

**KONTROL POSISI DAN KECEPATAN LIFT MENGGUNAKAN KENDALI PID**  
**LIFT POSITION AND SPEED CONTROL WITH PID CONTROLLER**

Muhammad Arsyad Budiman<sup>1</sup>, Agung Surya Wibowo, S.T.,M.T.<sup>2</sup>, Dr.Erwin Susanto,S.T, M.T.<sup>3</sup>

<sup>1,2,3</sup> Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[rsatbudiman@gmail.com](mailto:rsatbudiman@gmail.com) <sup>2</sup>[agungsw@telkomuniversity.ac.id](mailto:agungsw@telkomuniversity.ac.id),

<sup>3</sup>[erwinelektro@telkomuniversity.ac.id](mailto:erwinelektro@telkomuniversity.ac.id)

**Abstrak**

Latar belakang dari penelitian ini adalah untuk membuat sebuah sistem kendali yang dapat mengatur dan meningkatkan kepresisian dari kecepatan putar motor yang terhubung dengan katrol yang terdapat pada sebuah sistem lift dan posisi lift di sistem tersebut. Untuk mencapai tujuan tersebut, penulis menggunakan kendali kecepatan dan posisi yang menggunakan metode PID.

Dalam penelitian ini penulis menggunakan kendali posisi untuk mengendalikan posisi dan kendali kecepatan untuk mengendalikan kecepatan putar motor yang terdapat pada system lift. Kendali kecepatan dan posisi menggunakan metode PID untuk meminimalisir error yang timbul dari system lift tersebut. Setpoint yang dipakai adalah ketinggian dan kecepatan putar motor dari sistem lift tersebut. Input yang dipakai adalah ketinggian dari lift yang didapatkan menggunakan sensor ultrasonic dan kecepatan putar motor yang didapatkan dengan menggunakan sensor optocoupler. Data yang diperoleh lalu diproses dengan menggunakan Arduino UNO dengan metode PID untuk menghasilkan output berupa nilai PWM . Nilai PWM tersebut lalu digunakan untuk memutar motor DC, sehingga lift akan mencapai kecepatan dan ketinggian yang diinginkan.

Pada akhir penelitian ditemukan alat sudah dapat mengontrol kecepatan dan posisi dari lift hanya saja masih terdapat error didalam sistem control posisi yang dibuat.

*Abstract*

*The background of this research is to make a control sistem that can control and increase the precision of the speed of a motor and the position of a lift. In order to do that, the author use a speed and position control that is using a PID controller.*

*In this research, The author use a position control to control the position of a lift and a speed controller to control the rotation speed of a motor that is connected to the system. PID controller is utilized for both speed and position control to minimize the error that occur in the lift system. The setpoints that are used in this controller is the speed of the motor and the position of the lift. The inputs that are used in this system is the height of the lift that is acquired using a ultrasonic sensor and the rotation speed of the motor that is acquired using optocoupler sensor. The data that have been acquired then be processed using Arduino UNO using PID method to get the output that is in PWM. Then that output then be used to spin the DC motor so the lift will go to the position that we have set using setpoint.*

*In the end, the control system could control the position and speed of the lift but there is an error that still occur.*

**1. Pendahuluan**

Latar belakang dari penelitian ini adalah untuk membuat sebuah sistem kontrol yang dapat mengatur ketinggian dan kecepatan untuk sebuah lift. Sistem kontrol ini dibuat sehingga sebuah lift dapat diatur ketinggian dan kecepatannya dengan tingkat presisi yang relatif tinggi. Operator dapat mengatur kecepatan motor DC untuk mengatur seberapa cepat lift tiba pada tujuan. Operator yang dimaksud disini adalah orang yang mengatur kecepatan dan ketinggian lift dari luar lift. Operator juga dapat mengatur ketinggian lift. Operator dapat mengatur setpoint dengan menggunakan push button yang terdapat pada alat dan melihat ketinggian dan kecepatan dari lift secara langsung menggunakan LCD 16x2.

