

Perancangan Sistem Pengendali Beda Tekanan Berbasis Kontrol PID Pada Ventilator

Muhammad Ikhsan¹, Sony Sumaryo², Erwin Susanto³

^{1,2,3} Universitas Telkom, Bandung

ikhsanmhd@student.telkomuniversity.ac.id¹, sonysumaryo@telkomuniversity.ac.id²,
erwinelektro@telkomuniversity.ac.id³

ABSTRAK

Pandemi COVID-19 merupakan penyakit menular yang disebabkan oleh *coronavirus*. *Coronavirus* adalah penyakit yang sangat menular yang dimulai Wuhan di China pada akhir Desember 2019 dan sekarang memiliki tersebar di seluruh dunia ke sekitar 200 negara. *Coronavirus* menyebar melalui tetesan kecil dari hidung atau mulut pada saat batuk atau bersin. *Coronavirus* menyerang sistem pernafasan, manusia yang terinfeksi *coronavirus* akan mengalami sesak nafas, gagal nafas bahkan kematian. Manusia yang kesulitan bernafas memerlukan alat bantu pernafasan menggunakan ventilator.

Pada penelitian ini dirancang suatu sistem pengendali kecepatan blower pada ventilator dengan menggunakan kontrol PID (*Proportional, Integral, Derivative*). Input yang dipakai adalah nilai perbedaan tekanan yang didapat menggunakan Sensor mpxv5010dp dan Sensor *flow hamilton*, sedangkan nilai volume didapat dari nilai *flowrate* (lps) selama waktu inspirasi. Data yang diperoleh lalu di proses oleh Arduino Uno dengan kontrol PID akan menghasilkan output yang sama dengan nilai set point.

Pada penelitian ini parameter kontrol PID yang didapat yaitu $K_p = 5036.6705$, $K_i = 3572.1355$ dan $K_d = 1775.412$. Pada set point 0.6 Sistem mencapai risetime 0.15s, settling time 0.53s, overshoot sebesar 14.55%. Pada setpoint 0.1 sistem mencapai rise time 0.5s, settling time 1.2s, overshoot 4%. Pada set point 0.16 sistem mencapai risetime 0.7s, settling time 0.9 s dan tidak memiliki overshoot . Pada set point 0.2 sistem mencapai risetime 0.8s, settling time 1.2s, dan tidak memiliki over shoot. Set point 0.2 merupakan set point tertinggi, volume yang di hasilkan dengan sistem kontrol PID dalam waktu 1s yaitu 0.525 l.

Kata Kunci: Ventilator, Coronavirus, PID.

ABSTRACT

The COVID-19 pandemic is an infectious disease caused by a coronavirus. Coronavirus is a highly contagious disease that started in Wuhan in China at the end of December 2019 and now has spread worldwide to about 200 countries. Coronavirus spreads through small droplets from the nose or mouth when coughing or sneezing. Coronavirus attacks the respiratory system, humans infected with coronavirus will experience shortness of breath, respiratory failure, and even death. Humans who have difficulty breathing need a respirator using a ventilator.

In this research, a blower speed control system on the ventilator is designed using PID (*Proportional, Integral, Derivative*) control. The input used is the pressure difference value obtained using the mpxv5010dp sensor and the Hamilton flow sensor, while the volume value is obtained from the flowrate (lps) value during the inspiration time. The data obtained and then processed by Arduino Uno with PID control will produce the same output as the setpoint value.

In this research, the PID control parameters obtained $k_p = 5036.6705$, $k_i = 3572.1355$, and $k_d = 1775.412$. At set point 0.6, The system reaches rise time 0.15s, settling time 0.53s, overshoot by 14.5%. At setpoint 0.1 the system reaches a rise time of 0.5s, a settling time of 1.2s, an overshoot of 4%. At set point 0.16 the system reached a rise time of 0.7s, settling time of 0.9s, and had no overshoot. At a set point of 0.2 the system achieves a rise time of 0.8s, a settling time of 1.2s, and has no overshoot. Setpoint 0.2 is the highest set point, the volume produced by the PID control system in 2s is 0.525 l.

Keyword: Ventilator, Coronavirus, PID.

1. Pendahuluan

Pandemi COVID-19 merupakan penyakit menular yang disebabkan oleh virus corona baru. Ini adalah penyakit yang sangat menular yang dimulai Wuhan di China pada akhir Desember 2019 dan sekarang memiliki tersebar di seluruh dunia ke sekitar 200 negara[1]. Pasien COVID-19 dapat mengalami gangguan pernapasan akut Sindroma (ARDS), yang menyebabkan kesulitan bernapas yang ekstrem karena cairan yang bocor ke paru-paru. Ventilator dapat membantu untuk merawat pasien ini dengan menyediakan oksigen sementara penyakit yang mendasari berjalan dengan sendirinya. Pemberian oksigen yang tepat adalah andalan perawatan kritis dan dalam COVID-19 dapat mencegah kematian akibat ARDS dan hipoksemia[2].

Ventilator diperlukan dalam kasus memasok oksigen yang cukup (O_2) ke paru-paru dan juga mengeluarkannya karbon dioksida (CO_2), sebagai ukuran pendukung untuk menyelamatkan nyawa[1]. Ketidak-seimbangan antara tekanan yang masuk ke dalam paru paru dengan ritme pernafasan pada ventilator akan membuat pasien tidak nyaman, maka di perlukan kontrol cerdas[3]. Pandemi Covid-19 yang sedang berlangsung telah menempatkan ketegangan yang mencolok pada sistem perawatan kesehatan global. Dengan pasokan ventilator yang terbatas telah terjadi drastic peningkatan jumlah pasien yang terkena pandemi COVID-19 di rumah sakit dan ICU di seluruh dunia. Namun ventilatornya tidak cukup tersedia di rumah sakit saat ini[4].

Dalam penelitian ini, diusulkan sistem kontrol yaitu sistem kontrol PID untuk mengontrol kecepatan blower agar sesuai dengan set point yang diinginkan. Blower berguna untuk mendorong udara ke dalam paru paru. Kontrol PID berguna untuk meningkatkan ketepatan pada proses respirasi pasien dan mengurangi hal yang tidak di inginkan. Dengan ada nya sistem ini diharapkan bisa membantu pasien dan penyediaan fasilitas medis.

Rumusan masalah dalam penelitian ini adalah Bagaimana mengukur kecepatan (RPM) blower, mengimplementasikan kontrol PID pada ventilator blower sehingga dapat mengontrol kecepatan sesuai dengan setpoint dan membuat *Mode Volume Control* pada ventilator.

