

# Perancangan Sistem Kendali Kecepatan Putaran Mesin Bensin Menggunakan Pengendali PID

## *Control System Design of Gasoline Engine Rotation Speed Using PID Controller*

Yusrila Yeka Kerlooza<sup>1</sup>, Taufiq Nuzwir Nizar<sup>2\*</sup>, Asep Yusril Hidayat<sup>3</sup>

<sup>1,3)</sup> Program Studi Teknik Elektro

<sup>2)</sup> Program Studi Sistem Komputer

Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia

Jl. Dipati Ukur No. 112-116, Bandung, Indonesia 40132

\* Email: [taufiq.nuzwir@email.unikom.ac.id](mailto:taufiq.nuzwir@email.unikom.ac.id)

**Abstrak** – Pada saat ini sistem kontrol yang efektif dan efisien semakin dibutuhkan terutama dalam bidang industri. Pada industri pemilahan sampah sudah menggunakan *conveyor belt* bermesin bensin sebagai alat mobilisasi sampah yang akan disortir. Namun, beban sampah yang bervariasi mengakibatkan *conveyor belt* tidak stabil. Penelitian ini bertujuan untuk mengendalikan kecepatan putaran mesin bensin dengan sensor dan *aktuator* pengendali berdasarkan algoritma pengendali PID. Tujuan selanjutnya adalah menentukan nilai parameter  $K_p$ ,  $K_i$ , dan  $K_d$  yang tepat agar kecepatan dapat mencapai *set point* dengan waktu dan stabilitas terbaik. Penelitian dilakukan pada mesin bensin jenis Stihl PC-75 dengan sensor *hall effect* sebagai sensor kecepatan, sensor potensiometer untuk mengetahui posisi gas, serta *aktuator* pengendali kecepatan. *Aktuator* pengendali kecepatan yaitu motor DC untuk menarik kabel *throttle* gas. Hasil penelitian ini adalah sensor *hall effect* mampu membaca kecepatan putaran mesin sampai 18000 rpm dengan menggunakan *filter moving average* bernilai 25 sampel, dan *aktuator* pengendali gas mampu bekerja. Pengendalian kecepatan putaran mesin bensin dapat dilakukan berdasarkan pengendali PID dengan sensor dan *aktuator* pengendali. Namun, proses *tuning* parameter PID masih menggunakan metode *trial and error*. Kesimpulan penelitian ini adalah nilai parameter pengendali PID terbaik untuk mencapai *set point* 8000 rpm berada pada nilai  $K_p = 0.51$ ,  $K_i = 0.3$ ,  $K_d = 5$  yang membutuhkan 15,5 detik untuk mencapai keadaan stabil dan *error* kestabilan 3,88%.

**Kata kunci:** pengendali PID, mesin bensin, *hall effect*, kecepatan, kestabilan.

**Abstract** - At this time, an effective and efficient control system is increasingly needed, especially in industrial sector. The waste sorting industry already uses a gasoline-engined conveyor belt as a means of mobilizing waste to be sorted. However, the varying load of waste causes the conveyor belt to become unstable. This study aims to control the rotational speed of gasoline engines with sensors and control actuators based on PID algorithm. Next objective is to determine the appropriate  $K_p$ ,  $K_i$ , and  $K_d$  parameter values so that the speed can reach the set point with the best time and stability. The research is conducted on a Stihl PC-75 gasoline engine with hall effect sensor as a speed sensor, potentiometer sensor to determine the gas position, and speed control actuator. Speed control actuator is DC motor that pulls the throttle cable. The results of this study show that hall effect sensor is able to read the engine rotation speed up to 18000 rpm using a moving average filter worth 25 samples and that the gas control actuator is able to work. Controlling the speed of gasoline engine can be done based on the PID controller with sensors and actuators. However, the PID parameter tuning process still uses the trial-and-error method. The conclusion of this study is that the best PID controller parameter values to reach the 8000 rpm set point are  $K_p = 0.51$ ,  $K_i = 0.3$ , and  $K_d = 5$ , which require 15.5 seconds to reach a stable state and a stability error of 3.88%.

**Keywords:** PID controller, gasoline engine, hall effect, speed, stability.

## I. PENDAHULUAN

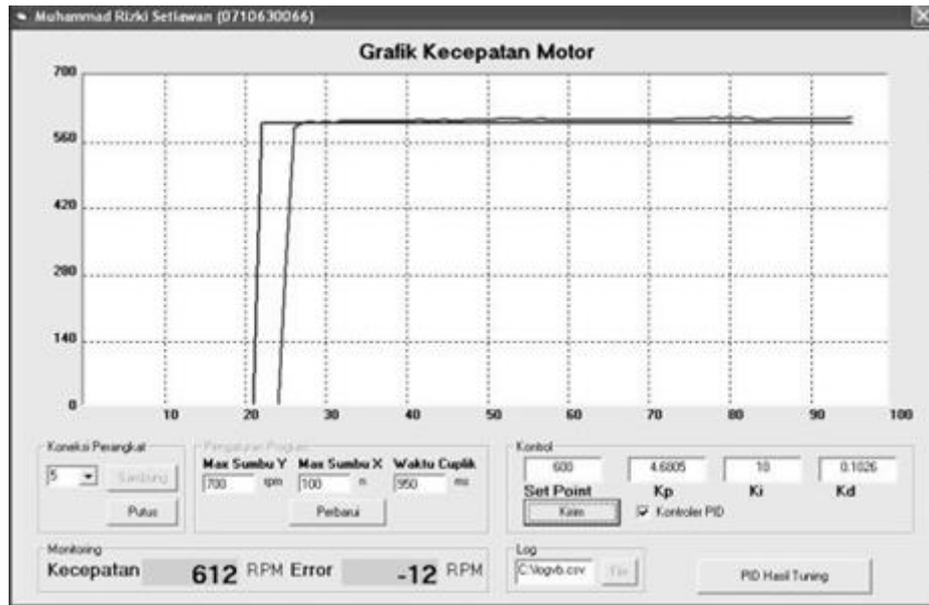
Kebutuhan akan sistem kontrol yang lebih efektif dan efisien di era modern ini semakin

meningkat. Contohnya industri pemilahan sampah sudah menggunakan *conveyor belt* bermesin bensin untuk mensortir sampah [1]. *Conveyor belt*

terhubung antara dua atau lebih katrol yang berputar untuk mengangkut material [2], misalnya sampah. Namun, beban sampah yang berbeda-beda mengakibatkan pergerakan *conveyer belt* tidak stabil [3]. Untuk menstabilkan pergerakan *conveyer belt* bermesin bensin hanya dapat dilakukan mengendalikan tarikan gasnya secara manual. Pada kasus motor DC sudah menerapkan pengendali untuk mengendalikan kecepatan motor DC secara otomatis. Algoritma pengendali yang digunakan adalah algoritma pengendali PID [4]. Pada **Gambar 1** dapat dilihat bahwa respon pengendali PID dapat cepat mencapai *set point* dengan stabilitas yang tinggi.

