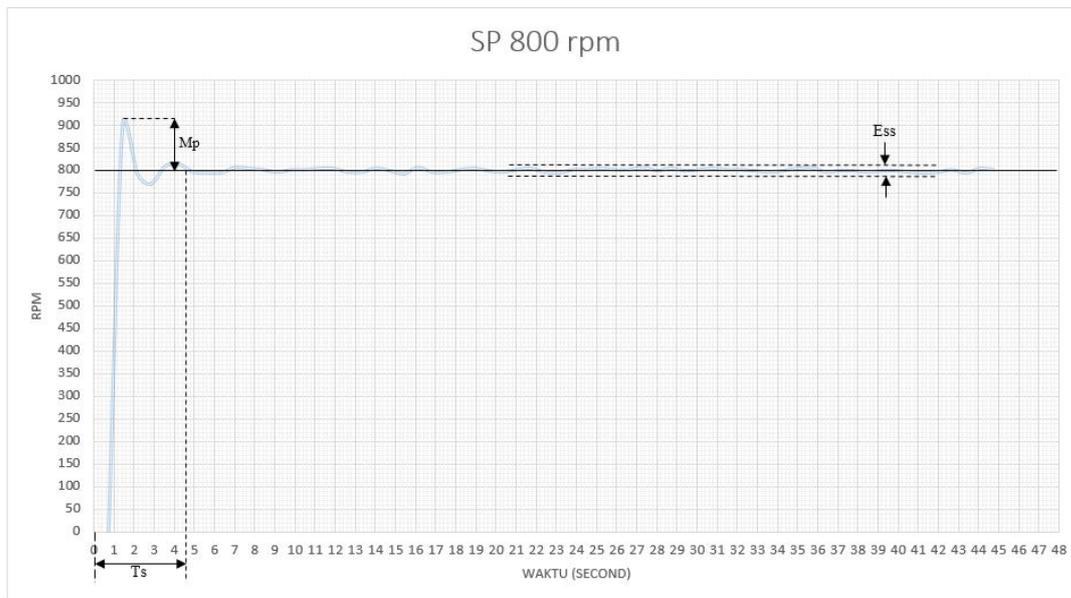
Setpoint (rpm)	Duty Cycle	Overshoot	Time Settling	Error Steady State
700	8,19 %	14,8 %	5,3 detik	2,4 %
750	8,58 %	13,6 %	5,2 detik	3,2 %
800	8,67 %	14,37 %	5,2 detik	2,5 %
850	8,67 %	14,2 %	5,1 detik	2,5 %
900	8,97 %	13,4 %	5,2 detik	2,4 %
950	8,97 %	12,2 %	5,2 detik	2,4 %
1000	9,36 %	11,8 %	4,8 detik	2,5 %
1050	9,37 %	10,4 %	4,7 detik	2,5 %
1100	9,37 %	8,52 %	4,7 detik	2,4 %
1150	9,75 %	8,32 %	4,5 detik	2,5 %
1200	10,15 %	8,12 %	4,6 detik	2,5 %

Untuk menganalisa data yang didapatkan, diambil salah satu nilai setpoint yaitu 800 rpm



Gambar 13. Tampilan Kecepatan Motor Pada Visual Basic

Untuk memudahkan menganalisa kurva respon sistem diatas, maka grafik diubah kedalam bentuk grafik excel sebagai berikut.



Gambar 14. Kurva Respon Kecepatan Motor dengan PID

Dari gambar di atas didapatkan nilai time settling atau t_s , *error steady state* atau ess dan maksimum *overshoot* atau mp beserta perhitungannya

- 1) T_s (*time settling*) yaitu waktu yang diperlukan kurva respon untuk mencapai dan menetap didaerah setpoint yang telah ditentukan sebelumnya. Pada *set point* 800 rpm didapatkan $T_s = 4,5s$
- 2) *Error steady state* yaitu selisih antara nilai keluaran dengan nilai masukan saat keadaan steady state.