Dalam penelitian ini penulis mencapai tujuan tersebut dengan cara membuat sebuah kendali posisi dan kecepatan yang lalu diimplementasikan ke sebuah sistem lift menggunakan metode PID. Sistem kendali posisi dan kecepatan itu akan bekerja untuk mengendali kecepatan putar motor yang terhubung dengan sistem lift dan mengendalikan ketinggian lift tersebut sementara kontroler PID bekerja untuk meminimalisir error yang terdapat pada sistem tersebut. Setpoint yang dipakai untuk sistem kendali adalah posisi vertikal lift dari tanah dan kecepatan putar per menit dari motor DC. Sensor akan digunakan untuk merekam kecepatan motor DC adalah sensor optocoupler LM393 dan sensor yang digunakan untuk mendapatkan ketinggian vertikal dari lift adalah sensor ultrasonic HC-SR04. Data yang telah direkam oleh kedua sensor ini lalu dibandingkan dengan setpoint yang telah ditentukan untuk mengetahui error yang terdapat pada sistem. Error lalu disimpan untuk dimasukkan ke rumus kontrol PID yang lalu akan mengeluarkan nilai yang dipakai untuk menjadi PWM ( Pulse Width Modulation ) yang diberikan untuk motor DC sehingga motor DC dapat menggerakkan lift ke kecepatan dan ketinggian yang diinginkan. Lift dapat diatur dengan menggunakan push button dan dapat dilihat keluarannya secara realtime dengan menggunakan LCD 16x2.

## 2. Tinjauan Pustaka

### 2.1 Prinsip Kerja Ide

Alat yang akan dibuat memiliki prinsip kerja sebagai berikut :

1. *Setpoint* yang digunakan untuk alat ini berupa ketinggian dari lift dan kecepatan putar motor DC. dimasukan juga nilai koefisien  $K_p$ ,  $K_i$ , dan  $K_d$  yang digunakan untuk kendali PID.
2. *Setpoint* yang telah dimasukan lalu dibandingkan dengan masukan yang datang dari *sensor* yang berupa ketinggian dari lift dan kecepatan putar motor DC dengan cara membuat selisih antara *setpoint* dan masukan yang datang dari *sensor*. Selisih ini lalu dijadikan error untuk dipakai dalam perhitungan kendali PID.
3. Error yang didapat lalu dikalikan dengan  $K_p$ ,  $K_i$  dan  $K_d$  sesuai dengan rumus PID yang akan dibahas pada bagian selanjutnya.
4. Hasil perkalian *error* dengan  $K_p$ ,  $K_i$  dan  $K_d$  akan ditambahkan yang akan menghasilkan nilai PID yang dibutuhkan. Nilai PID tersebut lalu digunakan untuk menghasilkan nilai PWM.
5. Nilai PWM yang dihasilkan lalu dipakai untuk memutar motor DC yang nantinya akan berputar sesuai dengan *setpoint* kecepatan yang ditentukan dan akan menggerakkan lift menuju ketinggian yang telah ditentukan.

### 2.2 Kontrol Posisi

Kendali kontrol posisi adalah kendali yang dibuat untuk mengatur posisi sebuah benda sehingga posisi benda tersebut sesuai dengan *setpoint* yang diinginkan.

Pada tugas akhir ini kendali posisi ditambahkan dengan kendali PID untuk meminimalisir error yang terjadi. *Setpoint* yang diinginkan adalah ketinggian lift dari tanah. Input yang akan diproses pada mikrokontroler Arduino UNO adalah ketinggian dari lift yang dihasilkan oleh sensor ultrasonic HC-SR04. Data yang telah didapat dari sensor ultrasonic HC-SR04 akan diproses menggunakan mikrokontroler Arduino UNO dengan menggunakan metoda PID sehingga menghasilkan output yang berupa PWM yang diinginkan. Nilai PWM yang didapat lalu dipakai untuk menggerakkan motor DC yang nantinya akan menggerakkan lift menuju ketinggian yang diinginkan.

### 2.3 Kontrol Kecepatan

Kendali kecepatan adalah kendali yang dibuat untuk mengatur kecepatan suatu benda. Kendali kecepatan ini biasanya digunakan jika seseorang menginginkan kecepatan spesifik untuk sebuah benda.

Pada tugas akhir ini, kendali kecepatan digunakan untuk mengendalikan kecepatan putar per menit motor DC. cara kerja kendali kecepatan ini adalah pertama operator mengatur *setpoint* rotasi per menit yang diinginkan. lalu *setpoint* tersebut digunakan untuk mencari tahu error yang ada pada sistem dengan cara mencari selisih antara data rotasi per menit motor pada saat itu juga yang didapat dari sensor LM393. Error tersebut lalu dipakai untuk mencari tahu nilai PWM yang dibutuhkan untuk menghilangkan error dengan menggunakan metode PID. Setelah nilai PWM didapat, PWM digunakan untuk memutar motor DC sehingga motor DC memiliki putaran per menit yang sesuai dengan *setpoint* yang telah ditentukan sebelumnya.