Adapun tujuan dari penelitian ini adalah untuk mengetahui kecepatan (RPM) blower. merancang kontrol PID untuk mengatur kecepatan blower sesuai dengan setpoint dan dapat mengontrol ventilator menggunakan *mode volume control*.

Batasan masalah dalam penelitian ini adalah menggunakan *flow sensor* Hamilton, sensor perbedaan tekanan MPXV5010DP, mikrokontroler Arduino Uno, menggunakan Blower WM7040 dan Driver Blower WM7040. Menggunakan kontrol PID untuk mengatur kecepatan blower dan nilai setpoint berupa nilai perbedaan tekanan.

2. Metode Penelitian

2.1 Sistem Pernafasan Manusia

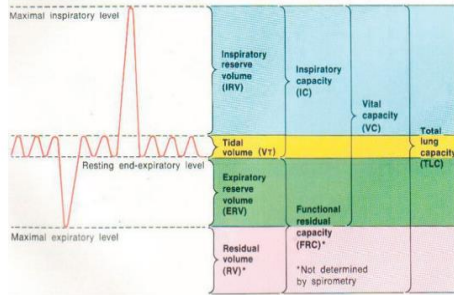
Pengertian pernafasan atau respirasi adalah suatu proses mulai dari pengambilan oksigen, pengeluaran karbohidrat hingga penggunaan energi di dalam tubuh. Manusia dalam bernapas menghirup oksigen dalam udara bebas dan membuang karbon dioksida ke lingkungan. Saluran pernafasan adalah sistem pernafasan pada Manusia terdiri dari hidung, faring, trakea, bronkus, bronkiouls dan paru-paru.

2.2 Volume dan Kapasitas Paru-paru Manusia

Volume dan Kapasitas Statik Pernafasan Volume dan kapasitas pernafasan merupakan gambaran fungsi ventilasi sistem respirasi. Dengan mengetahui besarnya volume dan kapasitas pernafasan dapat diketahui besarnya kapasitas ventilasi maupun ada tidaknya kelainan ventilasi pada seseorang.

- Volume tidal. (TV) = 0.5 L
- Volume cadangan inspirasi/ Inspiratory Reserve Volume (IRV) = 3.3 L
- Volume cadangan ekspirasi/ Expiratory Reserve Volume (ERV) = 1 L
- Volume residu/Residual Volume (RV) = 1.2 L
- Kapasitas inspirasi/ Inspiratory Capacity (IC) = 3.8 L
- Kapasitas residu fungsional/ Functional Residual Capacity (FRC) = 2.2 L
- Kapasitas vital/Vital Capacity (VC) = 4.8 L

-Kapasitas paru total/Total Lung Capacity (TLC) = 6 L[7].

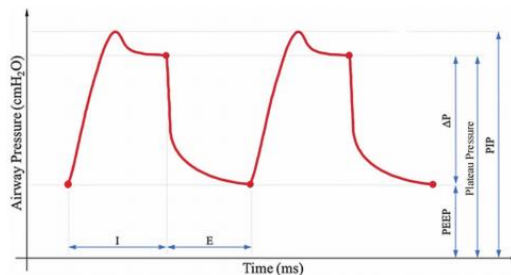


Gambar 2.1. Volume dan Kapasitas Paru Paru manusia

2.3 Tekanan Paru-paru Manusia

Pada proses pernafasan manusia, tekanan yang terjadi dibedakan dalam 3 macam diantaranya yaitu :

- PIP (*Peak Inspiratory Pressure*) = puncak tekanan yang dibutuhkan saat memberikan volume tidal pada pasien.
- Plateau Pressure = tekanan yang dibutuhkan untuk mempertahankan pengembangan paru saat inspirasi.
- PEEP (*Positive End Expiratory Pressure*) = tekanan positif yang dipertahankan saat akhir ekspirasi[8].



Gambar 2.2 Tekanan Paru Paru Manusia

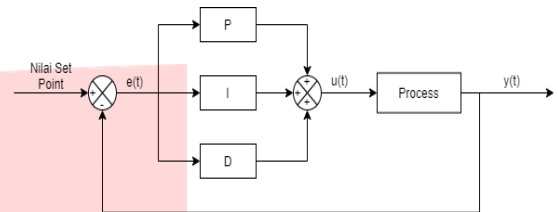
2.4 Covid-19

- Gejala ringan kasus infeksi virus Corona atau (COVID-19):
 - Batuk.
 - Letih.
 - Sesak napas dan ngilu di seluruh tubuh.
 - Secara umum merasa tidak enak badan.
- Gejala berat kasus infeksi virus Corona atau (COVID-19):
 - Kesulitan bernapas.
 - Infeksi pneumonia.
 - Sakit di bagian perut.
 - Nafsu makan turun[9].

2.5 Kontroler PID

Kontroler PID (Proportional Integral Derivative) merupakan salah satu sistem kendali yang menggunakan sistem

umpan balik. Dengan adanya karakteristik umpan balik kontroler banyak dipakai dalam dunia industri. Kontrol PID terdiri dari 3 pengendali yaitu Proporsional (P), Integral (I), dan Derivatif (D). Diagram Blok Kontrol PID dapat dilihat pada gambar 2.3. Berdasarkan persamaan P, I, dan D maka dapat diperoleh keluaran PID yang dapat dilihat persamaan (2.1)



Gambar 2.3 Blok diagram Kontrol PID

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

Dimana: $u(t)$ = Keluaran sistem kendali

- $e(t)$ = Sinyal error
- K_p = Konstanta proporsional
- K_i = Konstanta integral
- K_d = Konstanta derivatif

2.5.1 Proporsional(P)

$$u(t) = K_p e(t) \quad (2.2)$$

Kontrol P dinyatakan dalam persamaan (2.2), K_p merupakan konstanta proporsional. Kontrol P digunakan untuk memperbaiki respon transien, seperti mempercepat rise time dan settling time. Jika nilai konstanta yang diberikan terlalu besar maka respon sistem terlalu responsive dan menyebabkan sistem tidak stabil.