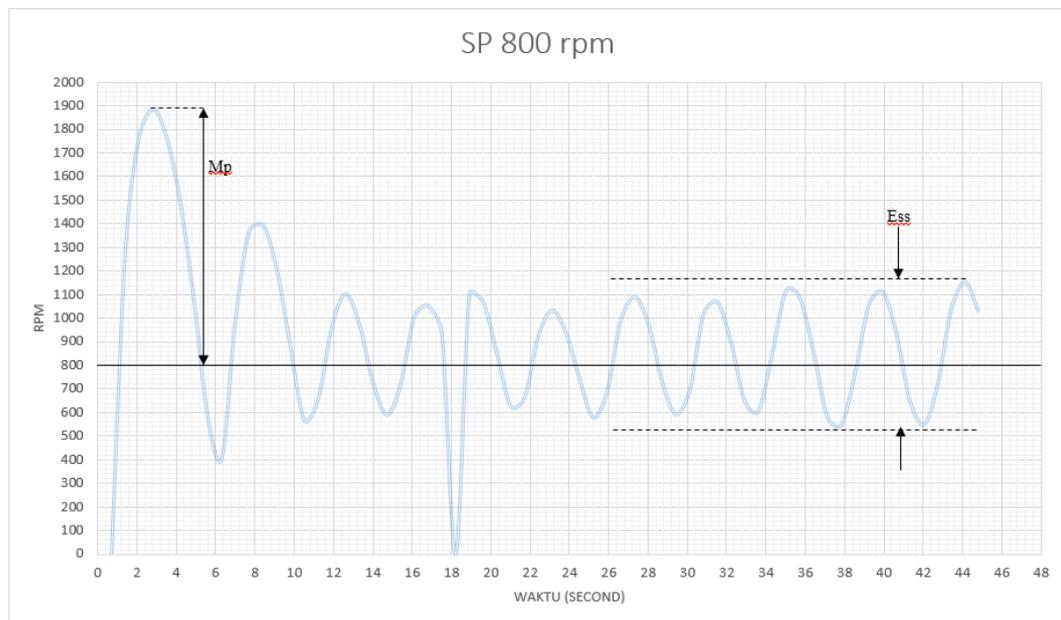
$$ess = \frac{(810-790)}{800} \times 100\% = 2,5\%$$

3) Maksimum *overshoot* (MP) merupakan nilai tertinggi dari kurva respon. Nilai tertinggi dari kurva respon adalah 915 rpm.

$$MP = \frac{(915-800)}{800} \times 100\%$$

$$= 14,37\%$$

Berikut adalah analisa jika tanpa menggunakan controller PID.



Gambar 15. Kurva Respon Kecepatan Motor tanpa PID

Gambar diatas merupakan bentuk kurva respon kecepatan motor tanpa menggunakan kontroller PID. Dapat dilihat nilai Mp (maksimum *overshoot*) sangat tinggi serta ess (*error steady state*) juga sangat besar hal ini dikarenakan *error* pada sistem diabaikan. Berikut adalah nilai ess dan Mp dari gambar diatas

$$1) \quad ess = \frac{(1180-800)}{800} \times 100\%$$

$$= 82,5\%$$

$$2) \quad MP = \frac{(1900-800)}{800} \times 100\%$$

$$= 137,5\%$$

PENUTUP

Berdasarkan hasil pengujian dan analisa pada pembuatan Sistem Kendali Kecepatan Motor DC dengan Metoda PID dan Antarmuka *Visual Basic* Berbasis Mikrokontroller maka dapat simpulkan bahwa:

1. Tuning parameter PID dilakukan dengan pendekatan *trial and error* dan didapatkan hasil $K_p = 0,027$; $K_i = 0,0063$; dan $K_d = 0,0093$.

2. Nilai *setpoint* ditetapkan pada alat ini yaitu dari rentang 700 rpm – 1200 rpm ini dikarenakan keterbatasan kemampuan pembacaan sensor.
3. Dari nilai tuning yang didapatkan hasil didapatkan nilai *overshoot* dengan rata-rata 11,79 %, *time settling* dengan rata-rata 4,48% dan *error steady state* 2,5%.
4. Dengan menerapkan kontroler PID pada sistem kendali kecepatan motor DC dapat mempercepat respon sistem terhadap perubahan sinyal input (*set point*) dan memperkecil sinyal kesalahan (*error*).
5. Untuk mengoperasikan PID kontroler sangatlah mudah, hanya melakukan tuning parameternya dengan cara mencoba-coba (*trial and error*) untuk menghasilkan sistem kontrol yang diinginkan.
6. Pengendalian kecepatan motor DC tanpa menggunakan controller didapatkan nilai *overshoot* dan *error steady state* yang besar, dan sistem akan terus berosilasi.