Sistem kendali kecepatan mesin bensin menggunakan pengendali PID masih relatif sedikit yang meneliti sebelumnya dan merupakan topik penelitian baru. Namun, rujukan penelitian ini tetap berdasarkan penelitian sebelumnya yang masih berhubungan dengan pengendalian kecepatan dan pengendali PID. Penelitian pengendalian kecepatan menggunakan pengendali PID sudah dilakukan oleh Setiawan dkk. [5], namun penelitiannya dilakukan pada motor DC dan hasil penelitian ini berupa keberhasilan menstabilkan kecepatan motor DC sesuai dengan *set point*. Pada penelitian selanjutnya yang dilakukan oleh Triwiyatno dkk. [6], menerapkan pengendali PID pada mesin bensin, tetapi penelitian ini dilakukan pada mesin bensin 4 langkah dan hanya mengendalikan pasokan bahan bakar dengan mengatur rasio perbandingan udara dan bahan bakar. Penelitian selanjutnya yang dilakukan oleh Silalahi [7]. Penelitian ini hampir sama dengan penelitian yang dilakukan oleh Setiawan dkk. Penelitian ini membahas pengaturan kecepatan motor DC menggunakan PID. Namun, pada penelitian ini bahasa pemrograman yang digunakan adalah bahasa pemrograman *code vision* AVR. Selanjutnya, penelitian yang dilakukan oleh Reimansyah dkk. [8]. Penelitian ini mirip dengan penelitian yang dilakukan oleh penulis. Penelitian ini menjelaskan mengenai pengaturan kecepatan mesin bensin dengan mengendalikan kabel *throttle*. Pengendalian kabel *throttle* dilakukan untuk mengendalikan pasokan bahan bakar. Akan tetapi, penelitian ini menggunakan metode logika fuzzy. Hasil dari penelitian ini adalah kecepatan stasioner mampu berada di sekitar *set point*. Penelitian berikutnya dilakukan oleh Yu dkk. [9].

Penelitian ini membahas pengaturan kecepatan pada mesin bensin. Metode yang diterapkan pada penelitian ini adalah *integral-separation fuzzy PID controller*. Penelitian ini dilakukan secara simulasi melalui Simulink *Matrix Laboratory* (MATLAB). Model mesin bensin yang digunakan merupakan mesin bensin tipe GX160. Tujuan dari penelitian ini adalah untuk membandingkan kinerja metode *integral-separation fuzzy PID controller*, *intergral PID controller*, dan pengendali PID tradisional. Hasil dari penelitian ini didapatkan bahwa kinerja metode *integral-separation fuzzy PID controller* lebih baik dibandingkan dengan metode lainnya. Dengan menggunakan metode *integral-separation fuzzy PID controller*, *overshoot* pada sistem berkurang, respon sistem cepat, dan akurasi tinggi. Peningkatan sistem tersebut dapat meningkatkan kehandalan mesin bensin dalam proses pekerjaan mesin. Selanjutnya, Lin dkk. [10] meneliti mengenai pengendalian pasokan bahan bakar pada mesin bensin. Metode yang digunakan pada penelitian ini adalah metode *triple-step*. Metode *triple-step* terdiri atas tiga langkah yakni *steady-state control*, *feed-forward control*, dan *feedback control*. Penelitian ini dilakukan secara simulasi menggunakan aplikasi *en-DYNA engine dynamics*. Metode *triple step* bergantung pada kecepatan mesin bensin, massa jenis udara melalui *throttle*, dan sensor gas oksigen (EGO). Kinerja dari sistem pengendalian pengisian bahan bakar pada mesin bensin dilihat dari beberapa kasus. Hasil dari penelitian ini menunjukkan bahwa kinerja dari sistem tersebut memiliki efisiensi yang baik pada kondisi transien. Selanjutnya adalah penelitian yang dilakukan oleh Anjum dkk. [11]. Penelitian ini membahas penentuan kecepatan dengan manajemen torsi pada mesin bensin. Metode yang digunakan dalam manajemen torsi pada penelitian ini adalah *Smooth Super-Twisting Algorithm* (SSTA). Pada penelitian terdapat dua *control loop* yaitu *air control loop* berupa sudut *throttle* dan *speed control loop* berupa kecepatan aliran bahan bakar. Penelitian ini dilakukan secara simulasi menggunakan Simulink MATLAB. SSTA digunakan untuk mengatur besar torsi agar memperoleh nilai kecepatan mesin yang tepat ketika adanya gangguan. Hasil dari penelitian ini memberikan tingkat efektivitas yang tinggi dalam menentukan kecepatan mesin berdasarkan manajemen torsi.



Gambar 1. Respon Motor DC dengan Pengendali PID [4]

Berdasarkan beberapa jurnal yang telah dijelaskan, pengendalian kecepatan putaran mesin masih dilakukan secara simulasi dan belum diimplementasikan secara langsung [9] [10] [11]. Kemudian, metode yang digunakan seperti *integral-separation fuzzy PID controller* membutuhkan penentuan aturan dan syarat fuzzy terlebih dahulu [9]. Metode lain seperti metode *triple-step* dan metode SSTA dilakukan dengan merancang model sistem melalui diagram blok sistem berdasarkan persamaan matematis yang sesuai [10] [11]. Hal ini menyebabkan sistem untuk mengendalikan kecepatan putaran mesin menjadi rumit dan kompleks.

Tujuan dari penelitian ini adalah menerapkan sistem sensor dan aktuator pengendali yang dapat diterapkan untuk mengendalikan kecepatan putaran mesin bensin menggunakan algoritma pengendali PID. Kemudian, tujuan selanjutnya adalah menentukan nilai besaran-besaran parameter  $K_p$ , parameter  $K_i$ , dan parameter  $K_d$  yang terbaik. Dengan diketahui nilai besaran ketiga parameter yang terbaik, *set point* pada putaran mesin bensin yang dikehendaki dapat dicapai dengan waktu optimal dan stabilitasnya yang terbaik.

## II. METODOLOGI

Pada bab ini akan membahas mengenai sistem, perancangan elektronik, perancangan mekanik, dan perancangan algoritma, serta menjelaskan cara kerja sistem kendali putaran mesin bensin menggunakan metode pengendali PID melalui diagram blok sistem dan *flowchart*. Pada sistem

terdapat sebuah *input* yaitu sensor *hall effect* untuk menghitung kecepatan putaran mesin bensin, dan sensor potensiometer yang digunakan untuk mengetahui posisi penarikan kabel *throttle* gas. Pengolahan data dari sensor akan dijadikan acuan untuk menjalankan algoritma pengendali PID yang akan diproses pada mikrokontroler Arduino Mega 2560. Setelah proses data dilakukan, maka mikrokontroler akan mengirimkan perintah ke motor *driver* dan motor DC akan melakukan penarikan kabel *throttle* gas sesuai dengan yang diperintah. Pada saat motor DC menarik kabel *throttle* gas, sensor potensiometer akan mengirimkan kembali data posisi gas ke mikrokontroler.