2.5.2 Integral (I)

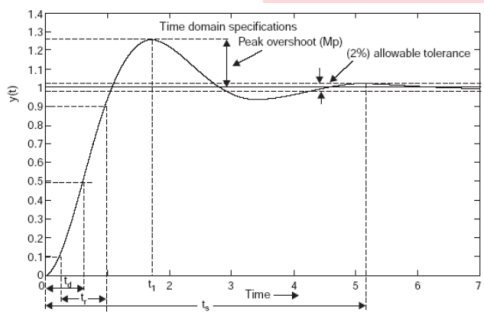
$$u(t) = K_i \int_0^t e(t) dt \quad (2.3)$$

Kontrol I dinyatakan dalam persamaan (2.3), K_i merupakan konstanta integral. Kontrol I digunakan untuk memperbaiki respon steady state. Jika pemilihan nilai K_i tidak tepat maka sistem menjadi tidak stabil dan respon transien menjadi tinggi.

2.5.3 Derivativ (D)

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.4)$$

Kontrol D dinyatakan dalam persamaan (2.4). K_d merupakan konstanta derivatif. Kontrol D digunakan untuk memperbaiki respon transien dengan cara memprediksi error yang akan terjadi. Semakin tinggi nilai K_d maka osilasi akan diredam akan semakin besar dan memperlambat respon sistem. Gambar 2.4 Kurva tanggapan keluaran sistem.



Gambar 2.4 Kurva Tanggapan Keluaran

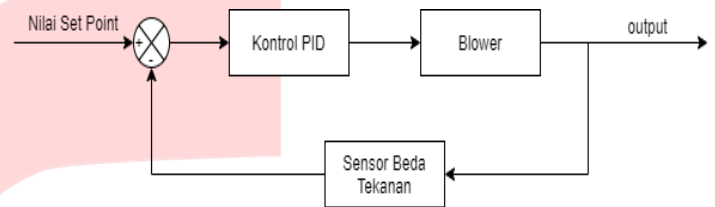
- *Delay time* (T_d), waktu yang dibutuhkan respon agar mencapai setengah nilai akhir yang pertama.
- *Rise Time* (T_r), waktu yang dibutuhkan oleh respon agar naik dari 10% sampai 90%.
- *Peaktime* (T_p), waktu yang dibutuhkan respon agar mencapai puncak pertama kali.
- *Maximum Time* (M_p), persen nilai puncak maksimum dari kurva respon.
- *Setling Time* (T_s), waktu yang dibutuhkan agar kurva respon mencapai dan menetap dalam daerah disekitar harga akhir[10].

3. Perancangan Sistem

3.1 Desain Sistem

Pada penelitian ini, nilai input pada sistem yaitu RR (*respiration rate*), I : E *ratio*, *Pressure* dan *Volume tidal*. Kemudian nilai tersebut diolah pada mikrokontroler Arduino Uno dan sistem kontrol berguna untuk mengatur dan menstabilkan kecepatan blower. Pada proses inhale blower bergerak dengan sistem kontrol selama waktu inspirasi, pada proses plateau pause blower

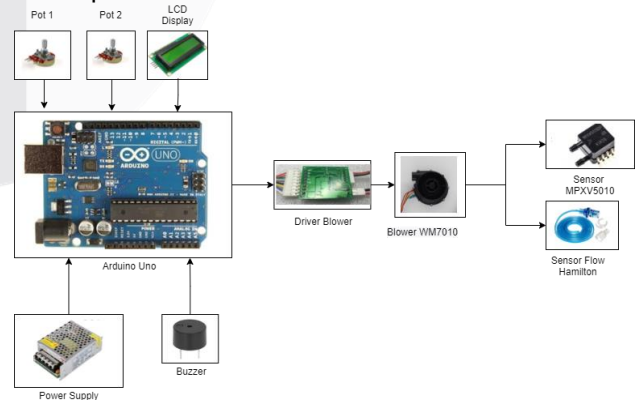
bergerak sesuai dengan ketentuan yaitu pwm 30 selama waktu plateau pause, dan pada proses expirasi blower akan berhenti bergerak selama waktu expirasi. Sensor perbedaan tekanan MPX5010DP akan memberikan hasil pembacaan pada mikrokontroler untuk diolah menjadi flowrate dan membandingkan antara nilai input dengan output. Agar sistem berjalan dengan cepat dan stabil maka di butuhkan sistem kontrol. Pada penelitian ini sistem kontrol yang di pakai yaitu kontrol PID. Gambar 3.1 merupakan diagram blok sistem



Gambar 3.1 Diagram blok sistem

3.2 Perangkat Keras

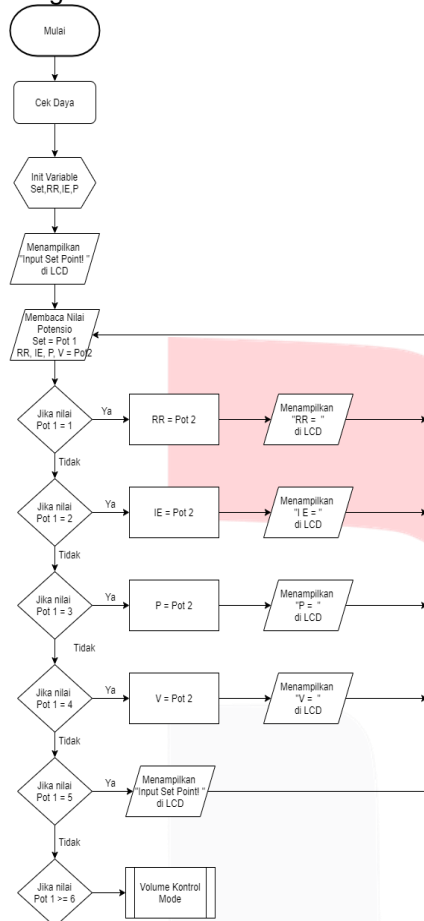
Perancangan perangkat terdiri dari mikrokontroler sebagai pengendali, blower sebagai penggerak oksigen dan udara agar mendapatkan bantuan oksigen dan udara tambahan, sensor perbedaan tekanan dan sensor *flow* Hamilton untuk mendapatkan nilai tekanan dan volume yang ada pada paru paru dan proses dapat di tampilkan pada lcd. Potensiometer berguna sebagai pengatur nilai set point dan Buzzer akan berfungsi memberikan informasi ketika terjadi *error*. Desain perangkat keras dapat dilihat pada Gambar 3.2.