### 2.4 Kendali PID

Kendali PID adalah sistem yang digunakan untuk meningkatkan kepresisian, waktu respon dan menghilangkan gangguan seperti osilasi dari sistem secara keseluruhan. Kendali PID ini terdiri dari tiga komponen, yaitu P (proportional), I (Integral) dan D (Derivative). Setiap komponen tersebut memiliki kelebihan dan kekurangan masing masing.

Komponen yang pertama adalah kendali proportional. Kendali proportional berguna untuk mempercepat respon dari sistem. Kendali proportional memiliki efek samping yaitu menyebabkan overshoot dan menyebabkan osilasi.

Komponen yang kedua adalah adalah kendali Integral. Kendali Integral ini berguna untuk menghilangkan error steady state pada respon system. Efek samping dari kendali integral adalah memperparah osilasi dan overshoot dari respon system.

Komponen yang terakhir adalah kendali Derivative. Kendali derivative ini berfungsi untuk menghilangkan overshoot dan osilasi dari respon sistem sehingga meningkatkan settling time dari respon sistem.

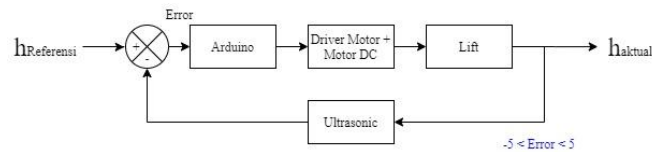
Berikut ini adalah rumus matematis dari kendali PID yang akan digunakan untuk menentukan nilai PID yang dibutuhkan untuk menghilangkan error yang terjadi dari kecepatan dan posisi dari alat.

$$PID = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.1)$$

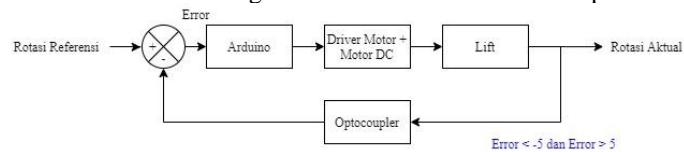
Pada dasarnya kendali PID menggunakan error yang terjadi dari alat ( error adalah selisih yang terjadi antara *setpoint* dengan data yang telah direkam oleh sensor ) dan mengalikannya dengan koefisien dari masing-masing komponen dari kendali PID (  $K_p$ ,  $K_i$ ,  $K_d$  ). Perbedaannya adalah, pada kendali proportional error tersebut hanya dikalikan dengan koefisien  $K_p$ . Pada kendali Integral, error tersebut diintegrasikan lalu dikalikan dengan koefisien  $K_i$ . Pada kendali derivative, error diturunkan lalu dikalikan dengan koefisien  $K_d$ . Lalu semua perhitungan tersebut akan ditambahkan dan akan menghasilkan nilai PID yang dibutuhkan untuk diberikan kepada aktuator yang akan menghilangkan error pada alat dengan cara mengkoreksinya dengan menggunakan data tersebut.