Pengendali PID merupakan gabungan dari kendali proporsional, integral dan derivatif [12] [13]. Maksud dari penggabungan tersebut karena apabila masing-masing pengendali berdiri sendiri, ketika terdapat kekurangan pada pengendali, maka pengendali tersebut tidak mampu untuk memperbaiki kekurangan tersebut. Oleh karena itu, penggabungan ketiga elemen kendali akan saling mengisi setiap kekurangan yang dihasilkan. Keunggulan dari pengendali PID ini didasarkan pada masing-masing kendali P, I, dan D yaitu untuk mempercepat reaksi sistem, menghilangkan *offset*, dan membuat sistem menjadi lebih stabil dengan cara meredam osilasi [14] [15].

Pengendali PID akan memberikan aksi berdasarkan *error* yang telah diperoleh, *error* disini merupakan perbedaan dari *set point* yang telah ditentukan. Keluaran dari pengendali PID adalah hasil penjumlahan dari ketiga komponen tersebut [16] [17]. Karakteristik pengendali PID ini

dipengaruhi oleh ketiga komponen kendali P, I, dan D, penyetelan nilai setiap konstanta  $K_p$ ,  $K_i$ , dan  $K_d$  akan mengakibatkan penonjolan sifat dari masing-masing komponen. Ketiganya bisa dipakai bersamaan ataupun sendiri-sendiri tergantung dari reaksi yang kita mau terhadap sesuatu *plant* [18] [19]. Pada perancangan sistem kontrol PID, biasanya dilakukan dengan metode *trial & error*.

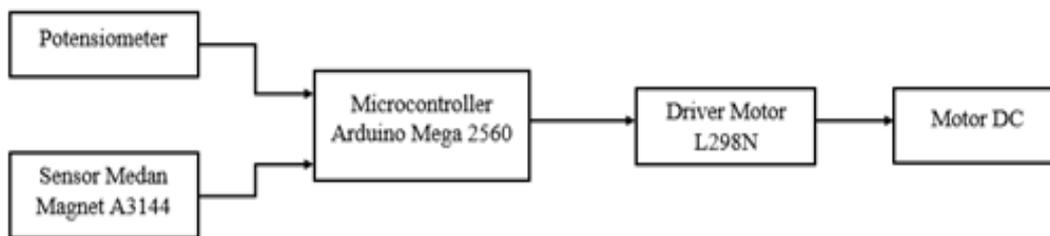
Hal ini disebabkan karena parameter  $K_p$ ,  $K_i$ , dan  $K_d$  tidak *independent*, untuk mendapatkan aksi kontrol yang baik diperlukan langkah *trial & error* dengan kombinasi antara P, I dan D sampai ditemukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  seperti yang diharapkan [20] [21] [22].

Untuk memudahkan pembuatan sebuah sistem, perlu dilakukan perancangan terlebih dahulu agar sistem yang dibuat dapat bekerja dengan baik sesuai yang diharapkan pada saat perancangan. Dalam perancangan bagian *hardware*, dapat dibuat dengan diagram blok, seperti dapat dilihat pada **Gambar 2**.

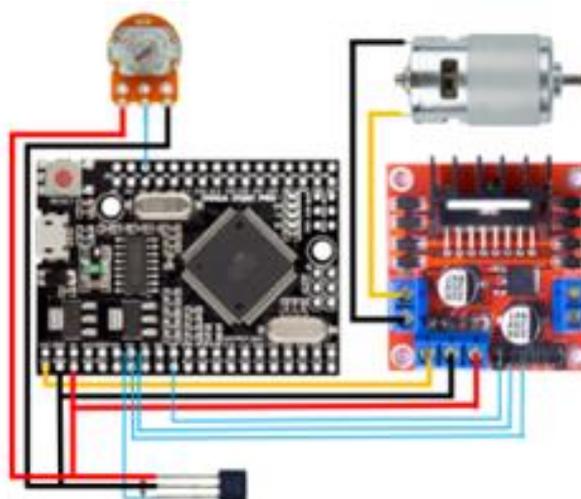
Secara singkat, pada **Gambar 2** terdapat beberapa komponen yang di antaranya ada potensiometer dan sensor medan magnet A3144 yang merupakan masukan dari sistem ini. Kemudian ada mikrokontroler Arduino Mega 2560 yang merupakan bagian proses dari keseluruhan

sistem ini untuk mengolah semua data. Keluaran dari sistem ini yaitu motor *driver* L298N dan motor DC sebagai penarik kabel *throttle* gas sesuai perintah dari mikrokontroler. Perancangan skema rangkaian kendali putaran mesin bensin menggunakan metode pengendali PID ini akan memperlihatkan peralatan elektronik yang akan dibangun. Dalam membuat sebuah alat, alangkah baiknya membuat skema rangkaiannya terlebih dahulu. Karena dengan adanya skema rangkaian, memudahkan untuk mengetahui komponen apa saja yang akan dipakai. Berikut adalah skema rangkaian untuk alat ini, yang dapat ditunjukkan pada **Gambar 3**.

Pada **Gambar 3** dapat dilihat dalam skema rangkaian tersebut beberapa komponen dan modul. Komponen yang dipakai dalam skema rangkaian tersebut adalah 1 buah sensor A3144, 1 buah sensor potensiometer, 1 buah resistor, dan 1 buah motor DC. Modul yang dipakai dalam skema rangkaian di atas adalah 1 buah *board* arduino mega 2560, 1 buah motor *driver* L298N. Setelah semua komponen sudah diketahui dan diletakkan, maka dilanjutkan dengan menghubungkan kaki-kaki komponen agar semua komponen bisa berfungsi sebagaimana mestinya.

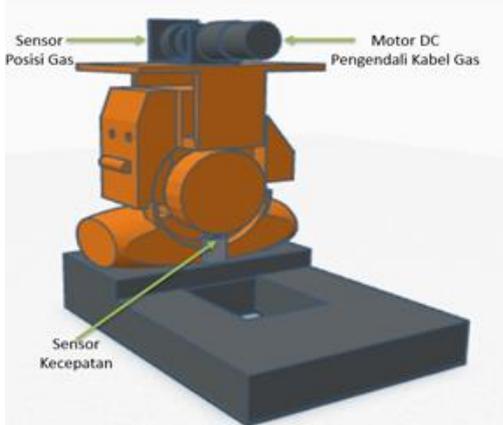


**Gambar 2.** Diagram Blok Sistem



**Gambar 3.** Skema Rangkaian Alat

Untuk membuat alat, perlu adanya desain mekanik terlebih dahulu agar tidak ada kesalahan saat pembuatan, mesin bensin yang digunakan adalah type Stihl PC 75 dan untuk bagian dudukannya terbuat dari besi dengan ukuran 28 cm x 20 cm dan untuk dudukan motor DC dan potensiometer menggunakan bahan PVC. Bentuk desain alatnya dapat ditunjukkan pada **Gambar 4**.