Gambar 3.2 Desain Perangkat Keras

3.3 Perangkat Lunak

3.3.1 Diagram Alir Menu Utama

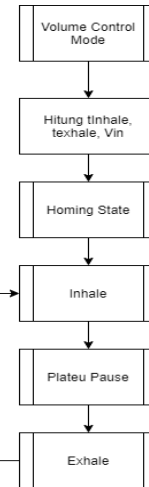


Gambar 3.3 Diagram Alir Menu Utama

Berikut alur diagram menu utama yang dibangun:

1. Sistem dimulai dari mengecek daya, menginisialisasi set dan variable.
2. Nilai Set didapat dari nilai Pot 1, dan nilai Variable RR, IE, P dan V didapat dari nilai Pot 2
3. Jika Pot1 bernilai 1 maka Pot 2 akan menginput nilai RR (*Respiration rate*).
4. Jika Pot1 bernilai 2 maka Pot 2 akan menginput nilai IE (*Inhale Exhale Ratio*).
5. Jika Pot1 bernilai 3 maka, Pot 2 akan menginput nilai P (*Pressure*).
6. Jika Pot1 bernilai 4 maka, Pot 2 akan menginput nilai V (*volume*).
7. Jika Pot1 bernilai 5 maka variable RR, IE, P dan V yang tersimpan akan ditampilkan pada lcd.
8. Jika Pot1 bernilai 6 maka sistem ventilator akan memasuki mode volume kontrol.

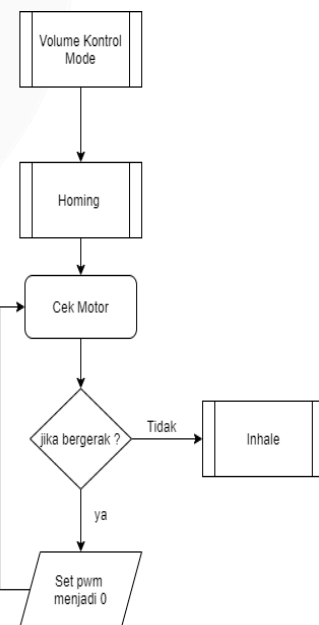
3.3.2 Diagram Alir Keseluruhan Sistem



Gambar 3.4 Diagram Alir Keseluruhan Sistem

Ketika sistem dalam kondisi start terdapat 4 pembagian yang menjadi proses inspirasi buatan kepada pasien yaitu: Menhitung nilai waktu inspirasi, waktu expirasi dan nilai volume, Inhale., Plateu Pause dan Exhale

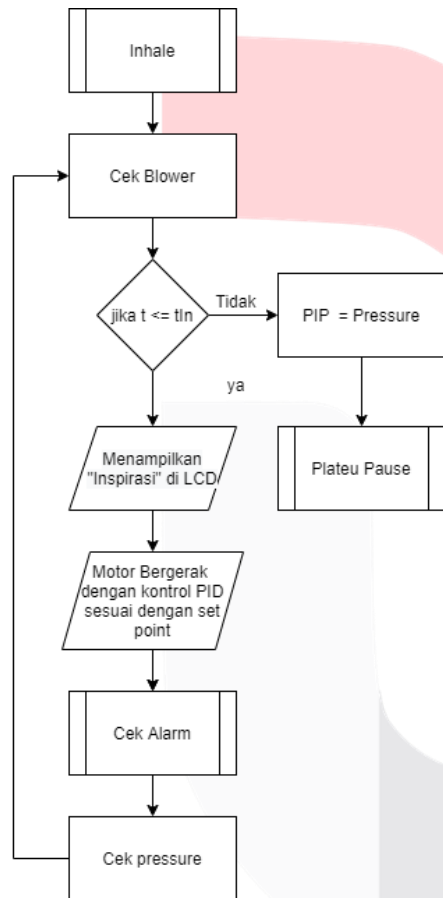
3.3.3 Diagram Alir Homing



Gambar 3.5 Diagram Alir Homing

Pada proses homing motor akan di periksa apakah sedang bergerak atau diam. Jika motor bergerak, maka nilai pwm akan di setting menjadi 0. Jika motor diam maka proses akan di lanjutkan ke proses inhale.

3.3.4 Diagram Alir Inhale

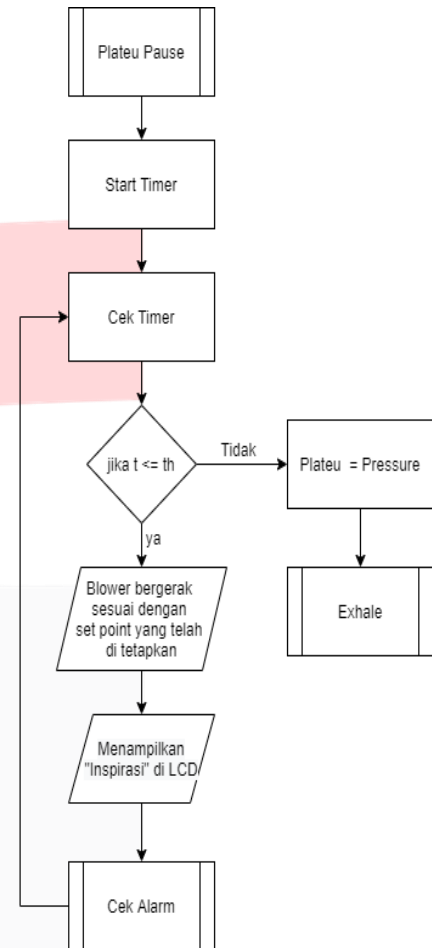


Gambar 3.6 Diagram Alir Inhale

Pada proses Inhale blower akan bergerak dengan metode kontrol PID pada waktu yang telah di input. Proses ini akan di tampilkan pada lcd. Nilai tIn didapat dari nilai variable RR dan IE. Pada proses Inhale ini blower akan bergerak sesuai dengan set point yaitu P yang telah di input sebelumnya. Jika t lebih besar dari tIn maka proses dilanjutkan ke Plateu Pause. Apabila terjadi kelebihan tekanan pada proses inhale maka alarm dan

buzzer akan menyala dan ditampilkan pada lcd bahwa telah terjadi kelebihan tekanan inspirasi.