## 3. Perancangan Sistem

### 3.1.1 Diagram Blok Sistem

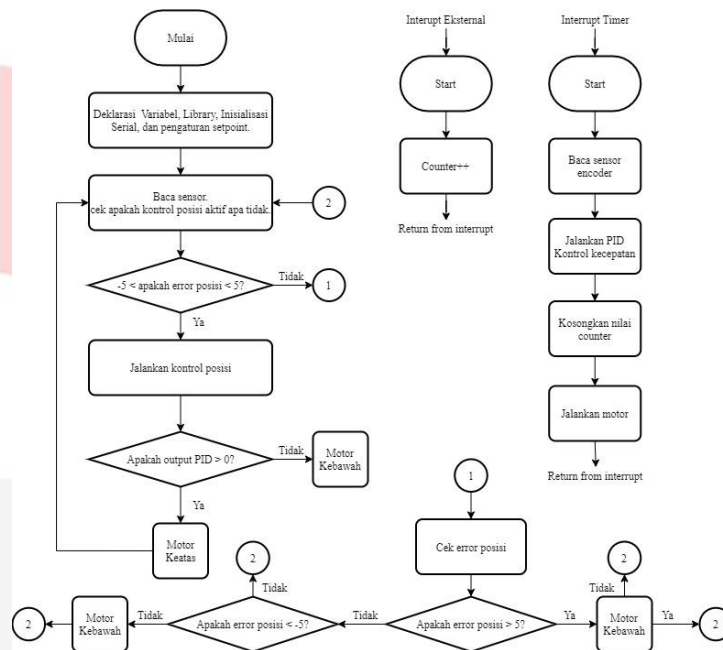


Gambar III-1. Diagram blok sistem mode kendali posisi



Gambar III-2. Diagram blok sistem mode kendali kecepatan

### 3.1.2 Flow Chart



Gambar III-3. Flowchart dari sistem

### 3.1.3 Fungsi dan Fitur

Fungsi dan fitur berdasarkan diagram blok di atas adalah sebagai berikut.

- Motor DC : berguna sebagai penggerak dari system, motor DC ini dipakai untuk menggerakkan beban yang dipasang pada tali yang melewati katrol keatas dan kebawah. Motor DC yang dipakai memiliki tegangan kerja sebesar 12V dan arus 1.5A.
- Sensor : sensor yang dipakai adalah LM393 untuk mencari tahu putaran per menit dari motor yang terdapat pada sistem. RPM yang didapat dari sensor dipakai untuk dibandingkan dengan *setpoint* yang nantinya akan menghasilkan error yang akan dipakai pada perhitungan PID. Untuk ketinggian, sistem menggunakan sensor ultrasonic HC-SR04 untuk membaca ketinggian dari lift.
- Mikrokontroler Arduino UNO : berfungsi sebagai otak dari system, mikrokontroler akan digunakan untuk melakukan kalkulasi PID dan mencari tahu nilai PWM yang dibutuhkan yang lalu nilai PWM tersebut diberikan ke motor *driver* untuk dijadikan nilai seberapa cepat motor berputar.
- Motor *driver* : berguna untuk mengontrol motor DC dan sebagai tempat pemasangan sumber dari luar, karena tegangan yang didapat motor DC dari mikrokontroler tidak kuat untuk memutar motor DC dengan sepenuhnya. Selain itu dapat digunakan untuk mengatur arah putar dan pwm dari motor DC dengan menggunakan sinyal yang didapat dari Arduino UNO.
- Aki 12V : Digunakan untuk menjadi sumber tegangan yang dapat menggerakkan motor. Aki ini dibutuhkan dikarenakan tegangan Arduino tidak cukup untuk menggerakkan motor dengan kecepatan penuh.
- Tiang penyangga : digunakan untuk menyangga beban.
- Ember : digunakan sebagai tempat untuk beban yang diangkat oleh lift.
- LCD 16x2 : Digunakan untuk memberikan operator info tentang kecepatan dan ketinggian lift secara *realtime*. Dapat juga digunakan untuk melihat *setpoint* kecepatan dan ketinggian yang telah diatur dengan menggunakan *push button*.

- *Push-button* : Digunakan untuk mengatur ketinggian dan kecepatan dari lift.

### 3.2 Desain Perangkat Keras

Perangkat keras yang dibuat adalah lift. Sistem lift yang dibuat menggunakan dari tiang dengan dimensi ketinggian 1m, lebar 0.5m dan panjang 1m yang terbuat dari besi bolong yang digunakan untuk menyangga katrol dan dan ember yang digunakan sebagai tempat beban untuk lift.

Lift ini menggunakan motor DC sebagai penggerakannya. Batang motor DC diikat dengan menggunakan tali yang lalu melalui sebuah katrol yang lalu tali tersebut tersambung dengan ember berukuran 5L untuk tempat bebannya. Motor DC dipasang dengan piringan optic yang melalui sensor optocoupler LM393 yang digunakan untuk menghitung kecepatan putaran permenit dari motor DC. Dibawah ember terdapat sensor ultrasonic yang digunakan untuk mencari ketinggian dari lift.