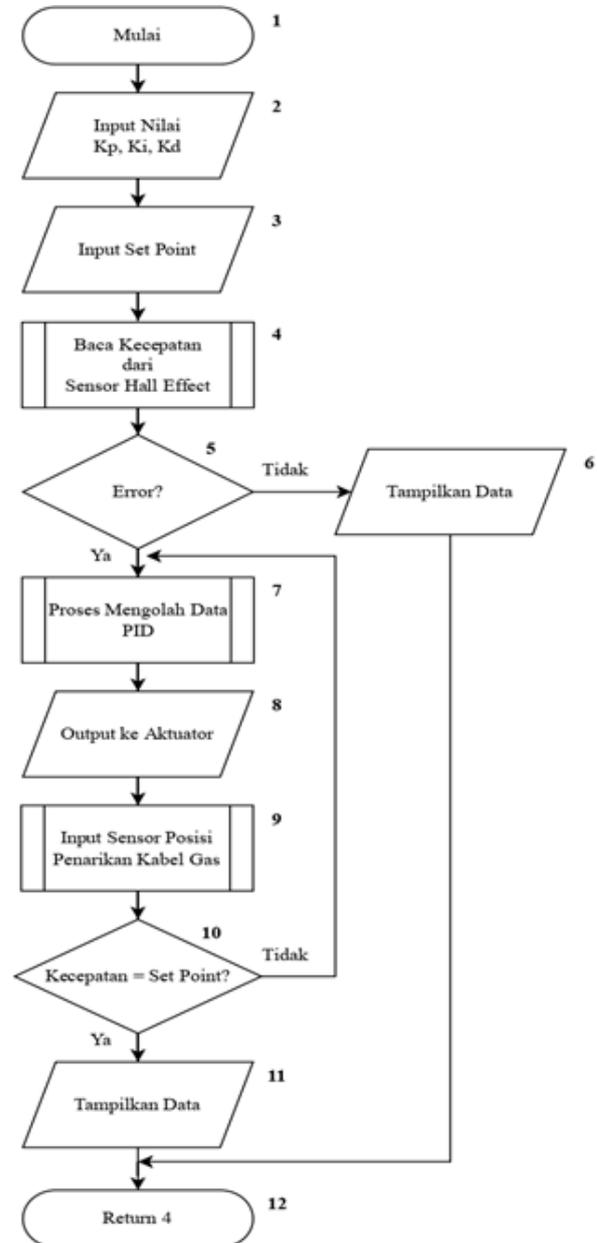
**Gambar 4.** Desain Alat

Dalam membuat alat yang memerlukan pemrograman, diperlukan membuat diagram alir atau *flowchart* agar dapat membantu analisis dan memecahkan masalah dalam pemrograman. Algoritma utama sistem menggambarkan alur kerja pada sistem secara keseluruhan dan juga menjelaskan urutan prosedur-prosedur yang dilakukan. Diagram alir ini diawali dengan mulai dan diakhiri dengan *return* pada langkah tertentu.

Masukan dari program ini adalah data yang dihasilkan dari sensor medan magnet yang digunakan sebagai sensor rpm, dan sensor potensiometer untuk mengetahui posisi penarikan gas. Nantinya data tersebut akan diproses dan diolah dalam mikrokontroler menggunakan metode pengendali PID. Hasil dari proses olah data tersebut akan dikirimkan ke aktuator yang digunakan sebagai pengatur kecepatan atau tarikan *throttle* gas mesin bensin. **Gambar 5** akan menjelaskan proses di atas melalui diagram alir utama sistem.

Algoritma pengolahan data sensor kecepatan menggambarkan alur kerja pada sistem arduino mega 2560 untuk proses pembacaan sensor digital *hall effect* dan juga menjelaskan urutan prosedur-prosedurnya. Diagram alir ini diawali dengan mulai dan diakhiri dengan proses *return* pada langkah ke-4. Data yang masuk dari sensor *hall effect* akan diproses atau dilakukan pengolahan data. Pengolahan data ini dilakukan dengan

menghitung durasi waktu sensor mendeteksi magnet yang satu dengan magnet selanjutnya (setiap satu putaran mendeteksi satu magnet).



**Gambar 5.** Diagram Alir Utama Sistem

Lalu proses perhitungannya dengan cara waktu 1 menit (60000 milisekon) dibagi dengan waktu sensor mendeteksi magnet atau waktu satu putaran mesin, untuk menghasilkan nilai rpm. Hasil rpm yang telah didapatkan lalu dibagi rata-rata agar nilai yang dihasilkan lebih terlihat stabil. Nilai yang telah dirata-ratakan lalu ditampilkan setiap 0,1 detik. **Gambar 6** akan menjelaskan proses di atas melalui diagram alir sub proses pengolahan data sensor kecepatan.



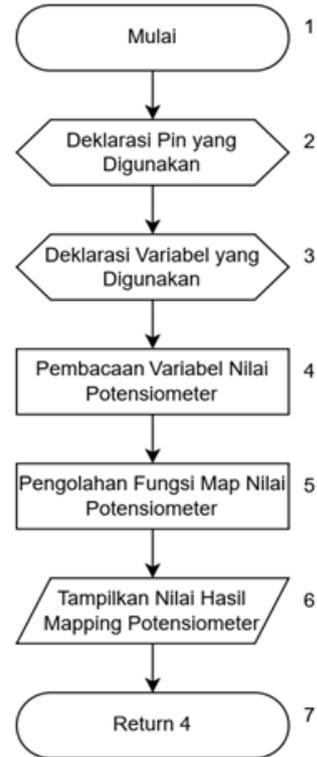
Gambar 6. Diagram Alir Sub Proses Pengolahan Data Sensor Kecepatan

Gambar 6 menampilkan diagram alir pengolahan data sensor kecepatan. Proses pengolahan data sensor kecepatan diawali dengan deklarasi pin antara pin sensor kecepatan dan pin mikrokontroler arduino mega 2560. Setelah itu, variabel yang digunakan dalam proses pengolahan data sensor kecepatan dideklarasikan. Untuk proses selanjutnya sama seperti pada penjelasan sebelumnya hingga ditampilkan nilai kecepatan putaran mesin.

Algoritma pengolahan data pada sensor potensiometer digambarkan melalui alur kerja pada sistem mikrokontroler arduino mega 2560. Proses pembacaan sensor analog potensiometer dan juga penjelasan urutan prosedur-prosedur yang dilakukan. Prosedur-prosedur tersebut dilakukan untuk memperoleh data analog sensor potensiometer yang akan diolah oleh mikrokontroler arduino mega 2560. Pengolahan data dilakukan dengan cara memakai fungsi *map* untuk memetakan pembacaan nilai analog yang digunakan (6-351) menjadi (0%-100%). Hasil pengolahan data potensiometer kemudian akan ditampilkan sesuai dengan nilai yang telah diolah berdasarkan fungsi *map*. Gambar 7 akan menjelaskan proses di atas melalui diagram alir sub proses pengolahan data potensiometer.