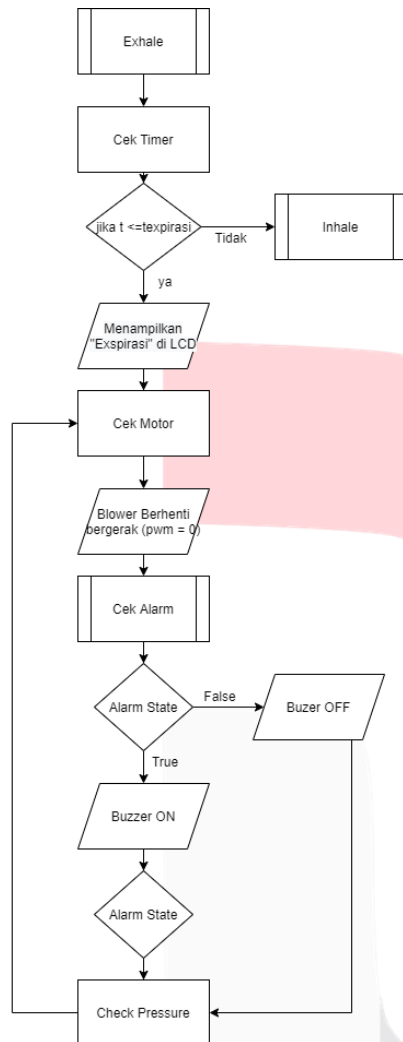
3.3.5 Diagram Alir Plateu Pause



Gambar 3.7 Diagram Alir Plateu Pause

Pada Proses Plateu pause, motor bergerak dengan mengikuti pwm yang telah terprogram. tHold merupakan waktu yang juga sudah terprogram sebelumnya yaitu 100ms. Proses ini akan di tampilkan pada lcd. Jika t lebih besar dari tHold maka proses akan di lanjutkan ke Exhale. Apabila tekanan pada plateu melebihi tekanan yang telah di tetapkan maka alarm dan buzzer akan berbunyi dan pada lcd akan di tampilkan bahwa tekanan berlebih.

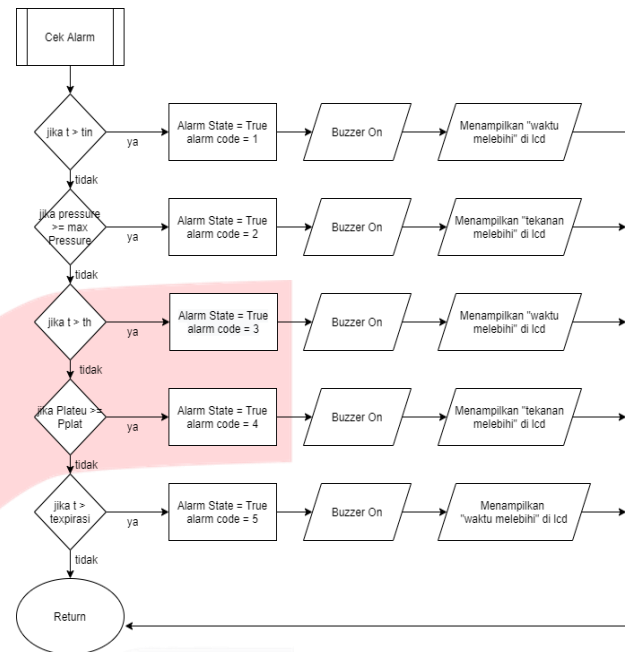
3.3.6 Diagram Alir Exhale



Gambar 3.8 Diagram Alir Exhale

Pada proses Exhale, motor akan berhenti bergerak. t_{Ex} merupakan waktu ketika motor berhenti bergerak. Proses ini akan di tampilkan pada lcd display. Pada proses ini pasien mengalami proses expirasi. t_{Ex} didapat dari waktu total yang di butuhkan selama respirasi yang di dapat dari RR dan IE dikurangnya dengan t_{IN} . Pada proses ini terdapat juga pengecekan status alarm apakah telah terjadi error atau tidak. Jika terjadi maka buzzer akan menyala dan akan melihat nilai tekanan pada proses ini.. setelah proses ini selesai akan dilanjutkan Kembali pada proses inspirasi.

3.3.7 Diagram Alir Plateu Pause



Gambar 3.9 Diagram Alir Alarm

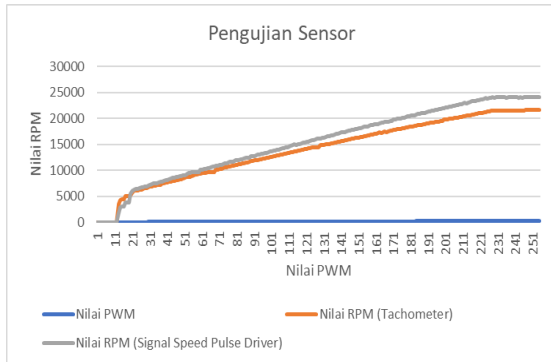
Pada alarm ada 5 pengecekan yaitu:

1. Waktu inspirasi
2. Tekanan inspirasi
3. Waktu plateu pause
4. Tekanan plateu
5. Waktu expirasi.

4. Hasil dan Pembahasan

4.1 Pengujian Motor

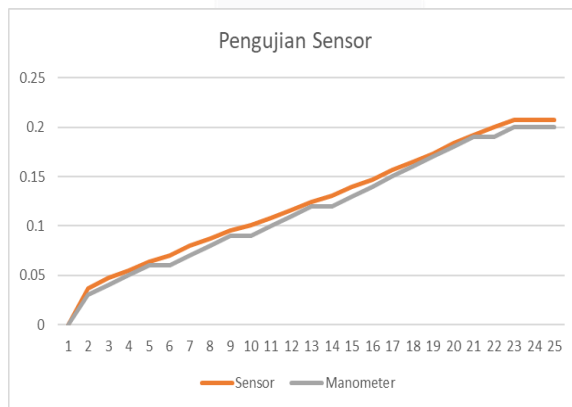
Pengujian motor BLDC Blower bertujuan untuk melihat error yang ada pada sensor hall. Pengujian dilakukan dengan memakai power supply sebesar 12V/3A. Pengujian dilakukan dengan memasukan nilai input berupa PWM dan menghasilkan Output berupa RPM, dengan pengujian PWM antara 1-255. Pengujian juga di ukur menggunakan tachometer. Error yang dimiliki sensor hall yaitu 9.97%. Jadi pada pemodelan sistem nilai input dan output yang dipakai yaitu nilai dari *tachometer*.



Gambar 4.1 Grafik Pengujian Sensor

4.2 Pengujian Sensor

Sensor perbedaan tekanan pada penelitian ini adalah sensor MPXV5010DP dengan range 0 - 10 kPa. Pengujian dilakukan dengan menginput nilai pwm yang di hubungkan dengan blower, dan memandingkan hasil dengan manometer. Grafik pada gambar 4.2 merupakan grafik perbandingan antara sensor perbedaan tekanan dengan manometer. Diperoleh error sensor perbedaan tekanan sebesar 5,6%.