Sensor tersebut disambungkan dengan mikrokontroler Arduino UNO yang menggunakan data dari kedua sensor untuk menghitung PWM dengan menggunakan metoda PID yang dibutuhkan untuk memutar motor DC yang lalu berputar dengan sepoint putaran permenit yang ditentukan dan menggerakkan lift untuk mencapai tujuan yang ditentukan. Motor DC 12V lalu disambungkan dengan motor driver yang memberikan motor DC 12V dengan tegangan sebesar 12V sehingga motor dapat berputar dengan kecepatan penuh. Motor driver tersebut disambungkan dengan mikokontroler Arduino UNO yang akan memberikan motor driver dengan nilai PWM yang dikirimkan ke pin ENB untuk menentukan kecepatan putar motor DC dan nilai HIGH dan LOW untuk pin IN3 dan IN4 untuk megatur arah putar motor DC. Ketinggian dan kecepatan lift dapat diatur dengan menggunakan push button dan menggunakan Arduino IDE pada saat program pertama kali di upload menggunakan Arduino IDE.

### 3.3 Desain Perangkat Lunak

Di bagian lampiran terdapat program yang digunakan untuk menjalankan alat.

Perangkat lunak dibuat dengan menggunakan Arduino IDE. Perangkat lunak ini berguna sebagai proses yang digunakan oleh alat untuk memproses masukan yang didapat dari sensor dan mengeluarkan keluaran berupa PWM yang digunakan motor driver untuk memutar motor DC. Motor Driver menggunakan sinyal HIGH dan LOW yang dikirimkan oleh arduino menuju pin IN3 dan IN4 untuk mengatur arah putar dari motor DC. Pin IN3 dan IN4 harus menerima sinyal yang berbeda misalnya IN3 menerima LOW sementara IN4 menerima HIGH sehingga motor dapat berputar. Untuk memutar motor dengan arah sebaliknya, cukup dengan menukar sinyal yang diterima oleh kedua pin tersebut ( contoh : IN3 = HIGH, dan IN4 = LOW ).

Didalam perangkat lunak ini kita dapat menentukan setpoint ketinggian dari lift dan kecepatan putar dari motor DC. Kita juga dapat mengaturnya ketika alat sudah berjalan dengan menggunakan tombol yang terdapat pada alat. Setpoint yang telah dimasukkan kedalam alat nantinya akan dibandingkan dengan posisi dan kecepatan dari lift untuk menghasilkan error pada alat. Error ini lalu dimasukkan kedalam persamaan Proportional (3.1) dan Integral (3.2) yang lalu kedua persamaan ini ditambahkan yang nantinya akan menghasilkan nilai PWM yang akan digunakan oleh motor pada persamaan (3.3). Dikarenakan alat memiliki dua mode yaitu mode kendali kecepatan dan kendali posisi. Alat memiliki dua buah konstanta PID yang berbeda yang dipakai untuk masing masing mode kendali.

$$P=Error \times Kp \quad (3.1)$$

$$I=Ki \times Error + I \quad (3.2)$$

$$PI=P+I \quad (3.3)$$

PWM ini dipakai untuk memutar motor DC sehingga lift berputar dengan kecepatan dan bergerak menuju ketinggian yang ditentukan oleh operator. Pada saat lift bergerak menuju tujuan sensor akan merekam kecepatan dan posisi lift untuk dibandingkan dengan setpoint untuk mencari error. Proses ini akan berulang terus menerus sampai lift tiba pada tujuan.

Perangkat lunak memiliki dua mode operasi, yaitu kontrol kecepatan dan posisi. Pada saat motor pertama kali berputar sampai dengan posisi yang kita tentukan dari setpoint posisi, motor akan berputar dengan menggunakan mode kendali kecepatan. Lalu pada saat lift mendekati setpoint ketinggian yang diinginkan kendali berganti mode menjadi kendali posisi. Lalu perangkat lunak akan mencoba untuk memberikan PWM yang cukup untuk membuat lift menuju setpoint ketinggian yang diinginkan.

## 4. Hasil Pengujian dan Analisa

### 4.1 Pengujian Sistem

Percobaan dilakukan dengan mengambil sample sebanyak 30x dan membandingkan rata rata yang didapat dengan setpoint yang ditentukan untuk mendapatkan error dan akurasi dari kendali kecepatan dan posisi.