DAFTAR PUSTAKA

- [1] Mardianto, Eric.(2018). *Rancang Bangun Antarmuka Operasi Motor DC Dengan Converter 4 Kuadran*. Tugas Akhir. Padang : Universitas Negeri Padang.
- [2] Simanullang, Sando Andre, Paula Santi Rudati, and Feriyonika Feriyonika. 2017 "Sistem PID Pengendali Level Ketinggian Air Berbasis Modbus/TCP-LCU dan Industrial Field Control Node-RTU." *Prosiding Industrial Research Workshop and National Seminar*. Vol. 8.
- [3] Rosid Dhanang. 2012. Pengendalian Kecepatan Motor Arus Searah Seri dengan DC Chopper. Tugas Akhir. Depok : Universitas Indonesia
- [4] Silalahi, Dony Araventa.2017. Pengendalian Kecepatan Putar (Rpm) Motor DC dengan Metode PID Berbasis Mikrokontroler Atmega328 Menggunakan Bahasa Pemrograman Code Vision Avr. Tugas Akhir. Medan : Universitas Sumatera Utara
- [5] Haryanto, Heri, and Sarif Hidayat. (2016)"Perancangan HMI (Human Machine Interface) Untuk Pengendalian Kecepatan Motor DC." *Setrum: Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer* 1.2: 58-65.
- [6] Susanto, Erwin. Kontrol proporsional integral derivatif (PID) untuk motor dc menggunakan personal computer. Departemen Teknik Elektro, Institut Teknologi Telkom Bandung.
- [7] Husnaini, Irma dan Krismadinata. Komparasi Pengendali PI dan PID Untuk Tegangan Keluaran Konverter Buck, (*Jurnal Nasional Teknik Elektro*), ISSN: 2302 – 2949, Vol. 6, No. 3, November 2017. Padang : Universitas Negeri Padang.
- [8] Setiawan, Muhammad Rizki, Muhammad Aziz Muslim, and Goegoes Dwi Nusantoro. "Kontrol Kecepatan Motor DC Dengan Metode PID Menggunakan Visual Basic 6.0 dan Mikrokontroler ATmega 16." *Jurnal Mahasiswa TEUB* 1.2 (2013).

-
- [9] Nasrul, Z. A., Yonita Putri Roja, and Novi Sylvia. 2019 "Aplikasi Kontrol PID pada Reaktor Pabrik Asam Formiat dengan Kapasitas 100.000 Ton/Tahun." *Jurnal Teknologi Kimia Unimal* 7.2: 135-152.
- [10] Retnowati, Ir. *Sistem Pengaturan Kecepatan Putaran Motor Pada Mesin Pemutar Gerabah Menggunakan Kontroler Proporsional Integral Deferenensial (PID) Berbasis Mikrokontroler*. Diss. Brawijaya University.
- [11] Rachmat, Antonius.(2010). *Algoritma daan Pemrograman dengan Bahasa C – Konsep, Teori,& Implementasi*. Yogyakarta: Andi
- [12] Setiawan, Rizki.(2012). *Kontrol Kecepatan Motor DC Dengan Metode PID Menggunakan Visual Basic 6.0 Dan Mikrokontroler ATmega 16*. Jurnal EECCIS Vol. 6, No. 2
- [13] Wiriawan, Alifa Restu Janwar, and Andry Irawan. "Pengaturan Kecepatan Motor DC dengan Kontrol Proporsional Integral Derifatif (PID) Berbasis LabView Motor DC Speed Adjustment By Propotional Integral Derivative (PID) Based on LabView.
- [14] Simanullang, Sando Andre.(2017). "Sistem PID Pengendali Level Ketinggian Air Berbasis Modbus/TCP-LCU dan Industrial Field Control Node-RTU". Politenik Negeri Bandung.

Biodata Penulis

Reza Muhardian, lahir di Padang Panjang, 31 Mei 1995. Sedang menyelesaikan studi di Program Studi Teknik Elektro Industri (DIV), di Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Padang dengan tahun masuk 2017.

Krismadinata, dilahirkan di Padang, 11 September 1977. Lulus dengan gelar Sarjana Teknik dari Universitas Andalas pada tahun 2000. Memperoleh gelar Magister Teknik dari Institut TEKNOLOGI Bandung (ITB) pada tahun 2004, dan S3 di Universitas Malaya pada tahun 2012.