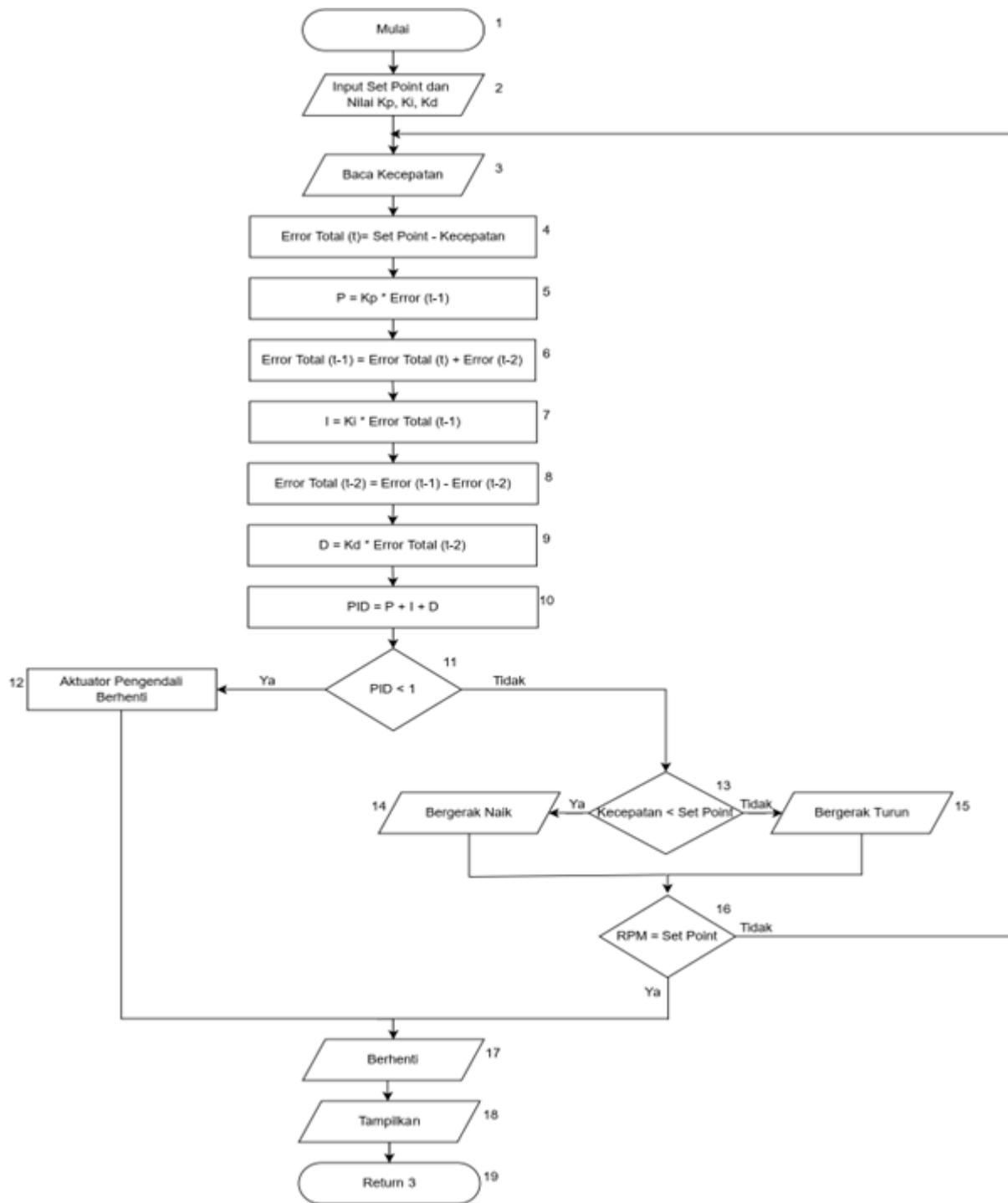
Pada Gambar 7 menunjukkan diagram alir sub proses pengolahan data pada sensor

potensiometer. Proses dimulai dengan adanya deklarasi pin yang dihubungkan antara pin pada sensor potensiometer dan mikrokontroler arduino mega 2560. Variabel yang digunakan pun dideklarasikan. Proses-proses berikutnya sama dengan penjelasan sebelumnya sampai nilai hasil *mapping* potensiometer ditampilkan.



Gambar 7. Diagram Alir Sub Proses Pengolahan Data Potensiometer

Algoritma pengolahan data menggunakan metode pengendali PID, menggambarkan alur kerja dan urutan prosedur-prosedur. Algoritma ini merupakan sistem utama pada penelitian ini. Proses pengolahan data pengendali PID dilakukan secara matematis sesuai dengan rumus pengendali PID. Hasil dari pengolahan data tersebut akan langsung diperintahkan untuk menggerakkan aktuator pengendali, pergerakan aktuator ada tiga kondisi yaitu apabila *set point* sudah sesuai dengan kecepatan maka aktuator pengendali akan berhenti, dan jika *set point* lebih besar dari kecepatan maka aktuator pengendali akan berputar menarik kabel *throttle gas*. Apabila jika kecepatan lebih besar dari *set point* maka aktuator pengendali akan bergerak sebaliknya atau menurunkan kabel *throttle gas*. Gambar 8 berikut ini akan menjelaskan proses yang telah dijelaskan menjadi diagram alir sub proses pengolahan data pengendali PID.



Gambar 8. Diagram Alir Sub Proses Pengolahan Data Pengendali PID

### III. HASIL DAN PEMBAHASAN

Untuk mengetahui keberhasilan dari implementasi alat untuk sistem kendali kecepatan mesin bensin menggunakan metode pengendali PID, maka diperlukan suatu pengujian. Proses pengujian yang akan dilakukan adalah pengujian perangkat keras untuk mengetahui proses pembacaan sensor kecepatan dan sensor posisi gas telah berfungsi dengan baik atau tidak. Kemudian

pada tahap akhir pengujian perangkat keras, akan ada pengujian keseluruhan perangkat keras untuk mengetahui pengaruh posisi gas terhadap pembacaan kecepatan.

Pertama, pengujian yang dilakukan adalah menguji sensor kecepatan putaran mesin bensin yang dikonversikan ke rpm. Pada pengujian ini mesin bensin akan dinyalakan dengan posisi gas paling rendah. Untuk menghasilkan pembacaan kecepatan yang optimal, pengujian dilakukan

dengan cara membandingkan nilai kecepatan yang tanpa *filter moving average* dengan yang menggunakan *filter moving average*. Pengujian pada sistem yang menggunakan *filter moving average* dilakukan dengan mengubah besaran sampling pada algoritma *filter moving average* ini.

Untuk pengujiannya dilakukan sebanyak 8 kali, diawali dengan pengujian tanpa filter dan selanjutnya menaikkan besaran sampling setiap 5 tahap sampai dengan 35 tahap, setiap pengambilan data dilakukan selama 30 detik. Untuk hasilnya dicari nilai selisih dengan cara nilai tertinggi dikurangi nilai terendah setiap percobaannya. Hasil pengujian sensor kecepatan dapat dilihat pada **Tabel I**.