Perekaman posisi dilakukan dengan menggunakan sensor ultrasonic HC-SR04. Sample yang keluar dari serial monitor diambil dan dijadikan data untuk percobaan posisi. Data dari serial monitor memiliki waktu sampling 0.5 detik. Cara kerja untuk sensor ultrasonic ini adalah dengan cara menembakan gelombang suara ke bawah ember yang digunakan untuk mengangkut beban yang lalu merekam hasil pantulannya. Jeda antara penembakan dan perekaman gelombang suara yang ditembakkan lalu dipakai untuk mencari jarak lift dari lantai. Persamaan yang

digunakan pada alat dapat dilihat pada persamaan 4.1. Satuan yang digunakan untuk jarak dalam percobaan ini adalah cm.

$$\text{Jarak} = (\text{Waktu} \times 340) / 2 \quad (4.1)$$

Angka 340 pada persamaan 4.1 menunjukkan kecepatan suara di udara. Perekaman kecepatan dilakukan dengan mencari tahu putaran per menit dari motor DC dengan menggunakan sensor LM393 dengan cara menggunakan pulse yang diberikan oleh sensor. Secara detail, alat menggunakan piringan hitam yang dinamakan optical encoder yang memiliki sekat-sekat yang digunakan untuk menghalangi sinar laser yang ditembakkan oleh sensor LM393. Setiap kali sinyal laser terhalangi maka sensor akan mengirimkan sinyal yang lalu akan direkam oleh Arduino UNO sebagai 1 step. Setelah 24 step terkumpul maka disitulah kita akan tahu bahwa motor DC telah melakukan 1 kali putaran. Alat mencari tahu jumlah putaran per menit dengan menggunakan persamaan 4.1. Satuan yang digunakan untuk putaran motor dalam percobaan ini adalah RPM atau rotasi per menit.

$$\text{Rotasi Per Menit} = ((\text{Jumlah Step}) / 24) \times 2 \times 60 \quad (4.2)$$

Rotasi permenit dilakukan dengan mencari tahu berapa jumlah step yang telah direkam oleh Arduino UNO yang lalu dibagi dengan 24. Angka 24 ini berasal dari jumlah lubang yang terdapat pada piringan optocoupler. Lalu hasil dikalikan dengan 2 dikarenakan waktu sampling adalah 0.5 detik. Lalu setelah itu dikalikan lagi dengan 60 untuk merubah rotasi per detik menjadi rotasi per menit.

#### 4.2 Pengujian Kecepatan

Pengujian kecepatan dilakukan untuk menguji kecuratan alat dan mencari tahu error yang terjadi pada kontrol kecepatan alat. Pengujian alat dilakukan dengan mengambil sample sebanyak 30x ketika alat bergerak menuju tujuan dengan posisi 1cm per sample. Tabel IV-1 adalah data yang didapat dari pengujian alat. Alat diuji dengan menggunakan beban 1kg,  $k_p = 0.2$ ,  $k_i = 0.1$ . setpoint yang digunakan adalah 100 rotasi per menit

Tabel IV-1. Pengujian kecepatan

	Jarak (cm)	Rotasi Per Menit	Output (PWM)
	10	85	27
	11	91	31
	12	104	28
	13	95	32
	14	93	33
	15	96	32
	16	99	32
	17	101	33
	18	104	31
	19	100	32
	20	96	35
	21	88	38
	22	102	35
	23	100	35
	24	90	37
	25	96	36
	26	102	35
	27	99	36
	28	96	37
	29	106	34
	30	101	34
	31	93	36
	32	96	36
	33	100	35
	34	96	37
	35	91	39
	36	95	38
	37	104	36
	38	89	41
	39	81	45
Rata-Rata	24.5	96.3	34.86667

Dari table tersebut kita dapat mendapatkan nilai error dengan menggunakan persamaan 4.3.

$$\text{Error} = ((\text{Setpoint} - \text{Rata rata kecepatan}) / (\text{setpoint})) \times 100\% \quad (4.3)$$

Dengan menggunakan persamaan 4.3 didapatkan error sebesar.

$$\text{Error} = \frac{(100 - 96.3)}{100} \times 100\% = 3,7\% \quad (4.4)$$

Dengan didapatkannya error kita juga dapat mengetahui bahwa akurasi dari kontrol kecepatan pada alat tugas akhir ini adalah 96,3%.

Error yang terdapat pada alat kemungkinan besar disebabkan oleh keterbatasan yang disebabkan oleh sensor. Ketika alat mengeluarkan output yang yang dapat dilihat pada saat alat berada pada ketinggian 15cm. alat mengeluarkan output yang sama dengan ketinggian 16cm tetapi sensor membaca rotasi per menit yang berbeda.