Gambar 4.2 Grafik Pengujian Sensor

4.3 Pemodelan Sistem

Pemodelan sistem dilakukan menggunakan *software* MATLAB2020b. Pemodelan menggunakan aplikasi pada *software* MATLAB2020b yaitu *System Identification*. *System Identification* dapat mengolah data input dan output dari suatu system menjadi berbagai macam model, diantaranya yaitu fungsi transfer. Fungsi transfer dapat diperoleh dengan memasukan nilai pole dan nilai zero. Tabel 4.1 merupakan hasil dari beberapa nilai pole dan nilai zero.

Tabel 4.1 Percobaan System Identification

| Model | Pole | Zero | Persentasi Ketepatan |
|-------|------|------|----------------------|
| TF 1 | 1 | 0 | 73.77 |
| TF 2 | 2 | 0 | 88.11 |
| TF 3 | 2 | 1 | 94.64 |
| TF 4 | 3 | 0 | 68.23 |
| TF 5 | 3 | 1 | 94.65 |
| TF 6 | 3 | 2 | 94.64 |
| TF 7 | 4 | 0 | 34.14 |
| TF 8 | 4 | 1 | 90.18 |
| TF 9 | 4 | 2 | 94.66 |
| TF 10 | 4 | 3 | 93.56 |
| TF 11 | 5 | 0 | 63.23 |
| TF 12 | 5 | 1 | -289.2 |
| TF 13 | 5 | 2 | 87.65 |
| TF 14 | 5 | 3 | 52.37 |
| TF 15 | 5 | 4 | -165.6 |

Dari 15 hasil percobaan didapatkan model dengan persentase ketepatan paling tinggi yaitu TF 9 dengan pole 4 dan zero 2. Memiliki persentase ketepatan sebesar 94.66. Tabel 4.1 merupakan hasil persentase pemodelan. Berikut fungsi transfer yang di hasilkan dari pemodelan system TF 9:

$$\frac{0.04558s^2 + 0.03369s + 0.005253}{s^4 + 2.897s^3 + 12.59s^2 + 17.18s + 6.766}$$

4.4 PID Tunner

PID tunner merupakan aplikasi untuk menemukan parameter PID (kp, ki, kd) sesuai dengan kondisi yang diinginkan. Pada penelitian ini ventilator membutuhkan rise time dan settling time yang cepat agar udara yang di masukan ke dalam paru paru pasien sesuai dengan kebutuhan. Proses tuning parameter ini dilakukan dengan memasukan Plant sebelumnya . Plant yang diambil adalah plant yang di pakai yaitu TF 9. Dengan menggeser pada bagian *Response time* ke *Faster* dan menggeser *Transient Behavior* ke arah *Robush* sesuai dengan kebutuhan maka menghasilkan Grafik dan Nilai parameters. Gambar 4.3 menunjukan parameter PID yang di dapat dilihat juga terdapat *rise time*, *settling time* dan *overshoot* pada TF 9.

| Controller Parameters | |
|----------------------------|---------------------|
| | Tuned |
| Kp | 5036.6705 |
| Ki | 3572.1355 |
| Kd | 1775.412 |
| Tf | n/a |
| | |
| Performance and Robustness | |
| | Tuned |
| Rise time | 0.0266 seconds |
| Settling time | 0.0458 seconds |
| Overshoot | 0 % |
| Peak | 1 |
| Gain margin | Inf dB @ NaN rad/s |
| Phase margin | 89.5 deg @ 81 rad/s |
| Closed-loop stability | Stable |

Gambar 4.3 Hasil Tuner

Pada percobaan proses tuning yang telah dilakukan pada PID tuner sebelumnya, diperoleh parameter pid: $K_p = 5036.6705$, $K_i = 3572.1355$ dan $K_d = 1775.412$. Dengan rise time = 0.0266s, settling time = 0.0458 s, dan overshoot 0%. Selanjutnya parameter ini akan diuji pada Simulink dan percobaan pada system ventilator.

4.5 Pengujian Simulink

Simulink merupakan bagian tambahan dari software MATLAB. Simulink dapat digunakan sebagai sarana pemodelan, simulasi dan analisis dari sistem dinamik dengan menggunakan antarmuka grafis. Pada percobaan simulasi, fungsi transfer yang di pakai merupakan fungsi transfer yang di dapat dari System Identification.

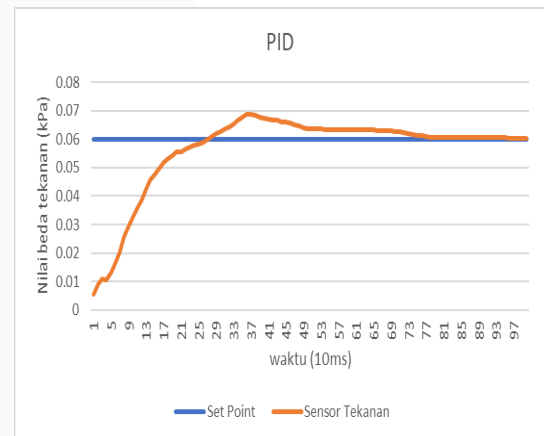
| Trace Selection | | Trace Selection | |
|--------------------------|-------------|--------------------------|--------------|
| x(t) | | y(t) | |
| Bilevel Measurements | | Bilevel Measurements | |
| Settings | | Settings | |
| Transitions | | Transitions | |
| High | 9.573e-01 | High | 3.721e-03 |
| Low | 5.665e-03 | Low | 1.659e-05 |
| Amplitude | 9.517e-01 | Amplitude | 3.704e-03 |
| + Edges | 1 | + Edges | 1 |
| + Rise Time | 18.537 ms | + Rise Time | 375.613 ms |
| + Slew Rate | 41.071 (/s) | + Slew Rate | 7.889 (/ks) |
| - Edges | 0 | - Edges | 1 |
| - Fall Time | -- | - Fall Time | 457.896 ms |
| - Slew Rate | -- | - Slew Rate | -6.471 (/ks) |
| Overshoots / Undershoots | | Overshoots / Undershoots | |
| + Preshoot | 0.595 % | + Preshoot | 0.448 % |
| + Overshoot | 18.452 % | + Overshoot | 29.452 % |
| + Undershoot | -3.181 % | + Undershoot | 0.002 % |
| + Settling Time | -- | + Settling Time | -- |
| - Preshoot | -- | - Preshoot | 29.452 % |
| - Overshoot | -- | - Overshoot | 42.121 % |
| - Undershoot | -- | - Undershoot | 7.534 % |
| - Settling Time | -- | - Settling Time | -- |
| Cycles | | Cycles | |

Gambar 4.4 Perbandingan antara memakai PID dan tanpa PID

Dapat dilihat bahwa system dapat mencapai Risetime 18.537 ms dengan memakai kontrol PID. Pada Grafik juga terlihat bahwa dengan memakai kontrol PID overshoot sebesar 18.452%. Sedangkan tanpa kontrol PID sistem mencapai rise time 375.13ms, dan overshoot sebesar 0.448%. Gambar 4.4 memperlihatkan perbandingan hasil output pada system dengan memakai kontrol PID dan tanpa menggunakan kontrol PID.