**Tabel I.** Hasil Pengujian Sensor Kecepatan

Besaran Nilai Sampling Filter	Nilai Kecepatan Tertinggi (rpm)	Nilai Kecepatan Terendah (rpm)	Selisih Nilai (rpm)
1	6666	2376	4290
5	5172	2875	2297
10	4025	2821	1204
15	4038	2995	1043
20	3928	2666	1262
25	4126	3263	863
30	4559	3467	1092
35	4749	3616	1133

Hasil pengujian yang tertera pada **Tabel I** sensor kecepatan yang tanpa filter dan menggunakan filter, didapat nilai selisih yang paling kecil yaitu berada pada sensor kecepatan yang menggunakan filter dan memiliki besaran *sampling* 25. Pengujian selanjutnya dilakukan pada sensor potensiometer untuk mengetahui posisi gas. Pengujian dilakukan sebanyak 4 kali perintah nilai posisi gas, diawali dengan memerintahkan nilai posisi gas 25% dan dilakukan secara bertahap dengan menaikkan posisi gas 25% terus menerus, sampai dilakukan perintah terakhir sebanyak 100%. Batas maksimal pembacaan kecepatan sebesar 18000 rpm.

Untuk pengujiannya dilakukan dengan memerintahkan nilai posisi yang diinginkan pada program, dan nantinya *aktuator* pengendali kabel *throttle* gas akan menarik kabel *throttle* gasnya. Panjang penarikan kabel *throttle* gas oleh sistem akan dibandingkan dengan hasil pengukuran menggunakan penggaris, pembanding panjang penarikan kabel *throttle* ketika posisi 0% yaitu 0 cm dan ketika posisi 100% yaitu sepanjang 1,2 cm atau 12 mm. Dengan cara melihat perbandingan

panjang penarikan kabel *throttle*, hasil yang didapatkan yaitu berupa hasil selisih antara besar nilai posisi gas oleh sensor potensiometer dengan hasil pengukuran panjang penarikan kabel *throttle* gas yang terlihat pada penggaris. Hasil pengujian sensor potensiometer dapat dilihat pada **Tabel II**.

**Tabel II.** Hasil Pengujian Sensor Potensiometer

Nilai Potensio (%)	Hasil Panjang Penarikan		Selisih Panjang Penarikan (mm)
	Oleh Sistem (mm)	Seharusnya (mm)	
5	30	30	0
50	60	60	0
75	90	90	0
100	120	120	0

Hasil pengujian pada **Tabel II** untuk mengetahui posisi gas menggunakan sensor potensiometer dibandingkan dengan hasil penarikan yang seharusnya, didapat nilai selisih sebesar 0 mm atau tidak ada error yang didapatkan. Pengujian yang dilakukan selanjutnya yaitu pengujian dengan menggabungkan sensor kecepatan dengan sensor potensiometer sebagai sensor posisi gas. Pengujian ini dilakukan untuk mengetahui pengaruh posisi gas terhadap kecepatan yang terbaca oleh sensor kecepatan.

Pada pengujian ini akan dilakukan sebanyak 7 kali perintah posisi gas, diawali dengan posisi gas 5% kemudian dinaikkan bertahap setiap 5% hingga berakhir di 35%. Setiap posisi gas dinaikkan 5%, kemudian sensor kecepatan akan membaca kecepatan mesin bensin tersebut menggunakan nilai *sampling filter* 25. Hasil pengujian keseluruhan perangkat keras dapat dilihat pada **Tabel III**.

**Tabel III.** Hasil Pengujian Keseluruhan Perangkat Keras

Nilai Potensio (%)	Nilai Kecepatan			Selisih Nilai (rpm)
	Nilai Kecepatan Tertinggi (rpm)	Nilai Kecepatan Terendah (rpm)		
5	9214	8789		425
10	11739	11336		403
15	12599	12106		493
20	14149	13099		1050
25	16624	15410		1214
30	17878	17237		641
35	12152	3470		8682

Hasil pengujian keseluruhan perangkat keras dengan menggunakan sensor posisi gas dan sensor kecepatan menggunakan *filter*, didapat nilai pembacaan sensor kecepatan maksimal berada pada posisi gas 30. Apabila dinaikkan menjadi pada posisi gas 35, pembacaan sensor kecepatan sudah dianggap *error*.

Dalam menentukan hasil akhir pada sistem kendali pada alat ini perlu dilakukan suatu pengujian. Pengujian yang akan dilakukan kali ini yaitu mencari nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang tepat untuk mencapai suatu *set point*. Pada pengujian ini mesin bensin akan dinyalakan dengan posisi gas paling rendah atau 0%, lalu sistem akan langsung dijalankan. Untuk mencapai *set point* yang diinginkan dilakukan dengan cara mengubah nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang tepat dengan cara menggunakan metode *tuning trial and error* sampai sistem bekerja dengan baik dan dapat mencapai kecepatan yang sesuai dengan *set point*.

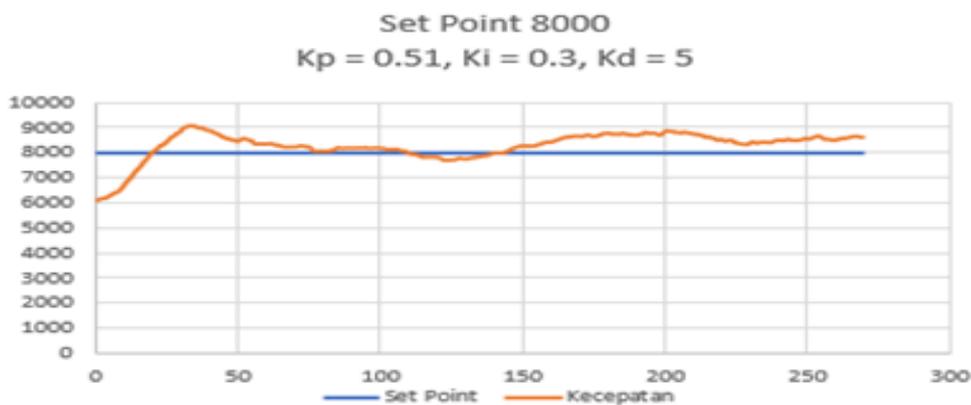
Untuk pengujian selanjutnya akan dilakukan dengan menggunakan metode *tuning trial and error* dengan dilakukan sebanyak 8 kali perubahan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang dapat dimasukkan pada

program sesuai dengan nilai diinginkan, dan untuk *set point* pada penelitian ini akan ditentukan sebanyak 8000 *rpm*. Untuk melihat hasil pengujianya akan dilihat nilai *rise time*, *maximum overshoot*, *settling time* dan *error steady state*. Grafik pengujian yang dinilai mempunyai respon terbaik akan disajikan pada pengujian ini. Tolak ukur keberhasilan sistem kendali PID pada mesin bensin ini, akan dilihat nilai *rise time* paling kecil dan *error steady state* paling kecil. Tolak ukur tersebut disesuaikan dengan tujuan penelitian yang telah dibuat sebelumnya. Hasil pengujian keseluruhan sistem dengan cara menentukan *set point* dapat dilihat pada **Tabel IV**.