#### 4.3 Pengujian Posisi

Pengujian posisi dilakukan dengan mengambil sample sebanyak 30 kali ketika alat berganti dari mode kecepatan ke mode posisi untuk mengetahui error dan akurasi kontrol posisi pada sistem. Percobaan ini dilakukan dengan menggunakan beban yang sama yaitu 1kg dan menggunakan konstanta  $K_p = 5$  dan  $K_i = 0.2$ .

Tabel IV-2. Pengujian posisi.

	Jarak(cm)	Output(PWM)
	45.02	25
	45.43	24
	45.98	22
	46.03	38
	46.15	39
	46.39	39
	46.6	40
	46.65	42
	47.16	40
	47.32	40
	47.94	39
	48.35	38
	48.42	38
	48.47	39
	48.69	38
	48.75	39
	48.9	38
	48.99	38
	49.14	37
	49.38	37
	50.03	34
	50.22	33
	49.28	38
	49.69	36
	49.73	36
	49.73	36
	49.74	36
	49.36	38
	49.42	38
	49.35	38
Rata-Rata	48.21033	36.43333

Diatas adalah hasil pengujian dari kontrol posisi. Dengan menggunakan data yang telah kita dapat. Kita dapat menggunakan rumus yang sama seperti sebelumnya untuk menentukan error yang terdapat pada kontrol posisi pada sistem. Persamaan yang digunakan untuk mencari error posisi adalah persamaan 4.5.

$$\text{Error} = \frac{(\text{Setpoint} - \text{Rata Rata Jarak})}{\text{setpoint}} \times 100\% \quad (4.5)$$

Dengan memasukan nilai yang kita dapat kedalam rumus tersebut kita dapat mendapatkan error.

$$\text{Error} = \frac{(50 - 48.21033)}{50} \times 100\% = 3,579334\% \quad (4.6)$$

Dengan menggunakan Persamaan 4.6 kita bisa menentukan bahwa error yang terdapat pada kendali posisi pada alat adalah sebesar 3,579334% dan akurasinya adalah 96,420666%.

#### 4.4 Pengaruh beban pada akurasi kontrol kecepatan.

Uji coba selanjutnya adalah untuk mencari tahu apakah kecepatan mempengaruhi akurasi alat. Dengan menggunakan beban yang lebih berat yaitu seberat 1.5 kg. Percobaan dilakukan dengan menggunakan setpoint kecepatan 100 RPM dan menggunakan menggunakan beban 1kg,  $k_p = 0.2$ ,  $k_i = 0.1$ .

Tabel IV-3. Pengujian kecepatan ke-2

	Distance (cm)	Rotasi Per Menit	Output(PWM)
	10	68	45
	11	105	35
	12	97	38
	13	101	37
	14	99	39
	15	100	38
	16	101	38
	17	102	38
	18	107	36
	19	95	46
	20	113	41
	21	94	46
	22	112	42
	23	90	51
	24	119	41
	25	81	59
	26	97	57
	27	125	46
	28	117	51
	29	122	47
	30	94	55
	31	110	50
	32	102	50
	33	90	55
	34	113	50
	35	91	54
	36	117	48
	37	90	56
	38	110	52
	39	112	49
Rata-Rata	24.5	102.4667	46.33333

Dengan menggunakan persamaan 4.7, kita dapat mencari tahu error dari alat setelah beban ditambahkan.

$$\text{Error} = \frac{(100 - 102.4667)}{100} \times 100\% = 2,4667\%$$

(4.7)

Didapatkan error yang terjadi pada alat ketika beban dikurangi adalah 2,4667% dan akurasi adalah 97,5333%. Error menjadi lebih kecil dibandingkan dengan error yang terjadi pada saat beban lebih berat. Tetapi jika dibandingkan pada saat beban lebih ringan alat memiliki rentang error yang lebih besar yaitu sekitar 80-120 RPM dibandingkan dengan pada saat beban lebih ringan yang memiliki rentang error sekitar 90-110 RPM

#### 4.5 Pengaruh beban pada akurasi kendali Posisi

Pengujian yang terakhir ini dilakukan untuk mencari tahu pengaruh beban pada akurasi kendali posisi. Beban ditambahkan menjadi 1.5 kg. Percobaan ini menggunakan jarak 50cm sebagai setpointnya dan menggunakan  $K_p = 5$  dan  $K_i = 0.2$  untuk konstanta proporsional dan integralnya. Tabel IV-2 menunjukkan hasil dari uji coba tersebut.