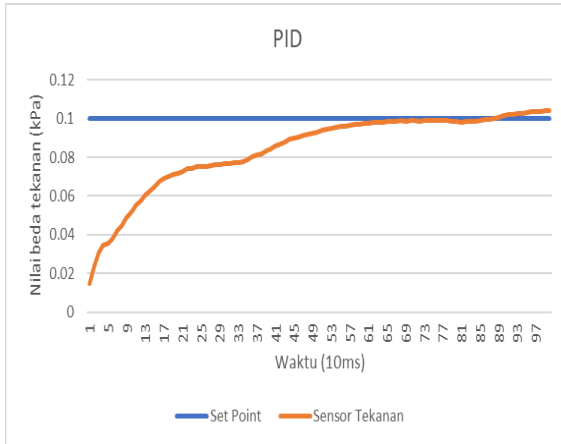
4.6 Pengujian Pada Ventilator

Percobaan dilakukan dengan menggunakan sensor pada ventilator, sensor yang di pakai yaitu MPX5010DP. Kodingan dibuat di Arduino IDE yang selanjutnya akan di uji coba pada ventilator. Nilai parameter yang didapat dari pid tuner $K_p = 5036.6705$, $K_i = 3572.1355$ dan $K_d = 1775.412$. Gambar 4.5 merupakan grafik yang telah diuji coba dengan dengan set point 0.6 . Sistem mencapai risetime 0.15s, settling time 0.53s, dan memiliki overshoot sebesar 14.5%.



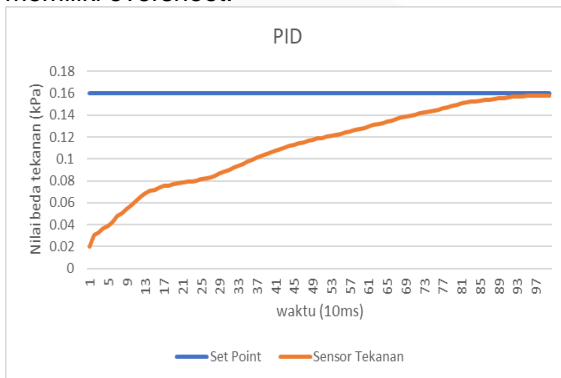
Gambar 4.5 Grafik dengan setpoint 0.06

Gambar 4.6 merupakan percobaan pada ventilator dengan setpoint 0.1. Pada percobaan terlihat pada gambar 4.11 bahwa sistem mencapai rise time 0.5 s, settling time 1.2 s dan memiliki overshoot sebesar 4%.

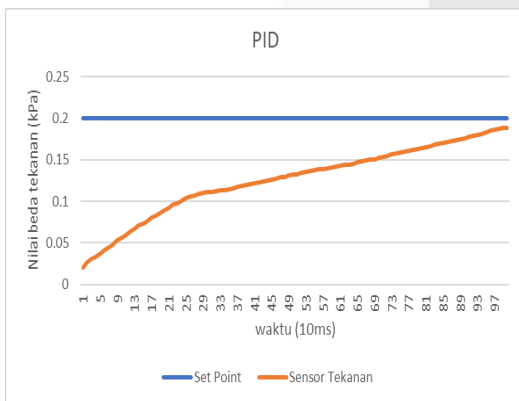


Gambar 4.6 Grafik dengansetpoint 0.1

Pada percobaan dengan set point 0.16 terlihat pada grafik gambar 4.12 bahwa sistem mencapai risetime 0.7s, settling time 0.9s dan tidak memiliki overshoot Pada Percobaan dengan set point 0.2 terlihat pada grafik gambar 4.13 bahwa sistem mencapai risetime 0.8s, settling time 1.2s, dan tidak memiliki overshoot.



Gambar 4.7 Grafik dengan set point 0.16



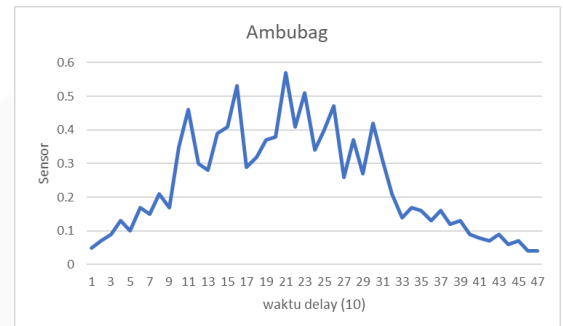
Gambar 4.8 Grafik dengan set point 0.2

4.7 Pengujian Sistem Mode Volume Kontrol.

Mode volume kontrol adalah sistem untuk mengatur seberapa banyak udara yang masuk ke dalam tubuh pasien, dengan persamaan 4.1 banyak volume yang masuk dapat di ukur.

$$V = Q \times t \quad (4.1)$$

Percobaan dilakukan dengan menggunakan sensor perbedaan tekanan dan sensor flow hamilton. Mode kontrol sudah ada sebelumnya pada ventilator ambubag. Ini dapat dijadikan acuan dikarenakan ambu bag memiliki kapasitas yang tetap. Gambar 4.14 percobaan pada ambu bag menggunakan sensor perbedaan tekanan MPXV5010DP dan sensor flow Hamilton.