Hasil yang didapat pada pengujian keseluruhan sistem dengan cara menentukan *set point* yang telah disajikan pada tabel di atas, didapatkan nilai parameter yang terbaik yaitu nilai  $K_p = 0.51$ ,  $K_i = 0.3$  dan  $K_d = 5$ . Pada nilai parameter tersebut didapat nilai *rise time* dan *error steady state* terkecil daripada nilai parameter lainnya. Untuk melihat hasil pengujian nilai parameter terbaik dalam bentuk grafik, dapat dilihat pada **Gambar 9**.

Tabel IV. Hasil Pengujian Set Point

Nilai Parameter			Set Point (rpm)	Rise Time (detik)	Max Over (rpm)	Settling Time (detik)	Error Steady State (rpm)
Kp	Ki	Kd					
0.4	0.1	3	8000	7.6	3000	16.7	2608
0.49	0.27	5.01	8000	4.9	3759	18.9	3118
0.49	0.4	5	8000	5.7	4612	26.9	4240
0.5	0.1	5	8000	13.2	1314	24.7	-1040
0.5	0.2	5	8000	-	-	15.6	-915
0.51	0.3	5	8000	2	1045	15.5	310
0.6	0.2	5	8000	4.4	2810	16.4	1197
0.6	0.3	6	8000	2.4	2641	20.3	3624



Gambar 9. Grafik Nilai Parameter Terbaik

Pengujian selanjutnya dilakukan dengan cara memberikan gangguan pada putaran mesin, untuk memastikan bahwa sistem sudah dapat bekerja dengan baik. Untuk pengujiannya dilakukan dengan mengatur nilai parameter  $K_p$ ,  $K_i$ ,  $K_d$  sesuai dengan nilai parameter terbaik ketika percobaan sebelumnya, dan untuk *set point* akan disamakan juga dengan yang sebelumnya. Ketika putaran mesin sudah mencapai kondisi *steady state*, maka akan dilakukan gangguan terhadap putaran mesinnya dengan cara memberikan tekanan pada putarannya, kemudian tekanan tersebut akan diukur menggunakan timbangan. Untuk grafik pengujian, salah satunya akan ditampilkan pada pengujian kali ini. Hasil pengujian keseluruhan sistem dengan cara memberikan gangguan dapat dilihat pada **Tabel V**.

Hasil yang didapat pada pengujian keseluruhan sistem dengan cara memberikan gangguan yang telah disajikan pada tabel di atas, didapatkan respon yang baik sesuai yang diharapkan. Untuk melihat hasil pengujian dengan memberikan gangguan tersebut dalam bentuk grafik, dapat dilihat pada **Gambar 10**.

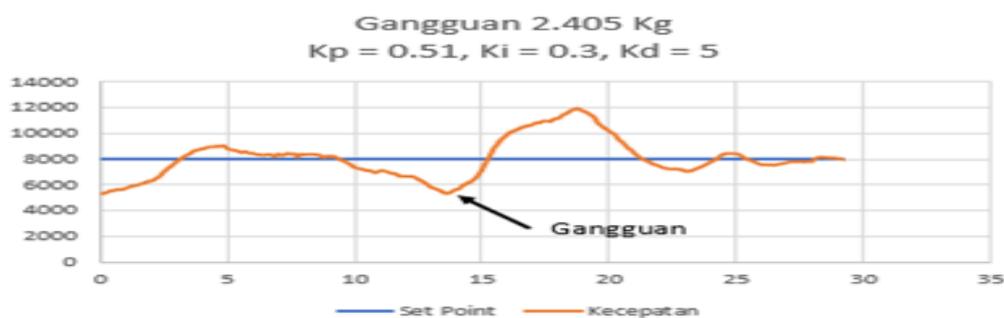
Pencarian nilai parameter  $K_p$ , parameter  $K_i$ , dan parameter  $K_d$  menggunakan metode *tuning trial and error*. Metode yang digunakan ini dapat menghasilkan nilai parameter terbaik yang sudah dilakukan pengujiannya yaitu berada pada nilai  $K_p = 0.51$ ,  $K_i = 0.3$  dan  $K_d = 5$ . Pada nilai-nilai parameter tersebut didapatkan waktu untuk mencapai kestabilannya adalah pada durasi waktu 15,5 detik. Kemudian, dan nilai *error*

kestabilannya diperoleh juga yang nilainya berada pada 310 rpm atau 3,88%. Untuk lebih meyakinkan kembali bahwa nilai-nilai parameter tersebut sudah dianggap terbaik, maka dilakukan pengujian. Pengujian dilakukan dengan memberikan gangguan berupa tekanan pada putaran mesin bensinnya. Hasil yang didapatkan dari pengujian dengan adanya gangguan menunjukkan bahwa memang nilai parameter tersebut sudah dapat dikatakan terbaik karena waktu untuk mencapai kestabilannya di bawah 20 detik dari puncak gangguan dan *error* kestabilannya di bawah 10%.

Pengendalian kecepatan putaran mesin pada mesin besin dengan menggunakan pengendali PID dapat dilakukan. Parameter hasil pengujian yang dapat diamati yaitu waktu untuk mencapai kestabilannya di bawah 20 detik dari puncak gangguan dan *error* kestabilannya di bawah 10%. Pada penelitian yang dilakukan oleh Yu dkk. [9] dengan metode *integral-separation fuzzy PID controller*, waktu untuk mencapai kestabilan lebih cepat yaitu di bawah 6 detik. Hal ini disebabkan oleh proses *tuning* parameter  $K_p$ , parameter  $K_i$ , dan parameter  $K_d$  menggunakan logika fuzzy. Kemudian, penelitian yang dilakukan oleh Lin dkk. [10] dengan metode *triple-step* memiliki nilai *maximum overshoot* yang tinggi dibandingkan dengan penelitian ini. Penyebab dari tingginya *maximum overshoot* adalah metode *triple-step* bersesuaian dengan metode pengendali PI sehingga parameter kendali D tidak diterapkan untuk menurunkan *maximum overshoot*.

Tabel V. Hasil Pengujian Dengan Gangguan

Nilai Parameter			Set Point (rpm)	Beban Gangguan (Kg)	Settling Time (second)	Error Steady State (rpm)
$K_p$	$K_i$	$K_d$				
0.51	0.3	5	8000	0.695	13.3	746
				1.235	19.3	517
				1.565	13.6	20
				2.405	14.6	120



Gambar 1. Grafik Pengujian Dengan Gangguan

#### IV. KESIMPULAN

Berdasarkan uraian yang telah dibahas pada bab sebelumnya dapat diambil beberapa kesimpulan, bahwa penerapan sensor kecepatan menggunakan *hall effect*, dapat diterapkan pada pengendalian kecepatan putaran mesin bensin menggunakan metode PID ini dengan batas maksimal pembacaan kecepatan 18000 rpm, atau bila dihitung menggunakan sensor potensiometer untuk mengetahui posisi gas yang digabungkan dengan motor DC sebagai aktuator pengendali gas yaitu batas pembacaan kecepatan maksimal berada pada posisi gas 30%. Besaran nilai parameter terbaik yang telah dilakukan selama pengujian dengan *set point* 8000 rpm yaitu bernilai  $K_p = 0.51$ ,  $K_i = 0.3$ ,  $K_d = 5$ . Nilai parameter tersebut dapat mencapai waktu kondisi kestabilan dengan optimal atau mencapai kondisi *settling time* 15,5 detik dan dengan *error* kestabilannya atau *error steady state* 3,88% atau 310 rpm.