Tabel IV-4. Hasil uji coba posisi 2

	Jarak (cm)	Output (PWM)
	45.05	28
	45.33	28
	45.6	28
	46.19	50
	46.49	53
	46.51	55
	46.87	65
	46.94	69
	47.18	69
	47.29	69
	47.41	69
	47.46	70
	47.54	70
	47.65	70

	47.82	69
	48.33	67
	48.44	67
	48.52	67
	49	64
	49.85	60
	50.31	59
	49.97	61
	49.73	62
	49.86	61
	49.59	63
	49.73	62
	50.17	60
	49.93	61
	50.02	61
	50.46	58
Rata-Rata	48.17467	59.83333

Dengan menggunakan persamaan 4.8 kita dapat mencari tahu error yang terjadi.

$$\text{Error} = ((50 - 48.17467) / 50) \times 100\% = 3,657066\%$$

(4.8)

Didapatkan error yang terjadi pada saat percobaan ini adalah 3,657066% dan akurasi 96,3492934%. Hal ini lebih besar dibandingkan dengan error yang terjadi pada percobaan 4.5. Hal ini disebabkan karena alat menggunakan beban yang lebih berat.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan dari hasil penelitian dan pengujian yang didapat penulis dapat menyimpulkan bahwa :

1. Alat yang dibuat mampu untuk mengontrol kecepatan dan posisi dari lift.
2. Pada saat alat menggunakan beban 1 kg,  $K_p = 0.2$ ,  $K_i = 0.1$  dan setpoint kecepatan 100 RPM kendali kecepatan pada alat memiliki error sebesar 3.7% dan akurasi sebesar 96.3%.
3. Pada saat alat menggunakan beban 1kg,  $K_p = 5$ ,  $K_i = 0.2$  dan setpoint 50 cm kendali posisi pada alat memiliki error sebesar 3,579334% dan akurasi 96,420666%.
4. Pada saat alat menggunakan beban 1.5 kg,  $K_p = 0.2$ ,  $K_i = 0.1$  dan setpoint kecepatan 100 RPM kendali kecepatan pada alat memiliki error sebesar 2.4667% dan akurasi sebesar 97,5333%.
5. Pada saat alat menggunakan beban 1.5 kg,  $K_p = 5$ ,  $K_i = 0.2$  dan setpoint jarak 50 cm. Kendali posisi pada alat menghasilkan error sebesar 3,657066% dan akurasi sebesar 96,3492934%.
6. Model skala lift sukses diimplementasikan dengan kendali kecepatan dan posisi.
7. Kendali PID telah berfungsi sesuai dengan harapan yaitu kendali PID dapat mengatur kecepatan putar motor sesuai dengan yang operator inginkan dan menyebabkan alat menuju ketinggian yang diinginkan oleh operator.

### 5.2 Saran

Penulis merasa masih banyak kekurangan yang terdapat dari tugas akhir ini. Maka penulis menyarankan untuk orang yang ingin mengembangkan alat ini lebih lanjut dengan cara :

1. Menggunakan penyangga yang lebih kokoh dari besi bolong.
2. Menggunakan pemberat ke tatakan alat sehingga alat tidak tertarik ke atas pada saat beban sangat berat.
3. Menggunakan motor dengan rentang PWM yang lebih tinggi.
4. Menggunakan sensor yang lebih akurat dari LM393 dan HC-SR04.
5. Menggunakan motor yang memiliki torsi lebih besar sehingga motor dapat menahan beban pada saat alat sedang dalam posisi turun.

### Daftar Pustaka

- [1] Pengertian Arduino UNO, [Online]. Diambil dari iLearning.me : <https://illearning.me/sample-page-162/Arduino/pengertian-Arduino-uno/>.
- [2] Arduino – Ultrasonic Sensor, [Online]. Diambil dari Tutorialspoint : [https://www.tutorialspoint.com/Arduino/Arduino\\_ultrasonic\\_sensor.htm](https://www.tutorialspoint.com/Arduino/Arduino_ultrasonic_sensor.htm).
- [3] Arduino DC Motor Kontrol Tutorial, [Online]. Diambil dari Howtomechatronics.com : <https://howtomechatronics.com/tutorials/Arduino/Arduino-dc-motor-kontrol-tutorial-1298n-pwm-h-bridge/>
- [4] Pengertian Kendali P.I.D (Proportional-Integral-Derivative Kontroller), [Online]. Diambil dari Catatan Elektro : <http://catatan-elektro.blogspot.com/2011/11/pengertian-kendali-pid.html>