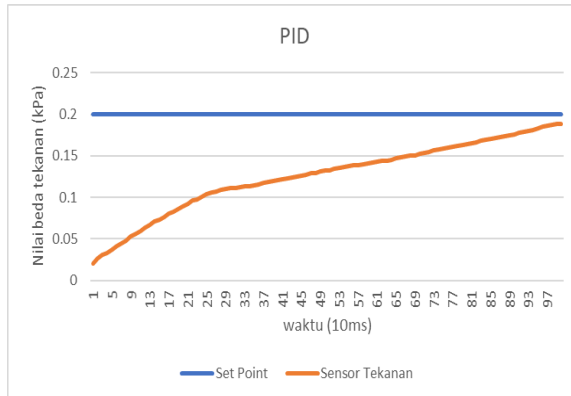
Gambar 4.9 Percobaan dengan ambubag

Percobaan dilakukan dengan ambubag dikarenakan ambubag mempunyai kapasitas max yaitu 700ml. percobaan dilakukan untuk mendapatkan nilai dari k dari persamaan 4.2 yang merupakan persamaan dari sensor flow Hamilton.

$$Q = k\sqrt{\Delta P} \quad (4.2)$$

$$k = \frac{V}{t \times \sqrt{\Delta P}} \quad (4.3)$$

$$k = 1.508842 \quad (4.4)$$



Gambar 4.9 Percobaan dengan setpoint 0.2

Pada gambar 4.9 diketahui bahwa nilai input yaitu 0.2 yang merupakan nilai perbedaan tekanan selama 1s dengan memakai kontrol PID. Dengan memasukan ke persamaan 4.1 maka didapat nilai volume sebesar 0.525 L. Pada mode volume kontrol ini persamaan yang digunakan untuk mendapat nilai perbedaan tekanan yang juga digunakan sebagai set point yaitu persamaan 4.5

$$\Delta P = \left(\frac{v}{K \times tIn} \right)^2 \quad (4.5)$$

5. Kesimpulan

Berdasarkan dari proses dan analisis dalam penelitian yang telah di kerjakan serta implementasi sistem pada ventilator, dapat diperoleh beberapa kesimpulan yaitu :

1. Berdasarkan pengujian sensor hall yang ada pada motor BLDC (Blower WM7040) dan tachometer, sensor memiliki error 9,97%. Pengukuran kecepatan pada blower dapat di lihat menggunakan sensot hall yang ada pada motor BLDC
2. Sensor MPXV5010DP dan Manometer, sensor memiliki error 5,6%.. Hasil pemodelan terbaik memiliki orde 4 dengan pole 4 dan zero 2 dengan hasil ketepatan sebesar 94.66. Hasil tunner parameter kontrol PID $K_p = 5036.6705$, $K_i = 3572.1355$ dan $K_d = 1775.412$. Dengan rise time = 0.0266 s, settling time = 0.0458 s, dan overshoot 0%. Hasil percobaan sistem pada Simulink. Dengan kontrol PID system dapat

mencapai risetime 18.537 ms, overshoot 18.452%. Sedangkan tanpa kontrol PID sistem mencapai rise time 375.13 ms, dan overshoot sebesar 0.448%. Berdasarkan hasil percobaan dengan ventilator. Pada set point 0.6 dalam waktu 2 detik. Sistem mencapai risetime 0.15 s, settling time 0.53 s, dan memiliki overshoot sebesar 14.5%. Pada setpoint 0.1 sistem mencapai rise time 0.5s, settling time 1.2 s dan memiliki overshoot 4%. Pada set point 0.16 sistem mencapai risetime 0.7 s, settling time 0.9 s dan tidak memiliki overshoot. Pada set point 0.2 sistem mencapai risetime 0.8 s, settling time 1.2 s, dan tidak memiliki overshoot.

3. Mode volume control dapat digunakan dengan acuan flowrate. Dimana nilai flowrate akan dikalikan dengan waktu inspirasi yang merupakan volume tidal yang di hasilkan.

6. Saran

Pada perancangan dan implementasi penelitian ini masih terdapat beberapa kekurangan dan hal hal yang perlu dikembangkan, diantaranya adalah

1. Pada penelitian ini sistem kontrol berkerja dengan baik terhadap set point yang diberikan. Diharapkan untuk kedepannya agar ada parameter lain selain perbedaan tekanan.
2. Diharapkan pada penelitian selanjutnya ventilator diuji menggunakan paru paru buatan agar sistem kontrol bekerja lebih baik.

Referensi

- [1] K. Iyengar, S. Bahl, Raju Vaishya, and A. Vaish, "Challenges and solutions in meeting up the urgent requirement of ventilators for COVID-19 patients," *Diabetes Metab. Syndr. Clin. Res. Rev.*, vol. 14, no. 4, pp. 499–501, 2020, doi: 10.1016/j.dsx.2020.04.048.
- [2] R. M. Corey *et al.*, "Low-Complexity System and Algorithm for an Emergency Ventilator Sensor and Alarm," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no.

- 5, pp. 1088–1096, 2020, doi:
10.1109/tbcas.2020.3020702.
- [3] Z. Li, M. Zhang, and M. Du, “The study of intelligent control algorithm in CPAP ventilator,” *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, pp. 386–391, 2017, doi:
10.1109/CAC.2017.8242797.
- [4] J. Zuckeberg, M. Shaik, K. Widmeier, T. Kilbaugh, and T. D. Nelin, “A lung for all: Novel mechanical ventilator for emergency and low-resource settings,” *Life Sci.*, vol. 257, no. June, p. 118113, 2020, doi: 10.1016/j.lfs.2020.118113.
- [5] S. M. Tamjid Hossain, M. Ranjan Halder, A. Aman, and T. Rahman, “Design Construction and Performance Test of a Low-Cost Portable Mechanical Ventilator for Respiratory Disorder,” *Int. Conf. Mech. Ind. Energy Eng.*, no. December 2018, pp. 1–6, 2018.
- [6] Z. L. Chen, Z. Y. Hu, and H. De Dai, “Control system design for a continuous positive airway pressure ventilator,” *Biomed. Eng. Online*, vol. 11, no. 1, p. 5, 2012, doi: 10.1186/1475-925X-11-5.
- [7] A. Bakhtiar and W. S. Amran, “Faal Paru Statis,” *J. Respirasi*, vol. 2, no. 3, p. 91, 2019, doi: 10.20473/jr.v2-i.3.2016.91-98.
- [8] A. Christou, M. Ntagios, A. Hart, and R. Dahiya, “GlasVent—The Rapidly Deployable Emergency Ventilator,” *Glob. Challenges*, vol. 4, no. 12, p. 2000046, 2020, doi: 10.1002/gch2.202000046.
- [9] K. Karyono, R. Rohadin, and D. Indriyani, “Penanganan Dan Pencegahan Pandemi Wabah Virus Corona (Covid-19) Kabupaten Indramayu,” *J. Kolaborasi Resolusi Konflik*, vol. 2, no. 2, p. 164, 2020, doi: 10.24198/jkrk.v2i2.29127.
- [10] M. Ali, “Pembelajaran Perancangan Sistem Kontrol Pid Dengan Software Matlab,” *J. Edukasi Elektro*, vol. 1, no. 1, p. 2, 2004.