#### DAFTAR PUSTAKA

- [1] M. Riski Nur, A. Ifnu Arba'i, and Muhyin, "Analisa Pengaruh Variasi Sudut Kemiringan Bucket Conveyor Dan Kecepatan Putaran Poros Terhadap Optimasi Kapasitas Yang Dihasilkan Oleh Mesin Pengemas Gabah (Grain Packaging Machine)," *Publikasi Online Mahasiswa Teknik Mesin*, vol. 3, 2020.
- [2] C. Marin, D. Popescu, E. Petre, dan D. Selisteanu, "Mathematical modelling of systems with state dependent delays. Applications for conveyor belt plants," dalam *2016 17th International Carpathian Control Conference (ICCC)*, High Tatras, Slovakia: IEEE, Mei 2016, hlm. 479–484. doi: 10.1109/CarpathianCC.2016.7501145.
- [3] Pujono, A. Setiawan, and D. Prabowo, "Rancang Bangun Mekanisme Pergerakan Conveyor Pada Mesin Sortir Sampah Kaleng Dan Botol Plastik," 2020.
- [4] T. Allam, M. Raju, and S. S. Kumar, "Design of PID controller for DC Motor Speed Control Using Arduino Microcontroller," *International Research Journal of Engineering and Technology*, 2016, [Online]. Available: www.irjet.net
- [5] M. Rizki Setiawan, M. Aziz Muslim, and G. Dwi Nusantoro, "Kontrol Kecepatan Motor DC Dengan Metode PID Menggunakan Visual Basic 6.0 Dan Mikrokontroler ATmega 16," 2012.
- [6] A. Triwiyatno, E. W. Sinuraya, J. D. Setiawan, dan S. Munahar, "Smart controller design of air to fuel ratio (AFR) and brake control system on gasoline engine," dalam *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, Indonesia: IEEE, Okt 2015, hlm. 233–238. doi: 10.1109/ICITACEE.2015.7437805.
- [7] D. Araventa Silalahi, "Pengendalian Kecepatan Putar (Rpm) Motor Dc Dengan Metode Pid Berbasis Mikrokontroler Atmega328 Menggunakan Bahasa Pemrograman Code Vision Avr," 2017.
- [8] M. Fajri Nur Reimansyah, A. Triwiyatno, and B. Setiyono, "Perancangan Kontrol Fuzzy Adaptif Pada Sistem Kontrol Kecepatan Stasioner Mesin Bensin," 2014.
- [9] H. Yu, Y. Liu, Y. Wang, M. Wang, dan X. Qin, "Speed Governing Controller of Gasoline Engine Based on Integral-Separation Fuzzy PID Control," dalam *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, Beijing, China: IEEE, Jul 2016, hlm. 1131–1135. doi: 10.1109/ICISCE.2016.243.
- [10] J. Lin, H. Chen, dan P. Wang, "AFR control of a gasoline engine using triple-step method," dalam *2017 29th Chinese Control And Decision Conference (CCDC)*, Chongqing, China: IEEE, Mei 2017, hlm. 1976–1981. doi: 10.1109/CCDC.2017.7978840.
- [11] R. Anjum, A. Yar, A. I. Bhatti, I. K. Yousufzai, dan Q. Ahmed, "Dual Loop Speed Tracking Control for Torque Management of Gasoline Engines," dalam *2019 18th European Control Conference (ECC)*, Naples, Italy: IEEE, Jun 2019, hlm. 3084–3089. doi: 10.23919/ECC.2019.8795863.
- [12] N. Yuliamas, S. Aisyah, and H. Toar, "Implementasi Kontrol PID pada Mesin Pengembang Roti," *Jurnal Rekayasa ElektriKa*, vol. 11, no. 3, pp. 109–113, Apr. 2015, doi: 10.17529/jre.v11i3.2304.
- [13] S.Rahmani, S. A. Rosana, dan G. H. Tian. "Sistem Kontrol Level Air Dengan Pengontrol PID." *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan* 10.2, 174-181, 2022.
- [14] R. Hartono and A. Kuart Jaenudin, "Implementasi Sistem Navigasi Wall Following Masukan Sensor Ultrasonik Menggunakan Metode Tuning Kendali PID," 2017.
- [15] I. Husnaini and Krismadinata, "Komparasi Pengendali Pi Dan Pid Untuk Tegangan Keluaran Konverter Buck," *JNTE*, vol. 6, no. 3, 2017, doi: 10.20449/jnte.v6i3.387.
- [16] D. Hendrawati, *Respon Sistem Ditinjau Dari Parameter Kontroler Pid Pada Kontrol Posisi Motor DC*. 2012.
- [17] Y. Aditya, B. Irawan, and R. Erfa Saputra, "Perancangan Dan Implementasi Sistem Kendali Penerbangan Quadcopter Berbasis Pid Menggunakan Gps Dan Komunikasi Lora Pada Pencarian Pendaki Gunung Yang Hilang," *e-Proceeding of Engineering*, vol. 4, no. 3, pp. 4186–4193, 2017.
- [18] Waluyo, A. Fitriansyah, and Syahril, "Analisis Penalaan Kontrol PID pada Simulasi Kendali Kecepatan Putaran Motor DC Berbeban menggunakan Metode Heuristik," 2013.
- [19] E. Setyawan Gembong, E. Setiawan, and W. Kurniawan, "Sistem Kendali Ketinggian Quadcopter Menggunakan PID," 2015.
- [20] T. N. Nizar, D. A. Jatmiko, R. Hartono, and A. I. G. Pratama, "Implementasi dan Uji Kinerja Kontrol PID untuk kestabilan Pesawat Tanpa Awak Tailsitter pada Keadaan Mengambang," *Komputika : Jurnal Sistem Komputer*, vol. 9, no. 2, pp. 53–59, Mar. 2020, doi: 10.34010/komputika.v10i1.3808.
- [21] S. Fatimah Anggraini, A. Ma'arif, and R. Dwi Puriyanto, "Pengendali PID pada Motor DC dan Tuning Menggunakan Metode Differential Evolution," *TELKA*, vol. 6, no. 2, pp. 147–159, 2020.
- [22] H. Marwa Amaroh, B. Anggraheny, and D. Dewatama, "Kontrol Kecepatan Putar Motor pada Proses Pembuatan Arum Manis dengan Metode PID," *ELKOLIND*, vol. 05, no. 1, pp. 17–22, 2018.